JOURNAL OF CHEMICAL PHYSICS VOLUME 117, NUMBER 3 15 JULY 2002

Constant-temperature molecular-dynamics algorithms for mixed hard-core/continuous potentials

Yao A. Houndonougbo and Brian B. Laird[a)]
Department of Chemistry, University of Kansas, Lawrence, Kansas 66045

(Received 8 April 2002; accepted 19 April 2002)

We present a set of second-order, time-reversible algorithms for the isothermal (NVT) molecular-dynamics (MD) simulation of systems with mixed hard-core/continuous potentials. The methods are generated by combining real-time Nosé thermostats with our previously developed Collision Verlet algorithm [Mol. Phys. **98**, 309 (1999)] for constant energy MD simulation. In all we present five methods, one based on the Nosé–Hoover [Phys. Rev. A **31**, 1695 (1985)] equations of motion and four based on the Nosé–Poincaré [J. Comput. Phys. **151**, 114 (1999)] real-time formulation of Nosé dynamics. The methods are tested using a system of hard spheres with attractive tails and all correctly reproduce a canonical distribution of instantaneous temperature. The Nosé–Hoover based method and two of the Nosé–Poincaré methods are shown to have good energy conservation in long simulations.
[DOI: 10.1063/1.1485072]

I. INTRODUCTION

Algorithms for molecular-dynamics simulation can generally be divided into two distinct classes depending upon the nature of the potential.[1] For systems with continuously differentiable potentials, the trajectory is generated through the numerical integration of the equations of motion—a coupled set of differential equations—typically with a fixed time step. At the other end of the spectrum are methods for discontinuous potentials, such as hard spheres or the square-well potential. Such algorithms are event driven in that the system is advanced ballistically between ''collisions,'' which are then resolved exactly. There exist, however, model interaction potentials of theoretical and practical importance that are hybrids of continuous and discontinuous potentials, for example, the restricted primitive model for electrolyte solutions or the dipolar hard-sphere model of polar fluids. To date, simulation studies for such systems have been primarily restricted to Monte Carlo studies due to the lack of a viable molecular-dynamics (MD) algorithm. To remedy this, we have recently introduced a new molecular-dynamics method for such systems.[2] The algorithm, referred to as Collision Verlet, has good energy conservation and is far more stable over long time simulation than previous integrators for hybrid continuous/discontinuous systems. The Collision Verlet algorithm was formulated as a constant energy simulation method, which generates configurations from a microcanonical (NVE) distribution. However, to mimic experimental conditions most modern simulations are run under isothermal (NVT) or isothermal/isobaric (NPT) conditions. In this work, we introduce and evaluate several reformulations of Collision Verlet to generate trajectories whose phase space points are canonically (isothermally) distributed.

The NVT (isothermal) Collision Verlet algorithms developed here are all based on the extended Hamiltonian of Nosé,[3] which is a standard technique for generating canonical trajectories for the simulation of systems with continuous interaction potentials. In the Nosé approach, the phase space of the system is augmented by the introduction of an auxilliary variable s and its conjugate momentum π (with ''mass'' Q). For a system with a potential V, the Nosé extended Hamiltonian is

$$\mathcal{H}_{\text{Nosé}}=\sum_i \frac{\tilde{p}_i^2}{2m_i s^2}+V(\mathbf{q})+\frac{\pi^2}{2Q}+gkT\ln s, \tag{1}$$

where $\tilde{p}_i$ is the momentum conjugate to the position q_i and is related to the actual momentum, p_i, by the relation $p_i=\tilde{p}_i/s$, and the parameter $g=N_f+1$, where N_f is the number of degrees of freedom of the system. With this choice of g, it can be readily shown,[3] assuming ergodicity, that constant energy (microcanonical) dynamics generated by the Nosé Hamiltonian produces a canonical (constant temperature) distribution in the reduced phase space $\{\tilde{\mathbf{p}}/s,\mathbf{q}\}$.

The generation of phase space configurations distributed in the canonical ensemble within the Nosé dynamical scheme is accomplished by a dynamical rescaling of time. The real time of the simulation, t, is related to the Nosé time, τ, by the transformation $d\tau/dt=s$. Since numerical integration methods generally operate with a fixed time step, the transformation to real time generates a nonuniform grid of time points,[4] which is inconvenient for the calculation of system averages. To remedy this, two schemes have been developed to produce equations of motion for Nosé dynamics that generate trajectories directly in real time. By applying time and coordinate transformations directly to the Nosé equations of motion, Hoover[5] derived a set of real-time equations of motion for Nosé dynamics, defining the so-called Nosé–Hoover method. Although this approach is currently the most popular isothermal simulation method, it has a drawback that the equations are non-Hamiltonian in structure due to the non-canonical coordinated transformation, which

[a)] Author to whom correspondence should be addressed.

precludes the use of symplectic integration schemes.[6] In an alternate approach Bond, Leimkuhler, and Laird[7] apply a Poincaré time transformation to the Nosé Hamiltonian to give the so-called Nosé–Poincare Hamiltonian, from which real-time, fully Hamiltonian equations of motion for Nosé dynamics are generated.

In this work we present constant-temperature simulation methods for mixed continuous/discontinuous interaction potentials generated by adapting the Collision Verlet method within both the Nosé–Hoover and Nosé–Poincaré schemes. In the next section we briefly review the standard Collision Verlet algorithm[2] followed by the introduction of the Nosé–Hoover Collision Verlet (NHCV) and Nosé–Poincaré Collision Verlet (NPCV) algorithms in Secs. III and IV, respectively. The algorithms are evaluated in Sec. V through numerical experiments on a model system. In Sec. VI, we conclude.

II. THE COLLISION VERLET ALGORITHM

In this section we review the Collision Verlet[2] algorithm for the numerical integration of the dynamics of systems with mixed continuous/discontinuous interaction potentials. We consider N particles interacting through a continuous potential plus a hard core, assumed here to be spherical. To facilitate the construction of numerical methods, it is useful to describe the dynamics of the system within a Hamiltonian format, but for a system with a discontinuous potential the construction of a Hamiltonian as the generator of the dynamical equations of motion is problematic. In this work, we observe that the hard-sphere interaction potential, $V_{\mathrm{hs}}(\{\mathbf{q}\})$, typically can be approximated to any degree of accuracy by a sequence of steeply repulsive continuous functions. In this sense, the energy function $\mathcal{H}$ of the mixed system is refered to here as a pseudo-Hamiltonian. Here the pseudo-Hamiltonian is given by

$$\mathcal{H}=T(\mathbf{p})+V_{\mathrm{hs}}(\{\mathbf{q}\})+V_c(\{\mathbf{q}\}), \tag{2}$$

where the kinetic energy $T(\mathbf{p})=\Sigma_i(p_i^2/2m_i)$, $V_{\mathrm{hs}}(\{\mathbf{q}\})$ is the hard sphere potential and $V_c(\{\mathbf{q}\})$ is a continuously differentiable potential energy function, that we assume to be pairwise additive, that is,

$$V_c(\{\mathbf{q}\})=\sum_i \sum_{j>i} v_c(q_{ij}),$$

where v_c is a pair potential, q_{ij} is the distance between two particles indexed by i and j, and the sum is over all pairs of particles.

The Collision Verlet algorithm is based on the splitting of the continuous pair potential, $v_c(q)$, into a short range part, $v_1(q)$, and a long range part, $v_2(q)$, according to

$$v_c(q)=v_1(q)+v_2(q). \tag{3}$$

The potential splitting is rendered so that the force due to the long-range part of the potential vanishes at the hard-sphere contact distance [i.e., $v_2'(\sigma)=0$]. This form of the potential splitting is necessary for the construction of a second-order method. For the motivation and specific details of this splitting technique the reader is referred to Ref. 2. The pseudo-Hamiltonian is then split accordingly. For generality, consider $\mathcal{H}$ as a pseudo-Hamiltonian of any given mixed impulsive-continuous system. Next, we partition $\mathcal{H}$ in the following way:

$$\mathcal{H}=\mathcal{H}_1+\mathcal{H}_2, \tag{4}$$

where $\mathcal{H}_1$ includes the kinetic energy, the hard sphere potential, V_{hs}, and the short range potential, V_1; $\mathcal{H}_2$ must include the long range potential, V_2. A Trotter factorization[6] then gives the following approximation for the dynamical flow map, $\phi_{\mathcal{H}}(\tau)$, defined as the operator (associated with the Hamiltonian $\mathcal{H}$) that advances the phase space configuration time τ into the future,

$$\phi_{\mathcal{H}}(\tau)=\phi_{\mathcal{H}_2}\left(\frac{\tau}{2}\right)\phi_{\mathcal{H}_1}(\tau)\phi_{\mathcal{H}_2}\left(\frac{\tau}{2}\right). \tag{5}$$

Since $\mathcal{H}_2$ only contains the long-range potential, the flow map $\phi_{\mathcal{H}_2}$ can be constructed exactly. The flow map corresponding to $\mathcal{H}_1$ is approximated in the following way:

$$\phi_{\mathcal{H}_1}\approx\phi_{T+V_1}(\tau^{(c)}_{n_c+1})\prod_{i=1}^{n_c}[\phi_{V_{\mathrm{hs}}}\phi_{T+V_1}(\tau^{(c)}_{n_c+1-i})], \tag{6}$$

where n_c is the number of hard-sphere collisions during the time step h, $\tau_i^{(c)}$ is the time between each collision [with $\tau_1^{(c)}$ being measured from the beginning of the time step until the first collision and $\tau^{(c)}_{n_c+1}$ measured from the last collision to the end of the time step so that $\Sigma_{i=1}^{n_c+1}\tau_c^i=\tau$], and $\phi_{V_{hs}}$ is the flow map for an instantaneous hard-sphere collision. Finally, the flow map for the motion of the particle between collisions is approximated using the Stömer–Verlet algorithm generated by a further Trotter factorization,

$$\phi_{T+V_1}(\tau)\approx\phi_{V_1}\left(\frac{\tau}{2}\right)\phi_T(\tau)\phi_{V_1}\left(\frac{\tau}{2}\right). \tag{7}$$

The most CPU intensive part of the Collision Verlet algorithm is the determination of the time to next collision τ_c. The collision condition for two particles i and j can be written as

$$\|\mathbf{q}_i(\tau_c)-\mathbf{q}_j(\tau_c)\|^2-\sigma^2=0. \tag{8}$$

Since the trajectories between collisions are approximated within the Collision Verlet scheme by quadratic equations, the collision condition (8) is a quartic equation. To ensure that all collisions are resolved correctly, it is necessary to accurately resolve the smallest positive root to this quartic equation. This is not a trivial problem as the root becomes increasingly ill conditioned as smaller time steps are used (i.e., when the time to collision is small). To increase efficiency and accuracy of the computation, we employed in all the simulations in this paper a root finding method based on Cauchy indices.[8] The details of the collision-time calculation are given in the Appendix.

III. COLLISION VERLET WITH A NOSÉ–HOOVER THERMOSTAT

As discussed in the Introduction, the Nosé–Hoover method for isothermal molecular-dynamics simulation is generated by applying time and coordinate transformations to the equations of motion generated by the Nosé Hamiltonian [Eq. (1)], which are

$$\frac{dq_i}{d\tau}=\frac{\tilde{p}_i}{m_i s^2}, \quad \frac{ds}{d\tau}=\frac{\tau}{Q}, \tag{9}$$

$$\frac{d\tilde{p}_i}{d\tau}=-\frac{\partial}{\partial q_i}V_c(q), \quad \frac{d\pi}{d\tau}=\sum_i \frac{\tilde{p}_i^2}{m_i s^3}-\frac{gkT}{s}. \tag{10}$$

Conversion to real time, t, is accomplished through the following transformations:

$$\mathbf{p}=\frac{\tilde{\mathbf{p}}}{s}, \quad \frac{d\tau}{dt}=s. \tag{11}$$

In addition, Hoover simplified the resulting equations of motion by introducing a further variable transformation

$$\eta=\ln s \quad \xi=\dot{\eta}, \tag{12}$$

resulting in the so-called Nosé–Hoover equations of motion,

$$\dot{q}_i=\frac{p_i}{m_i}, \quad \dot{p}_i=-\frac{\partial}{\partial q_i}V(q)-p_i\xi, \tag{13}$$

$$\dot{\eta}=\xi, \quad \dot{\xi}=\frac{1}{Q}\left(\sum_i \frac{p_i^2}{m_i}-gkT\right). \tag{14}$$

These equations of motion can be shown to generate configurations distributed according to an isothermal (canonical) distribution as long as the system is ergodic and $g=N_f$, where N_f is the number of degrees of freedom. Since the coordinate transformation is noncanonical, the equations of motion are not generated by a Hamiltonian; however, a conserved energy does exist and is given by

$$E=\sum_i \frac{p_i^2}{2m_i}+V(q)+\frac{1}{2}Q\xi^2+gkT\eta. \tag{15}$$

In order to simplify the construction of splitting methods for this non-Hamiltonian system and to make contact with the earlier literature, we write the flow map in terms of a Liouville operator, $\mathcal{L}$, as follows:

$$\phi(\tau)=e^{\mathcal{L}}. \tag{16}$$

The Liouville operator corresponding to the Nosé–Hoover equations of motion above is

$$\mathcal{L}=\sum_i \frac{p_i}{m_i}\frac{\partial}{\partial q_i}+\mathcal{L}_{\text{hs}}-\sum_i p_i\xi\frac{\partial}{\partial p_i}-\sum_i \frac{\partial}{\partial q_i}V(q)\frac{\partial}{\partial p_i} +\xi\frac{\partial}{\partial\eta}+\frac{1}{Q}\left(\sum_i \frac{p_i^2}{m_i}-gkT\right)\frac{\partial}{\partial\xi}, \tag{17}$$

where we have explicitly included a hard-sphere term, $\mathcal{L}_{\text{hs}}$.

To get a reversible method for the Nosé–Hoover method with mixed potentials, the above Liouville operator is split in the following way:

$$\mathcal{L}=\mathcal{L}_1+\mathcal{L}_2+\mathcal{L}_3, \tag{18}$$

with

$$\mathcal{L}_1=\mathcal{L}_{\text{hs}}+\sum_i \frac{p_i}{m_i}\frac{\partial}{\partial q_i}-\sum_i \frac{\partial}{\partial q_i}V_1(q)\frac{\partial}{\partial p_i}, \tag{19}$$

$$\mathcal{L}_2=-\frac{\partial}{\partial q_i}V_2(q)\frac{\partial}{\partial p_i}, \tag{20}$$

and

$$\mathcal{L}_3=-\sum_i p_i\xi\frac{\partial}{\partial p_i}+\frac{1}{Q}\left(\sum_i \frac{p_i^2}{m_i}-gkT\right)\frac{\partial}{\partial\xi}+\xi\frac{\partial}{\partial\eta}. \tag{21}$$

A Trotter factorization is now applied to this splitting,

$$e^{\mathcal{L}\tau}=e^{\mathcal{L}_3\tau/2}e^{\mathcal{L}_2\tau/2}e^{\mathcal{L}_1\tau}e^{\mathcal{L}_2\tau/2}e^{\mathcal{L}_3\tau/2}+\mathcal{O}(\tau^3). \tag{22}$$

The operator $e^{\mathcal{L}_1\tau}$ is approximated using the Collision Verlet method described in the previous section [see Eq. (6)]. The solution of the operator $e^{\mathcal{L}_2\tau/2}$ is straightforward. To find the solution of the operator $e^{\mathcal{L}_3\tau/2}$, i.e.,

$$\begin{pmatrix} q_{i,n+1} \\ p_{i,n+1} \\ \eta_{n+1} \\ \xi_{n+1} \end{pmatrix}=e^{\mathcal{L}_3\tau/2}\begin{pmatrix} q_{i,n} \\ p_{i,n} \\ \eta_n \\ \xi_n \end{pmatrix}, \tag{23}$$

we further split $\mathcal{L}_3$. That is,

$$\mathcal{L}_3=\mathcal{L}_3^{(1)}+\mathcal{L}_3^{(2)}, \tag{24}$$

with

$$\mathcal{L}_3^{(1)}=-\sum_i p_i\xi\frac{\partial}{\partial p_i}+\xi\frac{\partial}{\partial\eta}, \tag{25}$$

and

$$\mathcal{L}_3^{(2)}=\frac{1}{Q}\left(\sum_i \frac{p_i^2}{m_i}-gkT\right)\frac{\partial}{\partial\xi}. \tag{26}$$

The corresponding Trotter factorization of this splitting is

$$e^{\mathcal{L}_3\tau}\approx e^{\mathcal{L}_3^{(2)}\tau/2}e^{\mathcal{L}_3^{(1)}\tau}e^{\mathcal{L}_3^{(2)}\tau/2}. \tag{27}$$

The solution of the operator $e^{\mathcal{L}_3^{(2)}\tau/2}$ is straightforward. The operator $e^{\mathcal{L}_3^{(1)}\tau}$ is solve from a further splitting. The solution of the operator $e^{\mathcal{L}_3\tau/2}$ gives

$$\xi_{n+1/2}=\xi_n+\frac{\tau}{4Q}\left(\sum_i \frac{(p_{i,n})^2}{m_i}-gkT\right), \tag{28}$$

$$\eta_{n+1}=\eta_n+\frac{\tau}{2}\xi_{n+1/2}, \tag{29}$$

$$p_{i,n+1}=p_{i,n}\frac{1-\tau\xi_{n+1/2}/4}{1+\tau\xi_{n+1/2}/4}, \tag{30}$$

$$\xi_{n+1}=\xi_{n+1/2}+\frac{\tau}{4Q}\left(\sum_i \frac{(p_{i,n+1})^2}{m_i}-gkT\right). \tag{31}$$

The algorithm is tested in Sec. V for a system of hard spheres with inverse-sixth-power attractive tails. Certainly, the Liouville operator splitting used above is not the only possible method. For example, another splitting is

$$\mathcal{L}=\mathcal{L}_1+\mathcal{L}_2, \tag{32}$$

with

$$\mathcal{L}_1=\sum_i \frac{p_i}{m_i}\frac{\partial}{\partial q_i}+\mathcal{L}_{\text{hs}}-\sum_i \frac{\partial}{\partial q_i}V_1(q)\frac{\partial}{\partial p_i}, \tag{33}$$

and

$$\mathcal{L}_2=-\sum_i p_i\xi\frac{\partial}{\partial p_i}-\frac{\partial}{\partial q_i}V^2(q)\frac{\partial}{\partial p_i}+\frac{1}{Q}\left(\sum_i \frac{p_i^2}{m_i}-gkT\right)\frac{\partial}{\partial \xi}+\xi\frac{\partial}{\partial \eta} \tag{34}$$

can be used. Using a Trotter factorization gives

$$e^{\mathcal{L}\tau}\approx e^{\mathcal{L}_2\tau/2}e^{\mathcal{L}_1\tau}e^{\mathcal{L}_2\tau/2}. \tag{35}$$

IV. COLLISION VERLET WITH A NOSÉ–POINCARÉ THERMOSTAT

The Nosé–Hoover formulation of constant-temperature dynamics is non-Hamiltonian in structure, thereby preventing the use of symplectic integration schemes, which, for systems with continuous potentials, can be shown to enhance long-term stability.[6] Recently, Bond, Leimkuhler, and Laird[7] proposed a new real-time, but fully Hamiltonian, formulation of the Nosé constant-temperature dynamics. This is accomplished by performing a time transformation, not to the Nosé equations of motion as with Nosé–Hoover, but directly to the Hamiltonian using a Poincaré time transformation, as follows:

$$\mathcal{H}_{\text{NP}}=s(\mathcal{H}_{\text{Nosé}}-\mathcal{H}_0), \tag{36}$$

where H_0 is the initial value of $\mathcal{H}_{\text{Nosé}}$. Combining Eqs. (1) and (36) the Nosé–Poincaré thermostat Hamiltonian of a physical system consisting of N particles is expressed as the following:

$$\mathcal{H}_{\text{NP}}=s\left(\sum_i \frac{\tilde{p}_i^2}{2m_i s^2}+V_c(q)+\frac{\pi^2}{2Q}+gkT\ln s-\mathcal{H}_0\right). \tag{37}$$

In order to sample the correct canonical distribution, the constant g is taken to be the number of degrees of freedom,[7] $g=N_f$. The equations of motion are

$$\dot{q}_i=\frac{\tilde{p}_i}{m_i s},\quad \dot{s}=s\frac{\pi}{Q}, \tag{38}$$

$$\dot{\tilde{p}}_i=-s\frac{\partial}{\partial q_i}V_c(q),\quad \dot{\pi}=\sum_i \frac{\tilde{p}_i^2}{m_i s^2}-gkT-\Delta\mathcal{H}, \tag{39}$$

$$\Delta\mathcal{H}=\sum_i \frac{\tilde{p}_i^2}{2m_i s^2}+V_c(q)+\frac{\pi^2}{2Q}+gkT\ln s-\mathcal{H}_0. \tag{40}$$

Note that, when solved exactly, the Nosé-Poincaré and Nosé-Hoover equations generate identical trajectories. It is in the construction of approximate numerical methods that these two approaches differ.

For the present case, we write the Nosé–Poincaré thermostat pseudo-Hamiltonian (see Sec. II) for a mixed hard-core/continuous potentials system,

$$\mathcal{H}_{\text{NP}}=s\left(\sum_i \frac{\tilde{p}_i^2}{2m_i s^2}+V_{\text{hs}}(q)+V_c(q)+\frac{\pi^2}{2Q}+gkT\ln s-\mathcal{H}_0\right). \tag{41}$$

There are a variety of ways in which one can construct numerical integration algorithms using this Hamiltonian. To this end, we first consider two ways of splitting the overal NP Hamiltonian:
Splitting I:

$$\mathcal{H}_1=s\left(\sum_i \frac{\tilde{p}_i^2}{2m_i s^2}+V_{\text{hs}}(q)+V_1(q)+gkT\ln s-\mathcal{H}_0\right), \tag{42}$$

$$\mathcal{H}_2=s\left(V_2(q)+\frac{\pi^2}{2Q}\right), \tag{43}$$

Splitting II:

$$\mathcal{H}_1=s\left(\sum_i \frac{\tilde{p}_i^2}{2m_i s^2}+V_{\text{hs}}(q)+V_1(q)-\mathcal{H}_0\right), \tag{44}$$

$$\mathcal{H}_2=s\left(V_2(q)+\frac{\pi^2}{2Q}+gkT\ln s\right). \tag{45}$$

A Trotter factorization of the flow map [Eq. (5)] is applied to each splitting. To approximate the flow map generated by $\mathcal{H}_1$, we employ the Collision Verlet scheme given in Eq. (6) to integrate the system from collision to collision under the influence of the short-range potential. Since s is a constant in the dynamics generated by $\mathcal{H}_1$ in both splittings, the Störmer–Verlet algorithm can be used to integrate the trajectory between collisions, with the collision time being calculated as described in the Appendix. For splitting I, Störmer–Verlet gives

$$\tilde{p}_{i,n+1/2}=\tilde{p}_{i,n+1/2}-\frac{\tau}{2}s_n\frac{\partial}{\partial q_i}V_1(q_n), \tag{46}$$

$$\pi_{n+1/2}=\pi_{n+1/2}+\frac{\tau}{2}\left[\sum_i \frac{1}{m_i}\left(\frac{\tilde{p}_{i,n+1/2}}{s_n}\right)^2-\Delta H(q_n,\tilde{p}_{i,n+1/2},s_n)\right], \tag{47}$$

$$q_{i,n+1}=q_{i,n}+\tau\frac{\tilde{p}_{i,n+1/2}}{m_i s_n}, \tag{48}$$

$$\pi_{n+1}=\pi_{n+1/2}+\frac{\tau}{2}\left[\sum_i \frac{1}{m_i}\left(\frac{\tilde{p}_{i,n+1/2}}{s_n}\right)^2-\Delta H(q_{n+1},\tilde{p}_{i,n+1/2},s_n)\right], \tag{49}$$

$$\tilde{p}_{i,n+1}=\tilde{p}_{i,n+1/2}-\frac{\tau}{2}s_n\frac{\partial}{\partial q_i}V_1(q_{n+1}). \tag{50}$$

The equations for splitting II can be similarly generated.

In both splittings I and II the integration of $\mathcal{H}_2$ is complicated by the presence of both s and its conjugate momentum π, but here we consider two possible approaches:

$\mathcal{H}_2$ *Integration Method 1*: Since the Hamiltonian here is nonseparable, the Generalized Leapfrog[7,9,10] scheme, a fully symplectic extension of the Störmer–Verlet algorithm for nonseparable Hamiltonians, can be used. The resulting integration method for splitting I with time step τ is

$$\tilde{p}_{i,n+1/2}=\tilde{p}_{i,n}-\frac{\tau}{2}s_n\frac{\partial}{\partial q_i}V_2(q_n), \tag{51}$$

$$\pi_{n+1/2}=\pi_{s,n}-\frac{\tau}{2}(gkT+\Delta H_2(q_n,s_n,\pi_{n+1/2})), \tag{52}$$

$$s_{n+1}=s_n+\frac{\tau}{2}(s_n+s_{n+1})\frac{\pi_{n+1/2}}{Q}, \tag{53}$$

$$\pi_{n+1}=\pi_{n+1/2}-\frac{\tau}{2}(gkT+\Delta H_2(q_n,s_{n+1},\pi_{n+1/2})), \tag{54}$$

$$\tilde{p}_{i,n+1}=\tilde{p}_{i,n+1/2}-\frac{\tau}{2}s_{n+1}\frac{\partial}{\partial q_i}V_2(q_n). \tag{55}$$

The integrator given above is explicit. Equation (52) requires the solution of a scalar quadratic equation for $\pi_{n+1/2}$. Details of how to solve this equation without involving subtractive cancellation can be found in Ref. 7. The application of method 1 for the $\mathcal{H}_2$ in splitting II is similar and straightforward.

$\mathcal{H}_2$ *Integration Method 2*: Instead of using Generalized Leapfrog, we employ a splitting of $\mathcal{H}_2$,

$$\mathcal{H}_2=\mathcal{H}_2^{(1)}+\mathcal{H}_2^{(2)}. \tag{56}$$

For splitting I, we use

$$\mathcal{H}_s^{(1)}=\frac{s\pi^2}{2Q}, \tag{57}$$

$$\mathcal{H}_2^{(2)}=sV_2(q). \tag{58}$$

Since no conjugate pair appears in $\mathcal{H}_2^{(2)}$, its dynamics for a time step τ is exactly solvable,

$$\tilde{p}_{i,n+1}=\tilde{p}_{i,n}-\tau s_n\frac{\partial}{\partial q_i}V_2(q_n), \tag{59}$$

$$\pi_{n+1}=\pi_n-\tau V_2(q_n). \tag{60}$$

Only equations involving variables p and π are shown above because q and s are constants of motion. The solution of the dynamics of $\mathcal{H}_2^{(1)}$ involves a conjugate pair s and π, but it can be solved exactly.[11] Thus the time evolution of $\mathcal{H}_2^{(1)}$ for the time step τ is

$$s_{n+1}=s_n\left(1+\frac{\pi_n}{2Q}\tau\right)^2, \tag{61}$$

$$\pi_{n+1}=\frac{\pi_n}{1+\frac{\pi_n}{2Q}\tau}. \tag{62}$$

Here, it is q and $\tilde{p}$ that are constants of motion. Again, the application of method 2 for splitting II is similar and straightforward.

Combining the two overall splittings for the NP Hamiltonian with the two methods for integrating $\mathcal{H}_2$, gives a total of four proposed algorithms for the Nosé–Poincaré Collision-Verlet (NPCV) method. These are:

(1) NPCV1: Splitting I+$\mathcal{H}_2$ integration method 1;
(2) NPCV2: Splitting I+$\mathcal{H}_2$ integration method 2;
(3) NPCV3: Splitting II+$\mathcal{H}_2$ integration method 1;
(4) NPCV4: Splitting II+$\mathcal{H}_2$ integration method 2.

In the next section we test these four algorithms for a model system and compare them with each other and with the Nosé–Hoover Collision Verlet (NHCV) method outlined in the previous section.

V. NUMERICAL EXPERIMENTS ON A MODEL POTENTIAL

We test the various NVT Collision Verlet algorithms proposed in this paper using a system of hard spheres with an attractive inverse-sixth-power continuous potential,

$$v_c=-\epsilon\left(\frac{\sigma}{q}\right)^6, \tag{63}$$

where σ is the hard-sphere diameter. The potential is truncated at the distance $q_c=2.5\sigma$ and, to ensure its continuity, it is shifted and smoothed so that potential and the force vanish beyond the cutoff distance. We split the above potential into short and long-range parts, as prescribed in Eqs. (19)–(25) of Ref. 2. The values of input parameters q_1 and q_2 are 1.3σ and 1.5σ, respectively.

The MD simulations were carried out on systems of $N=500$ particles. A system of reduced units was chosen so that all quantities are dimensionless: as units of distance and energy we used the potential parameters σ and ϵ, respectively, and the mass of one atom as the unit mass. The unit of time is $(m\sigma^2/\epsilon)^{1/2}$. An asterisk superscript indicates reduced units. In all simulations the density was $\rho^*=\rho\sigma^3=0.7$ with reduced temperature $T^*=kT/\epsilon=1.5$. In addition, a cubic box with periodic boundary conditions was used. In order to improve efficiency, neighbor (Verlet) lists[1] were used for the evaluation of the short range force, the long range force, and the collision times. In all of our simulations, we set $g=N_f$ with $N_f=3(N-1)$ to correct for the fact that in a molecular-dynamics simulation the total linear momentum is conserved.[12] Each run was started from an initial configuration produced after an equilibration run of 200 000 time steps (with $\tau^*=0.001$) starting from a face centered cube (fcc) lattice with the particle velocities chosen from a Boltzmann distribution at $T^*=1.5$. The initial values of the extended variables in all of the numerical experiments are set to be $s_0=1$ and $p_{s,0}=0$ in the case of the Nosé–Poincaré thermo-

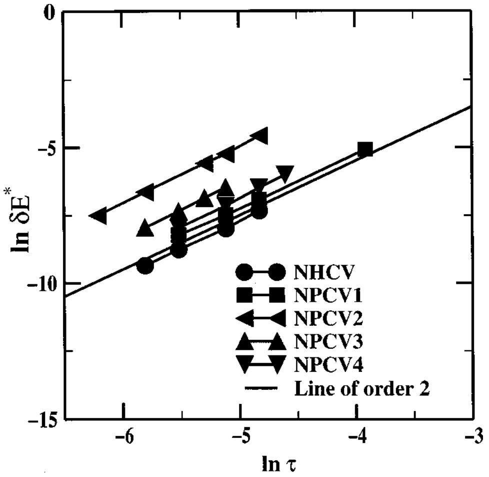

FIG. 1. Order of accuracy of the NHCV algorithm and NPCV algorithms 1–4. Comparison is made with a line of order 2.

stat methods. In the case of the Nosé–Hoover method, the initial values of the extended variables are thus $\eta_0 = 0$ and $\xi_0 = 0$.

In order to compare the short time accuracy of the methods and verify that each one exhibits second-order global error, we show in Fig. 1 a log–log plot of the maximum energy error for a run of total length $t^* = 12$ for each method as a function of time step, τ. For comparison, a line of slope 2 is plotted to show that the global error for each method is second order, as required. In these runs the thermostat mass Q was set to 1.0. Note that, due to the discontinuous nature of the dynamics, the second order global error is not simply a consequence of the time-reversibility of the algorithms, but it also a direct result of the particular potential splitting we have chosen.[2] From Fig. 1 we see that for short runs, the Nosé–Hoover based method has the smallest error constant.

For molecular-dynamics simulation the stability during long runs is more important that the order or short-term behavior of the algorithm. To test this we plot the energy trajectory, $\delta E = E(t) - E(t=0)$, versus time for each of our methods using 10^6 time steps of length $\tau^* = 5 \times 10^{-3}$ (total time 5000). Figure 2 shows this plot for each of the four Nosé–Poincaré based methods discussed in the previous section. For this system, NPCV methods 2 and 3 exhibit significant drift, whereas methods 1 and 4 are more stable for long time trajectories. The same plot for the Nosé–Hoover method presented in Sec. III is shown in Fig. 3 with the plot for the NPCV method 1 shown for comparison. The NPCV method 1 has slightly better energy conservation for this system than the Nosé–Hoover Collision Verlet algorithm, which is comparable to the NPCV method 4, but the differences are small and could change depending on the system.

The algorithms presented here are designed to give a canonical distribution of phase space points. A useful check of this is to examine the distribution of instantaneous temperature (as defined for a system with zero total momentum)

$$\hat{T} = \frac{2}{3(N-1)} \sum_i^N \frac{p_i^2}{2m}. \tag{64}$$

A canonical distribution in momenta requires that this quatitiy be Gaussian distributed about the target temperature

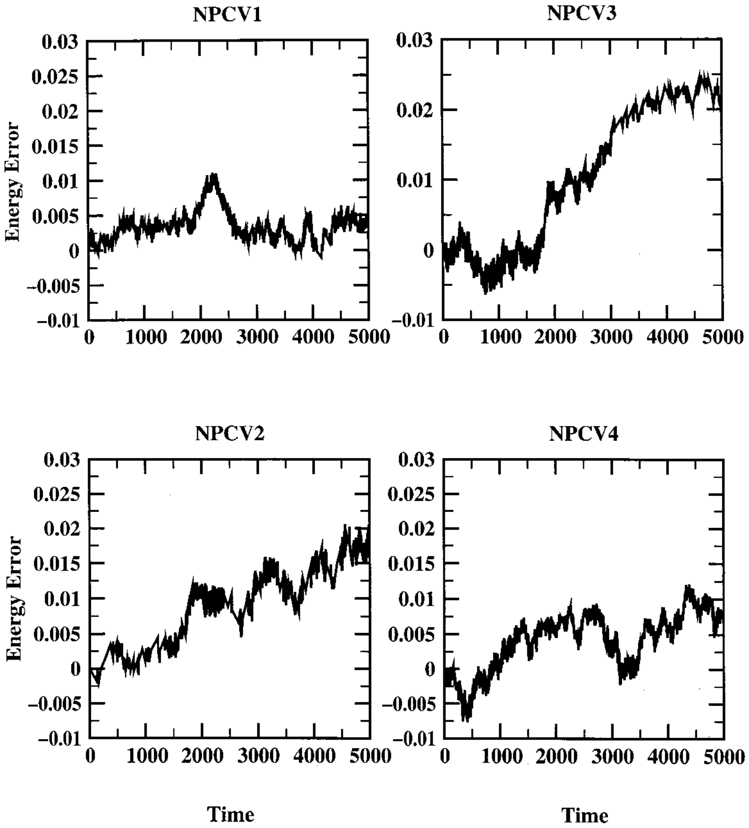

FIG. 2. Energy conservation in a long simulation run (10^6 time steps) for NPCV algorithms 1–4.

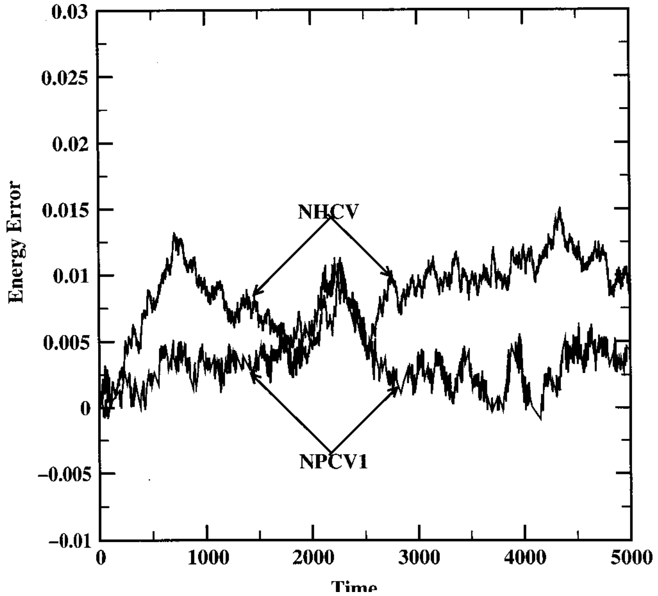

FIG. 3. Energy vs time in a long simulation run (10^6 time steps) using the NHCV and NPCV1 algorithms.

T with a variance of $2T^2/[3(N-1)]$. In Fig. 4 is plotted the temperature distributions for the 4 NPCV algorithms using a thermostat mass of 10 measured during runs of 270 000 time steps ($\tau^*=5\times10^{-3}$) after equilibration. Figure 5 shows the same quantity for the Nosé–Hoover Collision Verlet method. Comparison with the theoretical distribution, shown as a solid line in each plot, indicates that the canonical distribution is well reproduced by all proposed algorithms.

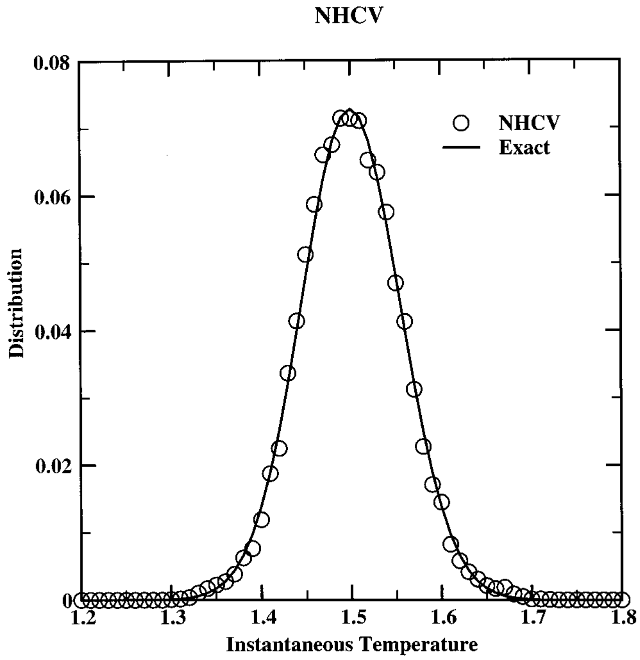

FIG. 5. Instantaneous temperature distribution for the NHCV simulations (circles). The exact canonical distribution is shown as a solid line.

VI. CONCLUSION

In this work we have developed several algorithms, based on the extended Hamiltonian thermostat of Nosé, to perform constant temperature (NVT) molecular-dynamics simulations of systems with mixed hard-core/continuous po-

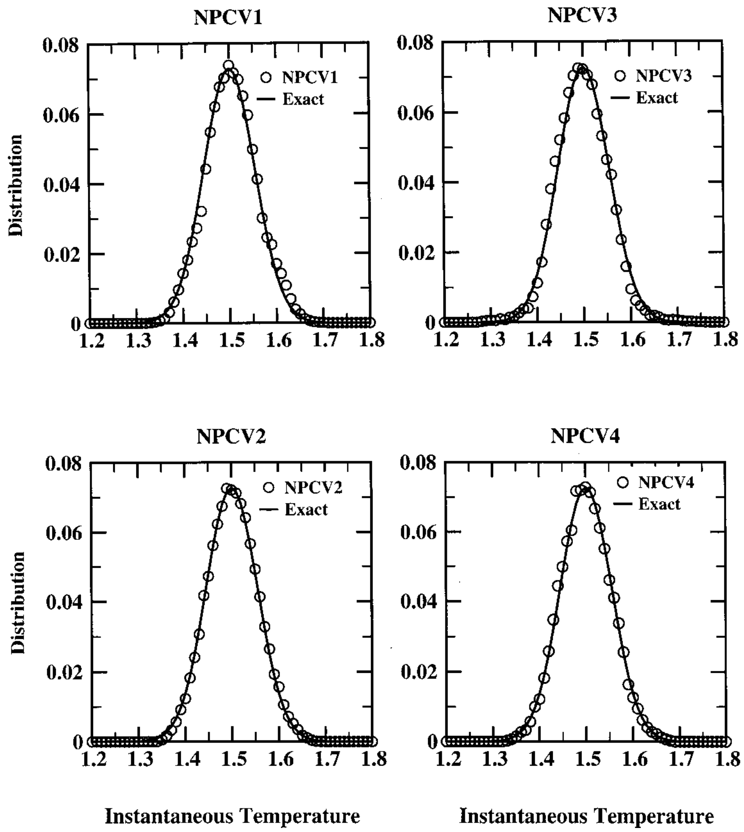

FIG. 4. Instantaneous temperature distributions for the NPCV algorithms 1–4. In each, the exact canonical distribution is shown as a solid line.

tentials. The methods are extentions of our recently developed Collision Verlet method[2] for constant energy (NVE) MD simulation. These new methods, to our knowledge, represent the first viable canonical molecular-dynamics simulation methods for hybrid discontinous/continuous potentials.

Specifically, five new algorithms have been presented and tested. The first algorithm, the Nosé–Hoover Collision Verlet (NHCV) algorithm, is based on application of the Nosé–Hoover thermostat[5] to the Collision Verlet scheme. The other four algorithms presented are based on the Nosé–Poincare formulation of real-time Nosé dynamics. These Nosé–Poincaré Collision Verlet methods differ from one another in the details of the numerical scheme used to integrate the equations of motion. All methods were shown to give second-order global error in test simulations with the NHCV method having the smallest error constant for short-time simulations. The NHCV algorithm and two of the presented NPCV algorithms (NPCV1 and NPCV4) were found to exhibit good stability in long time simulations involving 500 hard-sphere particles with attractive inverse-sixth-power tails. In addition, all methods were shown to correctly reproduce the canonical distribution of instantaneous temperature (kinetic energy). Note that, if the continuous potential is set to zero, the presented methods also provide a way of performing canonical, as opposed to isokinetic, hard-sphere molecular-dynamics simulations.

APPENDIX: CALCULATION OF TIME TO NEXT COLLISION

In this Appendix we address the issue of the collision time calculation for mixed hard-core/continuous potentials systems. The quartic equation for the collision condition [Eq. (8)] is solved for all pairs of particles and the smallest positive root is located as the time to the next collision. For mixed hard-core/continuous potentials systems, this is a time-consuming operation since collision times for all pairs must be recalculated after each collision. In addition, Eq. (8) is quartic and difficult to solve. As we said in Sec. II, the quartic equation must be solved accurately to give the smallest positive root in a given interval in order to ensure that no collisions are missed.

In Ref. 2, we employed Laguerre's method[13] for collision time calculation for mixed hard-core/continuous potentials systems. The method is sufficient for all but the very smallest times steps studied. But the method turns out to be very slow. This because for any given time interval and pair of particles, all four complex roots need to be calculated. Also Laguerre's method deals with complex arithmetic. In this Appendix, we propose a time saving collision-time calculation method for Collision Verlet. This method is based on a Cauchy indices of a Sturm sequence[8] of a real polynomial in a real interval.

The Cauchy index is an integer that can be associated with any real rational function and any interval whose end points are not the poles of the function. Let r be a rational function. The Cauchy index, $I_\alpha^\beta r(x)$, of r for the interval $[\alpha, \beta]$ is by definition the number of jumps of the function r from $+\infty$ to $-\infty$ on the interval $[\alpha, \beta]$. The Cauchy index can be calculated for any real polynomial that forms a Sturm sequence, $\{f_0, f_1, \ldots, f_m\}$, for the interval $[\alpha, \beta]$. The definition of the Sturm sequence of a real polynomials can be found in Ref. 8. The connection between the Cauchy index and the number of sign changes, $v(x)$ for arbitrary real x, in the numerical sequence, $\{f_0, f_1, \ldots, f_m\}$, is given by the following result due to Sturm.[14]

Theorem 1: *Let the real polynomials, $\{f_0, f_1, \ldots, f_m\}$ form a Sturm sequence for the interval $[\alpha, \beta]$, $\alpha \leqslant \beta$. Then,*

$$I_\alpha^\beta \frac{f_1}{f_0} = v(\alpha) - v(\beta). \tag{A1}$$

Using this theorem we can write the number of real roots for a given polynomial p in any real interval $[\alpha, \beta]$ in terms of the Cauchy index,

$$I_\alpha^\beta \frac{p'}{p_0} = v(\alpha) - v(\beta) \tag{A2}$$

of the sequence $\{p_k\}$, generated by the Euclidean algorithm[8] using the starting polynomials $p_0 := p$, $p_1 := p'$, with p' being the first derivative of the polynomial p. The elements of the rest of the sequence are linked by the relations

$$p_0(x) = q_1(x) p_1(x) - p_2(x), \tag{A3}$$

$$p_1(x) = q_2(x) p_2(x) - p_3(x), \tag{A4}$$

$$\vdots \tag{A5}$$

$$p_{k-1}(x) = q_k(x) p_k(x) - p_{k+1}(x),$$

$$\vdots \tag{A6}$$

$$p_{m-1}(x) = q_m(x) p_m(x).$$

The Euclidean algorithm also furnishes information about the multiplicity of the zeros. x_0 is a zero of multiplicity k of p if and only if it is a zero of multiplicity $k-1$ of p_m. We are now able to develop a collision time calculation method for Collision Verlet.

From the above, the first step for Collision Verlet collision time calculation is to determine in a given time interval the number of real roots by calculating the Cauchy index for the time interval. This means that we need an algorithm for polynomial division. The main problem with polynomials division is that the bit length of coefficients in the sequence can increase dramatically and also, because we are dividing, in some cases the denominator can vanish. To solve this problem, we use the Sturm–Habicht pseudodivisions subresultant (PRS) method.[15] The members of the polynomial remainder sequence $p_1(x), p_2(x), p_3(x), \ldots, p_h(x)$,

$$l_c[p_{i+1}(x)]^{n_i - n_{i+1} + 1} p_i(x) = p_{i+1}(x) q_i(x) - \beta_i p_{i+2}(x), \tag{A7}$$

$$\deg[p_{i+2}(x)] \leqslant \deg[p_{i+1}(x)], \tag{A8}$$

where $i = 1,2,\ldots,h-1$, for some h, $n_i = \deg[p_i(x)]$, and $l_c[p_i(x)]$ is the leading coefficient of p_i. The different values of β_i are

$$\beta_1=(-1)^{n_1-n_2+1},\quad \beta_i=(-1)^{n_i-n_{i+1}+1}l_c[p_i(x)] \cdot H_i^{n_i-n_{i+1}},\quad i=2,3,...,h-1, \tag{A9}$$

$$H_2=\{l_c[p_2'(x)]\}^{n_1-n_2}, \tag{A10}$$

$$H_i=\{l_c[p_i(x)]\}^{n_{i-1}-n_i}H_{i-1}^{1-(n_{i-1}-n_i)},\quad i=3,...,h-1. \tag{A11}$$

Let

$$p(x)=ax^4+bx^3+cx^2+dx+e, \tag{A12}$$

be the quartic polynomial obtained from the collision condition of Eq. (8), and $\{p_1,p_2,p_3,p_4,p_5\}$ its Sturm–Habitch sequence determined by using Eq. (71). We now determine the number of real roots of the equation $p(t)=0$ in a given time interval by calculating its Cauchy index, Eq. (66). If there is only one root, then we use the Newton–Raphson method[13] to approximate the root. If there is more than one root, then we combine the bisection method[13] and root counting method to isolate the time interval containing the smallest root.

This method for solving for the shortest collision time is quite efficient giving a factor of 20 speed-up from our previous simulations using the Laguerre method,[2] primarily because we no longer calculate all four roots of the quartic equation and avoid complex arithmetic.

[1] M. A. Allen and D. J. Tildesley, *Computer Simulation of Liquids* (Oxford Science Press, Oxford, 1987).

[2] Y. A. Houndonougbo, B. B. Laird, and B. J. Leimkuhler, Mol. Phys. **98**, 309 (2000).

[3] S. Nosé, Mol. Phys. **52**, 255 (1984).

[4] S. Nosé, J. Chem. Phys. **81**, 511 (1984).

[5] W. G. Hoover, Phys. Rev. A **31**, 1695 (1985).

[6] J. M. Sanz-Serna and M. P. Calvo, *Numerical Hamiltonian Problems* (Chapman and Hall, New York, 1995).

[7] S. D. Bond, B. J. Leimkuhler, and B. B. Laird, J. Comput. Phys. **151**, 114 (1999).

[8] P. Henrici, *Applied and Computational Complex Analysis* (Wiley, New York, 1974)

[9] E. Hairer, ''Backward error analysis of numerical integrators and symplectic methods,'' *Annals of Numerical Mathematics*.

[10] G. Sun, J. Comput. Math. **11**, 365 (1993).

[11] S. Nosé, J. Phys. Soc. Jpn. **70**, 75 (2001).

[12] T. Cagin and J. R. Ray, Phys. Rev. A **37**, 4510 (1988).

[13] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipies in Fortran* (Cambridge University Press, New York, 1992).

[14] Ch. F. Sturm, Mém. Savants. E'trang.-Paris **6**, 271 (1835).

[15] A. G. Akritas, *Elements of Computer Algebra with Applications* (Wiley, New York, 1989).