Efficient multiple-time-step integrators with distance-based force splitting for particle-mesh-Ewald molecular dynamics simulations

Xiaoliang Qian and Tamar Schlick[a)]
Department of Chemistry and Courant Institute of Mathematical Sciences, New York University, and Howard Hughes Medical Institute, New York, New York 10012

(Received 3 October 2001; accepted 16 January 2002)

We develop an efficient multiple-time-step force splitting scheme for particle-mesh-Ewald molecular dynamics simulations. Our method exploits smooth switch functions effectively to regulate direct and reciprocal space terms for the electrostatic interactions. The reciprocal term with the near field contributions removed is assigned to the slow class; the van der Waals and regulated particle-mesh-Ewald direct-space terms, each associated with a tailored switch function, are assigned to the medium class. All other bonded terms are assigned to the fast class. This versatile protocol yields good stability and accuracy for Newtonian algorithms, with temperature and pressure coupling, as well as for Langevin dynamics. Since the van der Waals interactions need not be cut at short distances to achieve moderate speedup, this integrator represents an enhancement of our prior multiple-time-step implementation for microcanonical ensembles. Our work also tests more rigorously the stability of such splitting schemes, in combination with switching methodology. Performance of the algorithms is optimized and tested on liquid water, solvated DNA, and solvated protein systems over 400 ps or longer simulations. With a 6 fs outer time step, we find computational speedup ratios of over 6.5 for Newtonian dynamics, compared with 0.5 fs single-time-step simulations. With modest Langevin damping, an outer time step of up to 16 fs can be used with a speedup ratio of 7.5. Theoretical analyses in our appendices produce guidelines for choosing the Langevin damping constant and show the close relationship among the leapfrog Verlet, velocity Verlet, and position Verlet variants. © *2002 American Institute of Physics.*
[DOI: 10.1063/1.1458542]

I. INTRODUCTION

The large-scale size of biomolecular simulations coupled with the growing demand for higher accuracy and physical relevance underscores the importance of developing more efficient and accurate simulation methods.[1] The Ewald summation method[2] is a well-established technique for computing electrostatic interactions accurately under periodic boundary conditions. Truncating the nonbonded interactions[3,4] is generally no longer considered competitive. The particle-mesh-Ewald (PME) method[5] is a promising variant derived from the standard Ewald method. The electrostatic interactions in particle-mesh-Ewald method are evaluated in a manner similar to that in the Ewald method, with real (or direct), reciprocal, and correction terms. Particle-mesh-Ewald algorithms approximate the reciprocal component through fast Fourier transform techniques, following smooth charge and potential interpolation on a grid. This effectively reduces the computational cost for the nonbonded terms from order $\mathcal{O}(N^2)$ to $\mathcal{O}(N \log N)$.[6] Still, the computational cost is high in biomolecular simulations due to both the large size of biomolecular systems and the large thermally-accessible conformational space needed to be sampled. In addition to faster computing platforms and parallel adaptations,[7,8] multiple-time-step methods have been adapted for molecular dynamics algorithms to reduce the overall computing time by performing electrostatic calculations less often than other energy and force components (e.g., Refs. 9, 10).

Most of the multiple-time-step methods for Ewald summations split the direct-space sum into two or more distance-based medium or slow classes while putting the intact reciprocal term into one of these classes.[11–13] Truncation-based multiple-time-step methods for periodic domains tend to achieve a larger outer time step than those based on the Ewald summation [e.g., the LN method (Refs. 14, 15) in CHARMM (Ref. 16)]. A numerical problem of particle-mesh-Ewald methods is that the finite number of wave vectors in the discrete approximation is thought to give rise to truncation and cancellation errors (due to the exclusion of intramolecular interactions).[17] This numerical feature restricts the largest timestep used for updating the reciprocal term, and hence the speedup that can be achieved with multiple-time-step/particle-mesh-Ewald methods. The main problem leading to this limitation is the association of the entire reciprocal term to a certain multiple-time-step class. Since the reciprocal term is the sum of all error functions (erf) of pairwise electrostatic interactions, it includes weight-reduced short-range interactions as well. Figure 1 shows that the direct component has a ‘‘tail’’ (slow terms) while the reciprocal component has a ‘‘head’’ (fast components). In practice, work has shown that the corresponding reciprocal

[a)]Author to whom correspondence should be addressed. Fax: 212-995-4152; electronic mail: qian@biomath.nyu.edu; schlick@nyu.edu

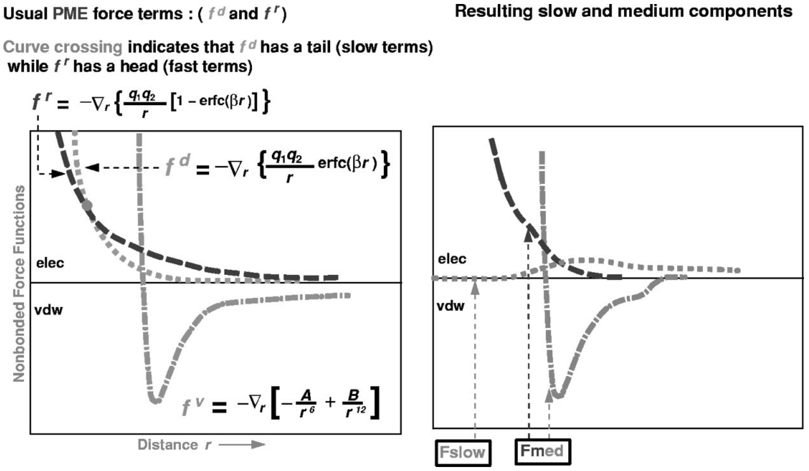

FIG. 1. Schematic illustration of the application of the improved force splitting scheme for PME applied to a pair of atoms with charges q_1 and q_2 of opposite sign, separated by interparticle distance r. The force switch function used is given in Eq. (10) (see Fig. 2). All forces are expressed by their magnitudes for simplicity. The components f^r, f^d, and f^v correspond to the original forces (left, see text), and F_{med} and F_{slow} are the final medium and slow forces, respectively (right). F_{med} includes the modified direct space term without a "tail" (switched off between a and b) and the modified van der Waals term (switched between v_1 and v_2); F_{slow} includes only the modified reciprocal term without a "head." The value c is the direct space truncation distance used in PME to derive default value of the Gaussian parameter β, and $c+\Delta c$ defines the size of the nonbonded pairlist.

force must be updated at most every 4 fs to conserve energy.[11,18]

Here we present a new distance based splitting scheme to rearrange the direct and reciprocal sums so that the near-field contribution to the reciprocal term is removed. This work represents an enhancement of our prior implementation for microcanonical ensembles[19] (which simply uses the reciprocal term for the slow force) since the van der Waals interactions need not be cut at short distances to achieve moderate speedup. We also test more rigorously the stability of such splitting schemes as well as switching methodology than a recent report,[20] whose short test simulations may not be representative.

Specifically, we present both Newtonian and Langevin multiple-time-step/particle-mesh-Ewald integrators, developed for the program AMBER,[21] with extensions to canonical, isothermal and isobaric-like ensembles as implemented in AMBER (based on Berendsen's weak coupling schemes[22,23]). The combined approach is found to be stable and accurate for outer time steps of 8 fs (Newtonian) and 16 fs (Langevin, with mild damping). Speedup factors approach 7 and 8 for Newtonian and Langevin, respectively, relative to single-time-step integrators at 0.5 fs for particle-mesh-Ewald protocols. Following a brief introduction to the Ewald method in Sec. II, we discuss distance-based force splitting in Sec. III and present the multiple-time-step/particle-mesh-Ewald integration in Sec. IV. Results are analyzed in Sec. V, and this is followed by a brief conclusion section. Two appendices analyze resonance in impulse and extrapolation-based MTS schemes and demonstrate the close relationship among the leapfrog, position, and velocity Verlet variants.

II. EWALD SUMMATION

With periodic boundary conditions, the energy for electrostatic interactions of a molecular system considers all atom pairs over all possible lattice cells. We denote the translation vector $\mathbf{n}=(n_xL, n_yL, n_zL)$ relative to the primary cell (where n_x, n_y, and n_z are integers and L is the cell dimension) and define the general distance vector between any pair of atoms i and j as

$$\mathbf{r}_{ij,n}=\mathbf{r}_{ij}+\mathbf{n}, \quad \text{where} \quad \mathbf{r}_{ij}=\mathbf{r}_j-\mathbf{r}_i,$$

and $\mathbf{r}_j$ and $\mathbf{r}_i$ are vector coordinates in the primary cell. We use (nonbold) $r_{ij,n}$ to denote the scalar magnitude of $\mathbf{r}_{ij,n}$. The total electrostatic energy is then expressed as

$$E_{\mathrm{elec}}=\frac{1}{2}\sum_{\mathbf{n}}\sum_{i,j}{}' \frac{q_iq_j}{r_{ij,n}}, \tag{1}$$

where the prime in the summation indicates that the $i=j$ interaction for $\mathbf{n}=0$ is not counted. The summation over i,j pairs in Eq. (1) extends over all N atoms in the system ($i,j=1,\ldots,N$) with partial charges $\{q_i\}$. The slowly-decaying nature of this long-range potential renders a straightforward summation impractical. Ewald, multipole, and other methods have been developed to remedy this problem.[24–28] The Ewald summation[2] effectively splits the task of evaluating Eq. (1) into two parts using a pair of complementary functions erf(x) and erfc(x)=1−erf(x), where

$$\mathrm{erf}(x)=\frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt.$$

For a detailed theory of Ewald summation, see Kittel,[29] for example. The resulting Ewald formula for the electrostatic energy then becomes

$$E_{\mathrm{elec}}=E^d+E^r, \tag{2}$$

where

$$E^d=\frac{1}{2}\sum_{\mathbf{n}}\sum_{i,j}{}' \frac{q_iq_j}{r_{ij,n}}\mathrm{erfc}(\beta r_{ij,n}), \tag{3}$$

and

$$E^r=\frac{1}{2}\sum_{\mathbf{n}}\sum_{i,j}{}' \frac{q_iq_j}{r_{ij,n}}\mathrm{erf}(\beta r_{ij,n}). \tag{4}$$

The inverse length β, a Gaussian-width parameter, alters the relative weights of the **direct** (E^d) and **reciprocal** (E^r) space contributions. For a given β, the real space term is calculated only for atom pairs within a certain distance range due to the fast decaying property of the erfc function. The reciprocal term is converted through Fourier transforms to

$$
\begin{aligned}
E^r &= \frac{1}{2}\sum_{\mathbf{n}} {\sum_{i,j}}' q_i q_j \sum_{\mathbf{k}} \exp(2\pi i\mathbf{k}\cdot\mathbf{r}_{ij,n}) \\
&\quad\times \int_0^\infty \exp(-2\pi i\mathbf{k}\cdot\mathbf{s})\,\frac{\operatorname{erf}(\beta s)}{s}\,d\mathbf{s} \\
&= \frac{1}{2V}{\sum_{i,j}}'\sum_{\mathbf{n}} q_i q_j\Bigg[\frac{1}{\pi}\sum_{\mathbf{k}\neq 0}\exp(2\pi i\mathbf{k}\cdot\mathbf{r}_{ij,n}) \\
&\quad\times \frac{\exp(-\pi^2k^2/\beta^2)}{k^2} + \int_0^\infty \frac{\operatorname{erf}(\beta s)}{s}\,d\mathbf{s}\Bigg],
\end{aligned}
$$

where $\mathbf{k}$ is the reciprocal space wave vector $(\mathbf{k}=(L/n_x, L/n_y, L/n_z))$ and V is the total volume of the system. For $\mathbf{n}=0$ and $i=j$, we have

$$
\lim_{r_{ij,n}\to 0}\frac{\operatorname{erf}(\beta r_{ij,n})}{r_{ij,n}} = \beta\operatorname{erf}'(0) = \frac{2\beta}{\sqrt{\pi}}.
$$

Thus, the term corresponding to $\mathbf{n}=0$ is removed to give

$$
\begin{aligned}
E^r &= \frac{1}{2V}\sum_{\mathbf{n}}\left(\sum_i q_i\right)^2\int\frac{\operatorname{erf}(\beta s)}{s}\,d\mathbf{s} \\
&\quad+\frac{1}{2\pi V}\sum_{\mathbf{n}}\sum_{i,j}\sum_{\mathbf{k}\neq 0}\frac{\exp(-\pi^2k^2/\beta^2)}{k^2}q_iq_j \\
&\quad\times\exp(2\pi i\mathbf{k}\cdot\mathbf{r}_{ij,n}) - \frac{\beta}{\sqrt{\pi}}\sum_i q_i^2 .
\end{aligned}
$$

In particular, for a neutral system $(\Sigma_i q_i = 0)$, we have

$$
\begin{aligned}
E^r &= \frac{1}{2\pi V_0}\sum_{i,j}\sum_{\mathbf{k}\neq 0}\frac{\exp(-\pi^2k^2/\beta^2)}{k^2}q_iq_j \\
&\quad\times\exp(2\pi i\mathbf{k}\cdot\mathbf{r}_{ij}) - \frac{\beta}{\sqrt{\pi}}\sum_i q_i^2 ,
\end{aligned}
\tag{5}
$$

where V_0 is the volume of the primary cell; above, and the sum over $\mathbf{n}$ was removed by using the identity $\Sigma_{\mathbf{n}}\exp(2\pi i\mathbf{k}\cdot\mathbf{n})/V = 1/V_0$. The first term in Eq. (5) is called the ***k*-space sum** (E^k) and can be simplified by defining

$$
S(\mathbf{k}) = \sum_i q_i\exp(2\pi i\mathbf{k}\cdot\mathbf{r}_i),
$$

known in crystallography as the **structure factor**.[29] It follows that

$$
E^k = \frac{1}{2\pi V_0}\sum_{\mathbf{k}\neq 0}\frac{\exp(-\pi^2k^2/\beta^2)}{k^2}S(\mathbf{k})S(-\mathbf{k}). \tag{6}
$$

The second term in the reciprocal sum [Eq. (5)] is called the **self-energy**,

$$
E^{\text{self}} = \frac{\beta}{\sqrt{\pi}}\sum_i q_i^2 . \tag{7}
$$

Added also to the decomposition of E_{elec} into E^d and E^r in Eq. (2) is a **dipole correction** term E^c depending on the dipole moment of the unit cell and the asymptotic order of summation,[30]

$$
E^c = \frac{2\pi}{(1+2\epsilon)V}\left(\sum_i q_i\mathbf{r}_i\right)\cdot\left(\sum_i q_i\mathbf{r}_i\right), \tag{8}
$$

where ϵ is the dielectric constant of the medium surrounding the assembly of unit cells. Note that for conducting boundary conditions, ϵ is ∞ and E^c vanishes. Finally, an intramolecular correction term is needed. This is because electrostatic interactions for atom pairs connected via three bonds or less are generally not considered in molecular dynamics programs (these atom pairs are collected in an exclusion list L_0). The intramolecular exclusion term E^{intra} becomes

$$
E^{\text{intra}} = \sum_{i,j\in L_0}\frac{\operatorname{erf}(\beta r_{ij})}{r_{ij}}. \tag{9}
$$

The complete Ewald formula is given by

$$
E_{\text{elec}} = E^d + E^r = E^d + E^c - E^{\text{self}} - E^{\text{intra}}.
$$

The advantage of decomposing the electrostatic energy as above is that the exponentially converging sum over $\mathbf{n}$ and $\mathbf{k}$ for E^d and E^k in Eqs. (3) and (6) allows the introduction of relatively small cutoffs (or effectively few wave vectors) without much loss of accuracy. Given real (R_0) and reciprocal space (k_0) cutoff values, there exists an optimal β such that the accuracy of the approximated Ewald sum is sufficiently high.[31] This follows the requirement that the real and reciprocal-space contributions to the error should be approximately equal.[31] Typically, β is chosen large enough so as to employ the minimum image convention for the direct term E^d, and the overall complexity of the direct sum is thereby $\mathcal{O}(N^2)$. For large systems, a fixed cutoff radius (e.g., 9 Å in AMBER) is generally used to further reduce the cost for direct sum[32] to $\mathcal{O}(N)$.

To produce an overall $\mathcal{O}(N\log N)$ method, the particle-mesh-Ewald method[5] approximates the reciprocal sum using fast Fourier transforms with convolutions on a grid where charges and potentials are interpolated onto the grid points. In addition, particle-mesh-Ewald does not interpolate but rather evaluates the forces analytically by differentiating the energies, thereby reducing memory requirements substantially.

III. DISTANCE BASED FORCE SPLITTING

Multiple-time-step techniques rely on the fact that the total force can be partitioned into distinct components which evolve (in time) on different time scales. The bond, angle, and torsion terms in the force field can be associated with time scales according to their characteristic periods derived from their harmonic potential forms. The nonbonded terms (van der Waals and electrostatic interactions) are not easily related to any unique time scales. Still, assignment to force classes can be made by assuming that the time scale of the nonbonded force decays as $1/r_{ij}$.[33] It is thus generally sufficient to define spherical shells of increasing radii around a particle to subdivide the nonbonded contributions to the force into terms characterized by different time scales.[33]

To avoid discontinuous changes of force and energies, each shell boundary is smoothed by a switch function which drops from 1 to 0 with a certain **healing length**. Such force

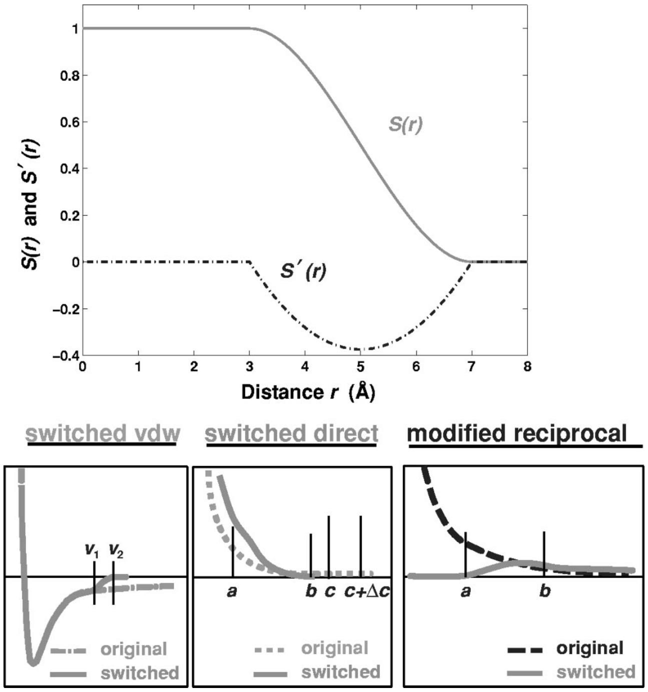

FIG. 2. The switch function $S(r)$ (solid line) and its derivative $S'(r)$ used in our multiple-time-step schemes [Eq. (10) (Ref. 36)]. Here a switch region between 3–7 Å is used. The continuity of the first-order derivative is an important requirement for switch functions. The application of the switch function to direct and reciprocal electrostatic interactions as well as van der Waals interactions is shown, so as to produce the right plot in Fig. 1.

splitting schemes are widely used in multiple-time-step integrators for both Ewald-based[11,12,34] and truncation-based protocols.[3,35,36]

Most of the Ewald-based multiple-time-step methods split the direct space term into two or more distance classes while putting the *intact* reciprocal term into one of these classes. Since the direct space interactions are typically truncated at a moderate value (8–10 Å), the enhancement in computational speedup associated with splitting the real space sum is limited due to the overhead of extra pairlist maintenance.[19] Moreover, since the Ewald reciprocal term is the sum of all erf-function-regulated pairwise electrostatic interactions (including weight-reduced short-range interaction term), the reciprocal force must be updated often (e.g., every 4 fs) to conserve energy. If the reciprocal term can also be regulated so the near-field contributions are removed, larger time steps for updating reciprocal interactions might be achieved.

In our improved version of force splitting for particle-mesh-Ewald, a nonbonded list up to $c+\Delta c$ is maintained for the evaluation of the direct-space and van der Waals terms (see Figs. 1 and 2). The van der Waals term is switched off between v_1 and v_2 and assigned to the medium class; all electrostatic interactions less than a cutoff distance b are also assigned to the medium class and smoothly switched off between a and b; the difference between this medium-class electrostatic contribution and the direct-space term is exactly the near-field contributions to be removed from the reciprocal-space term (see Figs. 1 and 2). Thus, the reciprocal term with the near-field contributions removed can be assigned to the slow class. All other bonded terms are assigned to the fast class. At each update of the medium-class force, the maximal particle movement is compared with a threshold value (~ 1 Å); once the threshold is reached, the nonbonded list is rebuilt. The buffer interval Δc is in the range of 0.5–1 Å.

The complete three classes force breakup can be summarized as follows:

$$F_{\text{fast}} = f_{\text{bond}} + f_{\text{angle}} + f_{\text{torsion}},$$

$$F_{\text{med}} = \tilde{f}^d + \tilde{f}^v,$$

$$F_{\text{slow}} = \tilde{f}^r,$$

where $\tilde{f}^v$, $\tilde{f}^d$, and $\tilde{f}^r$ are the switch-regulated van der Waals, direct, and reciprocal terms, respectively (see Fig. 2). Namely,

$$\tilde{f}^v = -\frac{1}{2}\sum_{\mathbf{n}} \sideset{}{'}\sum_{i,j:r_{ij,n}<v_2} S(r_{ij,n},v_1,v_2)\nabla_r\left[\frac{B_{ij}}{r_{ij,n}^{12}} - \frac{A_{ij}}{r_{ij,n}^{6}}\right],$$

$$\tilde{f}^d = -\frac{1}{2}\sum_{\mathbf{n}} \sideset{}{'}\sum_{i,j:r_{ij,n}<R_0} \left[S(r_{ij,n},a,b)\nabla_r \frac{q_i q_j}{r_{ij,n}}\right],$$

$$f^r = -\nabla_r E^r,$$

$$\tilde{f}^r = f^r + \frac{1}{2}\sum_{\mathbf{n}} \sideset{}{'}\sum_{i,j:r_{ij,n}<R_0} \left[S(r_{ij,n},a,b)\nabla_r \frac{q_i q_j}{r_{ij,n}} - \nabla_r \frac{q_i q_j}{r_{ij,n}} \operatorname{erfc}(\beta r_{ij,n})\right].$$

All force switches employ the following switch function[36] for the switch interval $[r_0,\ r_1]$, shown in Fig. 2,

$$S(r,r_0,r_1) = \begin{cases} 1 & \text{if } r \leqslant r_0, \\ x^2(2x-3)+1 & \text{if } r_0 < r < r_1, \\ 0 & \text{if } r \geqslant r_1, \end{cases} \tag{10}$$

where $x=(r-r_0)/(r_1-r_0)$. Note, that in multiple-time-step schemes, the total energy is typically computed at the outer time step.

Our separate treatment of van der Waals and electrostatic interactions allows the electrostatic interactions to be switched off in the near field (smaller cutoff parameter b), with a shifting of the remaining electrostatic computation to the reciprocal space (smaller Gaussian-width parameter β), without sacrificing the accuracy of the van der Waals term (which usually has the same cutoff as the direct-space term[19]). The removal of near-field interactions from reciprocal term is expected to allow a larger outer time step for updating the slow force.

IV. MULTIPLE-TIME-STEP INTEGRATORS FOR PARTICLE-MESH-EWALD

To guide the development of efficient multiple-time-step protocols, studies of very simple models, like the one-dimensional harmonic oscillator[15,37] are instructive. Though not much discussed until now, the two Verlet variants, known as position verlet (PV) and velocity verlet (VV) (Refs. 38–

40) offer different practical performance, though their error is the same in theory at the limit of infinitesimal time steps. Both are reversible[38] and symplectic.[41]

Consider Newton's equation of motion for an N-particle system,

$$M\dot{\mathbf{V}}=\mathbf{F}(\mathbf{X})=-\nabla E(\mathbf{X}(t)),$$

where M is the mass matrix, $\mathbf{X}$ and $\mathbf{V}$ are the collective position and velocity vectors, and the dot superscripts denote differentiation with respect to time t. The two Verlet schemes are described as[42]

$$\mathbf{V}^{n+1/2}=\mathbf{V}^{n}+\frac{\Delta\tau}{2}M^{-1}\mathbf{F}^{n},$$

$$\mathbf{X}^{n+1}=\mathbf{X}^{n}+\Delta\tau\mathbf{V}^{n+1/2},$$

$$\mathbf{F}^{n+1}=-\nabla E(\mathbf{X}^{n+1}),$$

$$\mathbf{V}^{n+1}=\mathbf{V}^{n+1/2}+\frac{\Delta\tau}{2}M^{-1}\mathbf{F}^{n+1},$$

for VV and

$$\mathbf{X}^{n+1/2}=\mathbf{X}^{n}+\frac{\Delta\tau}{2}\mathbf{V}^{n},$$

$$\mathbf{F}^{n+1/2}=-\nabla E(\mathbf{X}^{n+1/2}),$$

$$\mathbf{V}^{n+1}=\mathbf{V}^{n}+\Delta\tau M^{-1}\mathbf{F}^{n+1/2},$$

$$\mathbf{X}^{n+1}=\mathbf{X}^{n+1/2}+\frac{\Delta\tau}{2}\mathbf{V}^{n+1},$$

for PV. Here superscripts n denote the discrete approximation at time $n\Delta\tau$, where $\Delta\tau$ is the time step.

For Newtonian dynamics, though VV-based impulse multiple-time-step schemes[36] are widely used, the PV has stability advantages at large time steps.[40] Both Verlet variants can be traced back to the leapfrog/Verlet/Störmer scheme[24,43–45] (see Appendix B). For Langevin dynamics, constant extrapolation is ideal for slow force evaluation (to damp out resonances[15]); both the midpoint and constant extrapolation schemes with velocity corrections are good candidates for evaluation of the medium force. The LN multiple-time-step protocol[15,39] with midpoint and constant extrapolation for the medium and slow-force evaluation, respectively, has proven to be effective for truncation-based schemes.[3,35,36]

In our proposed multiple-time-step force splitting for particle-mesh-Ewald, recall that the slow force is composed of the reciprocal term with the cancellation term from the near field interactions; the switched electrostatic and van der Waals interactions are evaluated once for each medium-force update. We have found that PV-based constant extrapolation with velocity correction is more effective than analogous VV schemes (based on resonance analysis of a 1D harmonic oscillator; details are provided in the Ph.D. thesis of Qian[37]). Therefore, the PV-based impulse multiple-time-step method (PV-MTS, see Table I), is an optimal choice for Newtonian dynamics; PV-based constant extrapolation with velocity correction for the medium force, along with constant extrapolation for the slow force (LN2, see Table II), is a good candidate for Langevin dynamics. Furthermore, with Berendsen's thermostat and barostat coupling,[22,23] temperature and pressure can be controlled to mimic (but not reproduce rigorously) canonical, isothermal, and isobaric ensembles.

TABLE I. Position Verlet based impulse multiple-time-step schemes for Newtonian (PV-MTS) and Langevin (LANG-MTS) dynamics.

PV-MTS

```
$X = X^n$    $V = V^n$
  $F_{\text{fast}} = \texttt{Fast}(X)$
  $F_{\text{med}} = \texttt{Medium}(X)$
  $F_{\text{slow}} = \texttt{Slow}(X)$
  $V = V + F_{\text{slow}} k_1 k_2 \frac{\Delta\tau}{2}$
  $P_t = 0$
  For $i = 1 : k_2$
    $V = V + F_{\text{med}} k_1 \frac{\Delta\tau}{2}$
    For $j = 1 : k_1$
      $X = X + V\frac{\Delta\tau}{2}$
      $\texttt{SHAKE}(X, V, \frac{\Delta\tau}{2})$
      $P_t = P_t + \texttt{getP}(X, V)$
      $F_{\text{fast}} = \texttt{Fast}(X)$
      $V = V + F_{\text{fast}}\Delta\tau$
      $X = X + V\frac{\Delta\tau}{2}$
      $\texttt{SHAKE}(X, V, \frac{\Delta\tau}{2})$
    End
    $F_{\text{med}} = \texttt{Medium}(X)$
    $V = V + F_{\text{med}} k_1 \frac{\Delta\tau}{2}$
  End
  $P_t = P_t/(k_1 k_2)$
  $\texttt{scalX}(X, P_t, P_0, k_1 k_2 \Delta\tau)$
  $F_{\text{slow}} = \texttt{Slow}(X)$
  $V = V + F_{\text{slow}} k_1 k_2 \frac{\Delta\tau}{2}$
$X^{n+1} = X$    $V^{n+1} = V$
```

LANG-MTS

```
$X = X^n$    $V = V^n$
  $F_{\text{fast}} = \texttt{Fast}(X)$
  $F_{\text{med}} = \texttt{Medium}(X)$
  $F_{\text{slow}} = \texttt{Slow}(X)$
  $V = V + F_{\text{slow}} k_1 k_2 \frac{\Delta\tau}{2}$
  $P_t = 0$
  For $i = 1 : k_2$
    $V = V + F_{\text{med}} k_1 \frac{\Delta\tau}{2}$
    For $j = 1 : k_1$
      $X = X + V\frac{\Delta\tau}{2}$
      $\texttt{SHAKE}(X, V, \frac{\Delta\tau}{2})$
      $P_t = P_t + \texttt{getP}(X, V)$
      $F_{\text{fast}} = \texttt{FAST}(X)$
      $R = \texttt{rand}(\gamma, \Delta\tau, k_B T)$
      $V = V(1 - \gamma\Delta\tau) + (F_{\text{fast}} + R)\Delta\tau$
      $X = X + V\frac{\Delta\tau}{2}$
      $\texttt{SHAKE}(X, V, \frac{\Delta\tau}{2})$
    End
    $F_{\text{med}} = \texttt{Medium}(X)$
    $V = V + F_{\text{med}} k_1 \frac{\Delta\tau}{2}$
  End
  $P_t = P_t/(k_1 k_2)$
  $\texttt{scalX}(X, P_t, P_0, k_1 k_2 \Delta\tau)$
  $F_{\text{slow}} = \texttt{Slow}(X)$
  $V = V + F_{\text{slow}} k_1 k_2 \frac{\Delta\tau}{2}$
$X^{n+1} = X$    $V^{n+1} = V$
```

TABLE II. Position Verlet based extrapolation multiple-time-step schemes for Langevin dynamics (LN and LN2).

LN

```
$X = X^n$    $V = V^n$
  $F_{\text{fast}} = \texttt{Fast}(X)$
  $F_{\text{med}} = \texttt{Medium}(X)$
  $F_{\text{slow}} = \texttt{Slow}(X)$
  $P_t = 0$
  For $i = 1 : k_2$
    $X^* = X + V k_1 \frac{\Delta\tau}{2}$
    $F_{med} = \texttt{Medium}(X^*)$
    For $j = 1 : k_1$
      $X = X + V\frac{\Delta\tau}{2}$
      $\texttt{shk}(X, V, \frac{\Delta\tau}{2})$
      $P_t = P_t + \texttt{getP}(X, V)$
      $F_{\text{fast}} = \texttt{Fast}(X)$
      $R = \texttt{rand}(\gamma, \Delta\tau, k_B T)$
      $F = F_{\text{fast}} + F_{\text{med}} + F_{\text{slow}} + R$
      $V = V(1 - \gamma\Delta\tau) + F\Delta\tau$
      $X = X + V\frac{\Delta\tau}{2}$
      $\texttt{SHAKE}(X, V, \frac{\Delta\tau}{2})$
    End
  End
  $P_t = P_t/(k_1 k_2)$
  $\texttt{scalX}(X, P_t, P_0, k_1 k_2 \Delta\tau)$
  $F_{\text{slow}} = \texttt{Slow}(X)$
$X^{n+1} = X$    $V^{n+1} = V$
```

LN2

```
$X = X^n$    $V = V^n$
  $F_{\text{fast}} = \texttt{Fast}(X)$
  $F_{\text{med}} = \texttt{Medium}(X)$
  $F_{\text{slow}} = \texttt{Slow}(X)$
  $P_t = 0$
  For $i = 1 : k_2$
    For $j = 1 : k_1$
      $X = X + V\frac{\Delta\tau}{2}$
      $\texttt{SHAKE}(X, V, \frac{\Delta\tau}{2})$
      $P_t = P_t + \texttt{getP}(X, V)$
      $F_{\text{fast}} = \texttt{Fast}(X)$
      $R = \texttt{rand}(\gamma, \Delta\tau, k_B T)$
      $F = F_{\text{fast}} + F_{\text{med}} + F_{\text{slow}} + R$
      $V = V(1 - \gamma\Delta\tau) + F\Delta\tau$
      $X = X + V\frac{\Delta\tau}{2}$
      $\texttt{SHAKE}(X, V, \frac{\Delta\tau}{2})$
    End
    $F^{\text{old}}_{\text{med}} = F_{\text{med}}$
    $F_{\text{med}} = \texttt{Medium}(X)$
    $V = V + (F_{\text{med}} - F^{\text{old}}_{\text{med}}) k_1 \frac{\Delta\tau}{2}$
  End
  $P_t = P_t/(k_1 k_2)$
  $\texttt{scalX}(X, P_t, P_0, k_1 k_2 \Delta\tau)$
  $F_{\text{slow}} = \texttt{Slow}(X)$
$X^{n+1} = X$    $V^{n+1} = V$
```

For pressure coupling, the internal pressure obtained from the molecular virial and kinetic energy is measured every inner time step, but the scaling of box dimension and coordinates is performed once per outer time step (the scaling factor is derived from the average internal pressure). In this way, the slow force evaluation can be performed after the scaling operation to avoid approximation errors that can arise from multiple scaling in the inner cycle. For temperature coupling, in contrast, velocity scaling is performed at every inner time step to ensure smooth motion. The scaling factor is re-evaluated every outer time step from the average kinetic energy over the multiple inner cycles.

Both our multiple-time-step schemes for Newtonian and Langevin dynamics (PV-MTS and LN2, respectively) are given in Tables I and II with Berendsen's pressure coupling and SHAKE (Ref. 46) constraints applied; Berendsen's temperature coupling for Newtonian dynamics is similar to pressure coupling and omitted for simplicity. The original LN multiple-time-step protocol[15,39] is also given for comparison (Table II). All symbols in these tables have their usual meaning as defined above. In addition, P_t is the accumulated pressure and P_0 the reference pressure. Given an inner time step $\Delta\tau$, the medium force is updated every k_1 inner time steps at $\Delta t_m = k_1 \Delta\tau$, and the slow force is recalculated every k_2 medium cycles at $\Delta t = k_2 \Delta t_m = k_1 k_2 \Delta\tau$. The symbol `getP` in the code sketched is the pseudofunction that obtains the current pressure; `SHAKE` represents the SHAKE constrained dynamics operation[46] (applied to coordinates only), and `scalX` indicates coordinate rescaling in the pressure-control protocol. Although it seems that VV schemes might save an extra SHAKE evaluation (with respect to PV schemes), the latter can in fact be rearranged in a way to avoid two SHAKE evaluations per inner loop.[19] In practice, we find that the advantage of extrapolation-based MTS methods[39] is ruined by the cancellation error of Ewald methods[17] (results not shown); the alternative is to use impulse-based multiple-time-step methods for Langevin dynamics (LANG-MTS, see Table I) as well. We emphasize that with improved treatment of the cancellation error (not yet available in AMBER6.0), it is likely extrapolative MTS methods will regain their advantage and the outer time step can be pushed yet further.

V. NUMERICAL EXPERIMENTS

A. Biomolecular systems

Our three representative test cases are a water box (49 $\times$49$\times$49 Å^3) of 4096 molecules (12 288 total atoms); a protein (dihydrofolate reductase) solvated in a water box (70 $\times$60$\times$54 Å^3) with counterions (11 Na$^+$, 22 930 total atoms); and a 14-base-pair DNA double helix (DNA: GCTAAAAAAGGGCA) with counterions and solvent water molecules (26 Na$^+$, 15 320 total atoms) in a box (71$\times$50 $\times$43 Å^3).[47] All systems are minimized for 1000 cycles using the steepest descent method followed by 5000 cycles of conjugate gradient. The three systems are heated to 300 K over 10 ps, with SHAKE (Ref. 46) constraints on all bonds involving hydrogen, and equilibrated for 18 ps by the original leapfrog integrator in AMBER6.0 (Ref. 21) with a 9 Å direct space cutoff distance and a time step of 1 fs.

B. Multiple-time-step performance assessors

Two energy conservation parameters have been used in the past as quality control for energy conservation.[34,48,28] One is the relative energy error (η), given as

$$\eta = \frac{1}{N_T} \sum_{i=1}^{N_T} \left| \frac{E_{\text{tot}}^i - E_{\text{tot}}^0}{E_{\text{tot}}^0} \right|, \tag{11}$$

where E_{tot}^i is the total energy at step i, E_{tot}^0 is the initial energy, and N_T is the total number of sampling points. A value of $\eta \leqslant 0.003$, i.e., $\log_{10}\eta < -2.5$, is considered acceptable in terms of numerical accuracy.[49] This parameter can be a good indicator of energy conservation, if the simulation is performed long enough to reflect error accumulation. We have found that simulations of length 400 ps or longer are required to verify integrator stability. That is, small but systematic drifts can take several hundred picoseconds to emerge. Hence, multiple-time-step/particle-mesh-Ewald tests based on several picoseconds (e.g., as in Ref. 20) are far too short to demonstrate stability and accuracy, and reported speedups and accuracies of integrators may be misleading.

Another trajectory assessment parameter is the energy conservation ratio (R) defined by

$$R = \Delta E_{\text{tot}} / \Delta E_k, \tag{12}$$

where ΔE_{tot} and ΔE_k are the RMS deviations of total energy and kinetic energy, respectively. The total energy is generally considered well conserved when $R \leqslant 0.05$.[50] However, as noted by Procacci *et al.*,[18] this criterion is misleading for comparing multiple-time-step to single-time-step methods. In particular, well designed multiple-time-step integrators can compute structural and dynamical properties of the system more accurately than single-time-step simulations of comparable and smaller R values.[33]

A more direct indication of energy conservation for multiple-time-step methods is the relative energy drift rate κ, which we define to be the slope of the least-squares best fit of the energy evolution over time to a straight line. In the least square sense (denoted by $\doteq$), we can write

$$\frac{E_{\text{tot}}^i}{E_{\text{tot}}^0} \doteq \frac{\kappa}{t_u} t_i + b_0, \tag{13}$$

where $t_i = i\Delta t$ at step i, b_0 is a parameter, and t_u is the preferred time unit to make κ unitless. If t_u is in units of picoseconds, κ gives the relative energy drift per picosecond. A small κ value is a good indicator of energy conservation. For reference, $\log_{10}\kappa$ values for the single-time-step leapfrog integrator in AMBER are about -6.7, -6.4, and -6.3 for $\Delta\tau = 0.5$, 1.0, and 2.0 fs, respectively.

C. Time step and switch function parameters

In addition to the regular parameters for the particle-mesh-Ewald setup, the multiple-time-step/particle-mesh-Ewald integrators have several tunable variables. These are the inner time step ($\Delta\tau$), medium and slow force update frequencies (k_1 and k_2), and the switch-interval parameters for electrostatic (a and b) and van der Waals (v_1 and v_2) interactions. Since force switching aims to maintain good energy

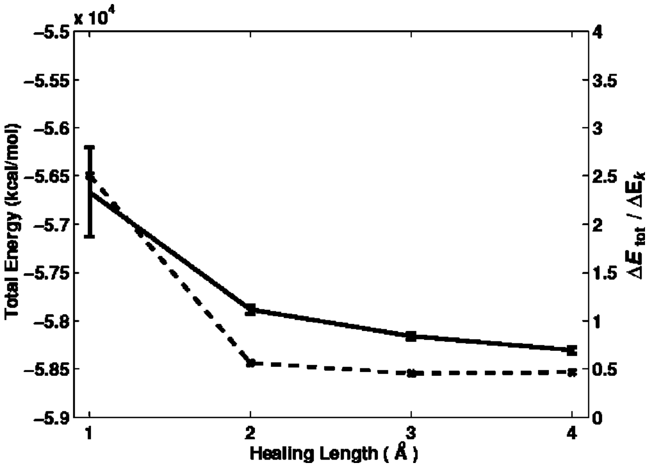

FIG. 3. Energy conservation for different healing lengths as evaluated by 200 ps dynamics simulations for our solvated protein system. The solid line with error bars is the total energy and its standard deviation (scale at left vertical axis). The dashed line is the conservation ratio $\Delta E_{\mathrm{tot}}/\Delta E_k$ (with scale at right vertical axis), the fluctuation of total energy divided by the fluctuation of total kinetic energy [Eq. (12)].

conservation for multiple-time-step integrators, the width of the switch region (the "healing length" is $b-a$) influences energy conservation and must be chosen with care. For Langevin dynamics, the damping constant γ can also be adjusted to control the coupling strength. This is because the use of Langevin dynamics is numerical, to damp instabilities,[39] rather than physical.[10] A variety of numerical tests are performed on three test cases to define the acceptable parameter regions and optimal parameter sets for different molecular dynamics protocols (see below).

D. Results on appropriate buffer lengths

Simulations with the multiple-time-step protocol of 1/2/6 fs, for fast/medium/slow force partitioning were performed for the solvated protein for 200 ps to determine an appropriate switch buffer length. The electrostatic force in the medium class is switched off from $a=5$ Å to b and the van der Waals force from $v_1=6$ Å to v_2. The healing length $b-a=v_2-v_1$ was varied from 1 to 4 Å. For an outer time step 4 fs or less, we find that the total energy is not sensitive to the healing length or the switch functions (results not shown). For outer time steps of 6 fs or larger, a healing length of 3 Å or more for both the electrostatic and van der Waals interactions is necessary to suppress the energy drift (Fig. 3). Implementation of multiple-time-step integrators without force switches (e.g., step function used in Ref. 20) suffer from this

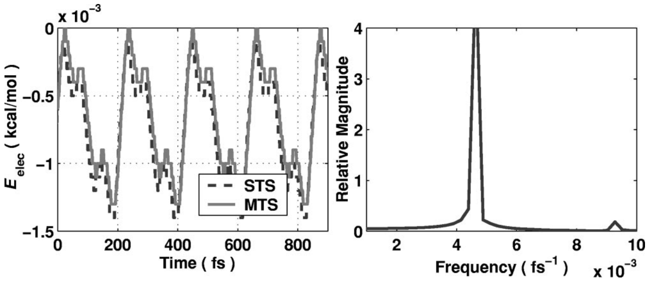

FIG. 4. Characteristic periods for the cancellation term for particle-mesh-Ewald in AMBER6.0. Left: electrostatic energies of the single water molecule system (Ref. 17) calculated from the single-time-step method (leapfrog Verlet, $\Delta\tau=0.5$ fs) and MTS method (position Verlet, $\Delta\tau=0.5$ fs, $k_1=2$ and $k_2=2$). Right: the Fourier transform of the autocorrelation function of electrostatic energy. Two periods are captured (110 and 200 fs).

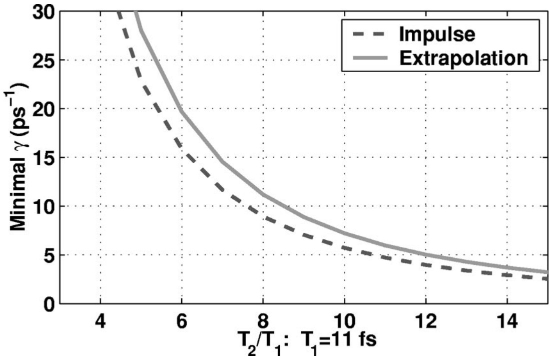

FIG. 5. Lower bound for γ for extrapolation and impulse multiple-time-step methods as a function of the slow to fast period ratio, as calculated in Appendix A. The fast period ($T_1=11$ fs) roughly corresponds to the characteristic period of C–H bond stretching.

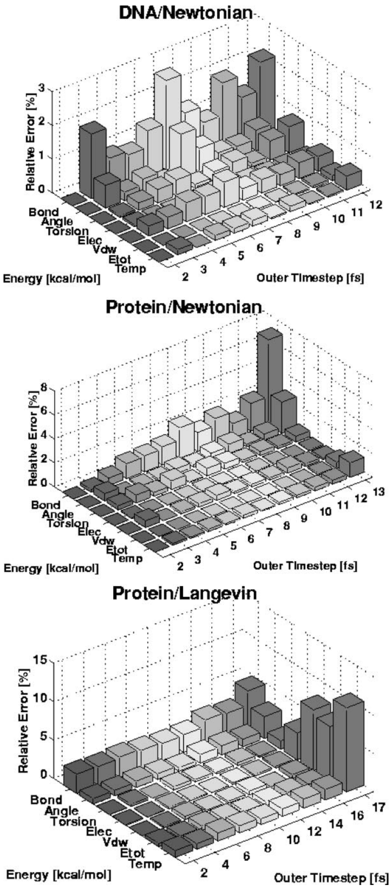

FIG. 6. The deviation of energy components relative to the reference trajectories for impulse multiple-time-step Newtonian and Langevin integrators for solvated systems. All simulations have an inner time step of 0.5 fs and a medium time step of 1 fs. Newtonian multiple-time-step simulations with 0.5/1/2 fs protocols are used as reference. (This reference is used rather than a single-time-step scheme to mimic the same switch applied to the van der Waals force; similar results are obtained for comparisons without a van der Waals switch and will be reported by Barash and Schlick.) All simulations are 5 ps in length.

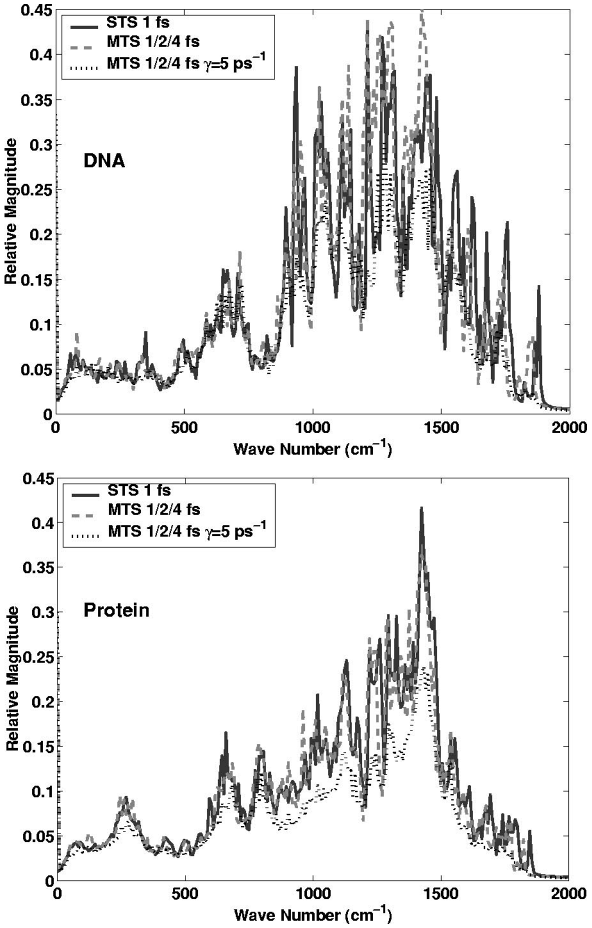

FIG. 7. Spectral densities of solvated DNA (top) and protein (bottom) systems (derived from all solute atoms) for various protocols. For protocol details, see Sec. V E.

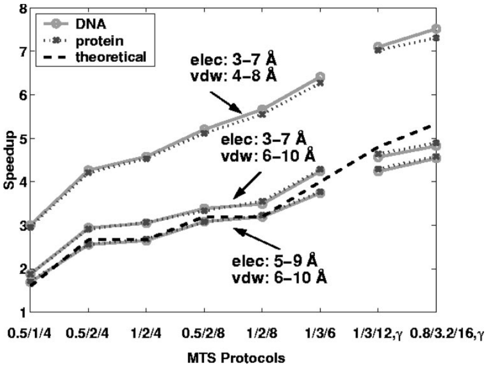

FIG. 8. Speedup for different multiple-time-step protocols (listed in the horizontal axis) relative to the single-time-step method for three sets of electrostatic/van der Waals switches. The damping constant of $\gamma = 5\ \mathrm{ps}^{-1}$ is used in the two Langevin simulations reported. The dashed line represents the analytical speedup estimate assuming the same nonbonded list size and direct-space cutoff for both multiple and single-time-step methods [see Eq. (14)]. The base-line single-time-step simulation has a nonbonded list of 10 Å and a 9 Å direct-space cutoff. All simulations are 1 ps in length.

outer time step barrier. Based on our experiments, we choose the buffer length to be 3 or 4 Å in our preferred multiple-time-step/particle-mesh-Ewald protocols.

E. Limits for outer time step

Procacci *et al.*[17] revealed a characteristic period associated with the cancellation error of Ewald methods. This error arises from the truncation (due to the discrete summation) at some cutoff wave vector value k_0 and removal of the erf-weighted electrostatic interactions (e.g., intramolecular excluded interactions) from the reciprocal-space term. The simple system devised in Ref. 17 (single water molecule in a cubic box with side length of 64 Å) yields a total electrostatic energy that corresponds exactly to the cancellation error. The cancellation errors from Procacci *et al.* show a period of the order 10–20 fs.

The instability of the cancellation errors can be suppressed if a proposed correction term is included,[17] but this is not implemented in AMBER6.0. We found that the PME implementation in AMBER6.0 (Ref. 51) shows similar behavior for the correction term, with two characteristic periods on the order of 110 and 200 fs (Fig. 4). This imposes an upper bound on the outer time step [the linear stability limit is the period over π (Ref. 39)] of 35 fs and of 25 fs [period over $\sqrt{2}\pi$ (Refs. 15,42)] to avoid fourth-order resonance. Langevin dynamics protocols with a proper γ can suppress the first resonance spike near half of the fast period. The periodic slow force also enforces a lower bound for γ of about $5\ \mathrm{ps}^{-1}$ to suppress the first resonance spike (see Fig. 5 and Appendix A).

To find the largest outer time step applicable for our impulse multiple-time-step integrators for both Newtonian and Langevin dynamics, we varied the outer time step from 1 to 20 fs, with fixed inner time step ($\Delta\tau = 0.5$ fs) and medium force update frequency ($k_1 = 2$, that is $\Delta t_m = 1$ fs). The electrostatic force in the medium class is switched off from 5 to 9 Å, and the van der Waals force is similarly treated from 6 to 10 Å. A nonbonded-interaction list over 10 Å is maintained with a 1 Å buffer region and a 9 Å direct space cutoff. All simulations are performed for 5 ps (short simulations). (The candidates that will be produced for optimized protocols will be studied for longer simulations to verify stability and accuracy.) Both the DNA and protein test cases are examined. Both show an upper bound of outer time step near 8 fs (Fig. 6) for Newtonian dynamics and 16 fs for Langevin dynamics (Fig. 6). This indicates that these limits are system independent. (Longer simulations with an outer time step larger than 8 fs exhibit notable energy drifts soon after 20 ps; results not shown.)

To guarantee that these multiple-time-step implementations do indeed generate the correct dynamics, the spectral density plots from multiple and single-time-step methods are compared in Fig. 7. All simulations for these analyses are based on 9.6 ps trajectories with velocities of solute atoms recorded every 2 fs. All bonds involving hydrogens are constrained with SHAKE. A time step of 0.5 fs is used for the reference single-time-step method and 1/2/4 fs for multiple-time-step. Our figure denotes the multiple-time-step protocol by $\Delta\tau/\Delta t_m/\Delta t$ for Newtonian dynamics and $\Delta\tau/\Delta t_m/\Delta t, \gamma$ for Langevin dynamics. The damping parameter $\gamma = 5\ \mathrm{ps}^{-1}$ is used for Langevin dynamics.

Figure 7 demonstrates that our multiple-time-step implementations give similar spectral density distributions relative to single-time-step methods for Newtonian dynamics, with

TABLE III. Results of long multiple-time-step simulations for stability and accuracy analysis.

System	ID	Switch[†] (Å)	MTS[‡]	t (ps)	$\Delta E_{\mathrm{tot}}/\Delta E_k$ (kcal/mol)	T/ΔT (K)	R	$\log_{10}\eta$	$\log_{10}\kappa$	comment
solvated DNA	0	9	0.5	200	−45426.16/ 0.76	297.55/1.93	0.013	−4.72	−6.72	reference
	a	5-9/6-10	1/2/8	400	−45779.10/ 32.31	298.50/1.81	0.57	−3.25	−5.04	good
	b	3-7/4-8	1/2/8	400	−46252.81/ 28.37	298.93/1.95	0.46	−3.30	−4.40	unstable
	c	**3-7/4-8**	**1/3/6**	**400**	**−46234.83/ 13.07**	**299.01/1.91**	**0.21**	**−3.60**	**−5.30**	good
	d	**3-7/4-8**	**0.8/3.2/16(5)**	**400**	**−45614.65/131.22**	**305.98/2.46**	**1.71**	**−2.65**	**−5.07**	good
	e	3-7/6-10	0.8/3.2/16(5)	500	−45714.33/134.49	307.87/2.54	1.70	−2.62	−5.24	good
	f	5-9/6-10	1/3/12(5)	400	−45341.20/128.46	304.62/2.20	1.87	−2.64	−5.09	good
solvated protein	0	9	0.5	200	−57747.81/ 0.88	298.62/1.56	0.012	−5.05	−7.02	reference
	a	5-9/6-10	1/2/8	800	−58246.78/ 41.36	300.21/1.47	0.60	−3.40	−4.99	good
	b	3-7/4-8	1/2/8	400	−58996.54/ 35.96	300.26/1.59	0.48	−3.29	−4.40	unstable
	c	**3-7/4-8**	**1/3/6**	**400**	**−58963.60/ 16.03**	**300.61/1.56**	**0.21**	**−3.66**	**−5.32**	best Newtonian
	d	**3-7/4-8**	**0.8/3.2/16(5)**	**400**	**−58059.49/160.14**	**307.47/2.03**	**1.68**	**−2.65**	**−5.01**	good
	g	5-9/6-10	1/2/4	400	−58394.18/ 23.12	298.87/1.54	0.32	−3.50	−6.32	excellent
	h	5-9/6-10	1/2/4,NPT*	1200	−58299.20/ 46.96	300.00/1.51	0.66	−3.19	−6.05	excellent
	i	5-9/6-10	0.8/3.2/16(5)	400	−57716.25/167.70	305.82/1.75	2.04	−2.64	−5.20	good
	j	3-7/6-10	1/3/6	400	−59357.42/ 32.63	300.45/1.50	0.20	−3.70	−6.60	excellent
	k	3-7/4-8	1/3/6/NVT[§]	400	−59039.15/ 45.01	300.09/1.54	0.62	−3.21	−6.12	excellent
water	l	5-8/6-10	1/2/8	400	−32579.26/ 32.60	298.13/2.12	0.63	−3.20	−4.99	good

[†]: Switches are given for the electrostatic potential followed by van der Waals parameters.

[‡]: Timesteps are given in the form of $\Delta\tau/\Delta t_m/\Delta t$ in fs followed by γ for Langevin dynamics in the unit of ps^{-1}.

*: isothermal isobaric simulations.

[§]: constant temperature simulations.

comparable quality. The Langevin multiple-time-step simulations produce peaks at the same wave numbers with lower magnitudes due to the stochasticity, as expected.

F. Optimized protocols

For a given multiple-time-step protocol ($\Delta\tau/\Delta t_m/\Delta t$), the electrostatic and van der Waals interactions in the medium class are evaluated from the nonbonded atom pairlist at each medium time step with a computational cost of C_m per update. During the slow force update at each outer time step, the nonbonded atom pairlist is accessed again to obtain the correction term for electrostatic force. Assuming that the nonbonded computation dominates in total computational cost, and b is close to $c+\Delta c$, evaluating the slow force requires close to C_m work. For an outer time step Δt, the total CPU costs are thus approximately $(k_2+1)C_m$. For a single-time-step method with a reference time step $\Delta\tau_0$, the total computational cost to cover an interval of length Δt is $C_m(\Delta t/\Delta\tau_0)=k_1k_2C_m(\Delta\tau/\Delta\tau_0)$. Thus, an analytical estimate for the limiting multiple-time-step speedup is given from these two estimates as

$$\frac{k_1k_2}{k_2+1}\left(\frac{\Delta\tau}{\Delta\tau_0}\right). \tag{14}$$

We can see from Fig. 8 that the maximal achievable speedup is quite modest. A slight improvement can be made by moving the switch region of the electrostatic interactions into the near field, which reduces the nonbonded list maintenance time for the electrostatic force update. Greater speedups can also be achieved by reducing the nonbonded list further through moving the switch region of van der Waals interactions into the near field.[19] Short simulations of length 1 ps for our DNA and protein systems were performed for a variety of MTS protocols to experiment with associated parameter sets.

Figure 8 shows that the speedup of each protocol is independent of the system size and content. The best protocol for Newtonian dynamics is 1/3/6 fs with a switch for electrostatic interaction from 3 to 7 Å and a switch for van der Waals interactions from 4 to 8 Å. A speedup factor of 6.5 is achieved relative to 0.5 fs single-time-step simulations. For Langevin dynamics, the optimized protocol has the time-step combination 0.8/3.2/16 fs for $\gamma=5\,\mathrm{ps}^{-1}$ with the same switches mentioned above. The speedup factor is 7.5 for Langevin dynamics. Of course, larger γ values can be used to push up the Langevin outer time step further and hence the speedup.

Simulations of length 400 ps or longer were then performed for different protocols to verify their long time stabilities (Table III). All simulations in Table III are stable, with no notable energy drifts during the simulation, with the exception of protocol b for Newtonian dynamics, which has near field switch regions ($b=8$ Å) and larger outer time step ($\Delta t=8$ fs).

For reference, single-time-step simulations of 200 ps with a time step of 0.5 fs for our protein or DNA systems have the drift rate $\kappa\sim2\times10^{-7}$, i.e., $\log_{10}\kappa\sim-6.7$ (protocol 0, Table III). In our study, we found that values of $\kappa\leqslant3\times10^{-5}$, i.e., $\log_{10}\kappa<-4.5$ for multiple-time-step integrators, indicate good energy conservation (Table III). The best protocol for Newtonian dynamics also has the best energy conservation ratio R ($R=0.21$). Systematically, all Langevin simulations yield larger R values (about 1.7) as expected from the stochastic formulation.

For Langevin dynamics, both R and $\log_{10}\eta$ values are not good indicators of energy conservation; the $\log_{10}\kappa$ values seem more proper for stability assessment.

Figure 9 shows the energy evolution of the DNA system with an 8 Å cutoff (protocols b, c, and d). While Langevin dynamics exhibits intrinsically larger energy fluctuations than Newtonian dynamics, both protocols c and d have very low relative energy drift rates ($\log_{10}\kappa=-5.30$ and -5.07, respectively). The Newtonian dynamics trajectories at 8 fs with 8 Å cutoffs (protocol b) have noticeable energy

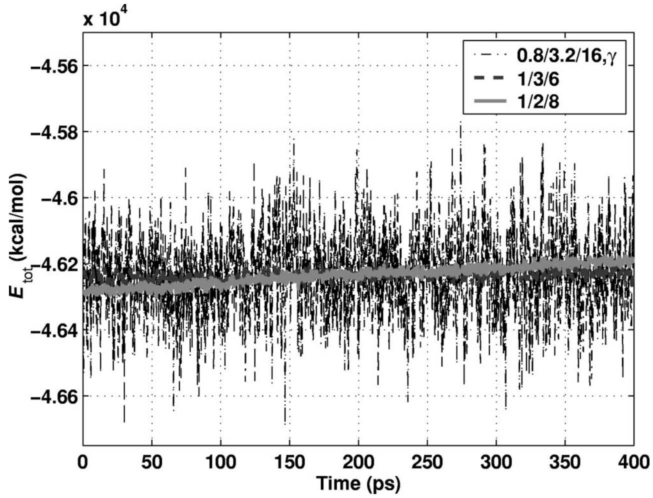

FIG. 9. Energy evolution for the solvated DNA system with a 3–7 Å switch for electrostatic force and a 4–8 Å switch for van der Waals force. All simulations are 400 ps in length.

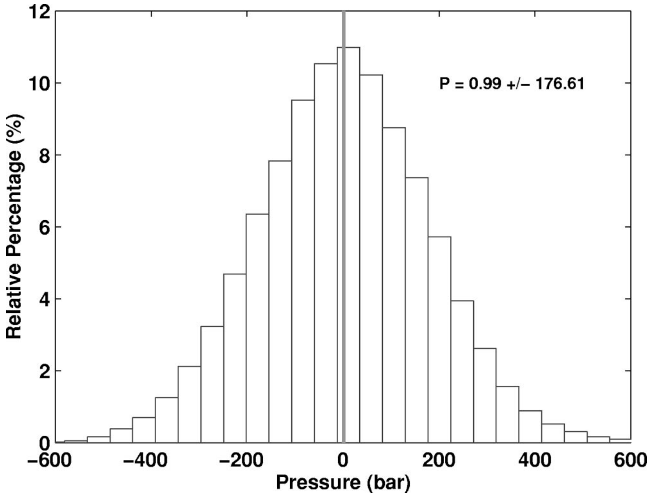

FIG. 10. Pressure histogram of the solvated protein over the last 800 ps for a 1.2 ns simulation with the multiple-time-step protocol 1/2/4 fs. The vertical line in the center indicates the external pressure (1 bar).

drift rate ($\log_{10}\kappa = -4.4$), about four times larger than protocol d.

At the time of our writing this manuscript, a preprint communicated to us by the authors Zhou *et al.*[20] pointed out the same problem of fast components presented in reciprocal term of Ewald methods. Their very similar solution uses a step function rather than a smooth function to remove the fast component from reciprocal term. In our study, we find that the medium force stability can be improved by a smooth switch function. Though switch functions are used for smooth transition of different force classes, the unaddressed problem of sudden truncation resulting from the step function formulation will probably restrict the largest medium time step applicable and introduce energy drift for long duration simulations. Indeed, the simulation length of 1 ps (Ref. 20) is much too short to assess multiple-time-step/particle-mesh-Ewald integration stability.

G. Temperature and pressure controls

Our protocols for thermostat and barostat coupling with multiple-time-step integrators are verified by a 1.2 ns simulation for the solvated protein (protocol h). The pressure histogram for the last 800 ps of the simulation shows an average internal pressure of 0.99 bar and a root-mean square deviation of 176.61 bar (Fig. 10).

H. Parallel scalability

The parallel scalability of our current multiple-time-step implementations is explored by experimenting from 1 to 8 processors of an SGI Origin3000 computer. Again we see performance that is independent of the system size. The MTS implementation has the same scalability as the original single-time-step method up to 8 processors, which is the best expected given the extra bookkeeping work in multiple-time-step protocols (Fig. 11).

VI. CONCLUSIONS

We have developed efficient and versatile multiple-time-step/particle-mesh-Ewald protocols based on a smooth switch function to regulate direct and reciprocal-space Ewald terms so that the fast component in the reciprocal term is removed. This approach yields improved stability for the medium force, and thus large outer time steps can be used. Long simulations, as done here, are essential to demonstrate this stability; shorter tests, as reported,[20] may not represent behavior accurately. We also use separate switch functions to handle the van der Waals interactions so that the direct force associated with the electrostatic interactions can be further reduced, without compromising the accuracy of van der Waals interactions, as done in Ref. 19. In addition to this

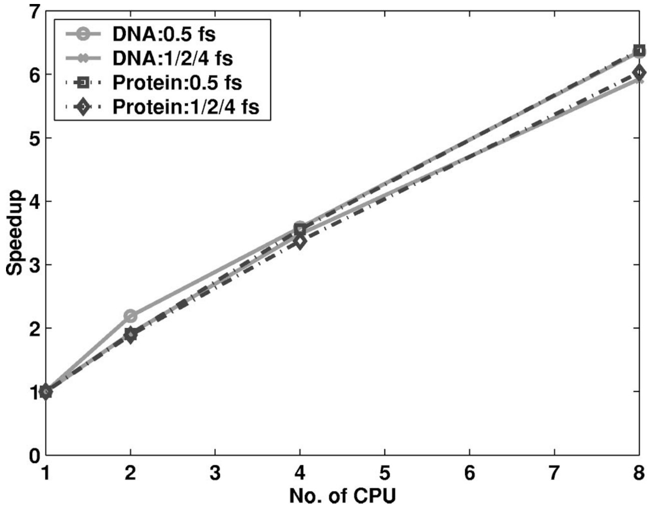

FIG. 11. Parallel scalability for multiple-time-step integrators. All simulations are performed on an SGI Origin3000 computer. Speedups are given as the ratio of computational times (excluding setup times) reported by AMBER6.0 relative to single processor results. The multiple-time-step simulations use protocol g in Table III. A 0.5 fs time step and a 5–9 Å cutoff range are used for single-time-step simulations. All simulations are 1 ps in length.

improvement over our prior implementation,[19] extensions to constant temperature and pressure simulations have been implemented.

For Newtonian dynamics, our optimized protocol (1/3/6 fs with a 3–7 Å switch for electrostatics and 4–8 Å switch for van der Waals forces) yields a speedup of 6.5 relative to 0.5 fs single-time-step simulations; for Langevin dynamics, the optimized protocol (0.8/3.2/16 fs with $\gamma = 5\ \mathrm{ps}^{-1}$) has a speedup factor of 7.5. These values are very close to our estimated maximal achievable speedup values (Fig. 11). The stability and accuracy of temperature and pressure control within our multiple-time-step implementations are verified with long simulations. These program segments are now included in a test version of AMBER and are expected to be released with future AMBER versions [Barash and Schlick (in preparation)].

Further speedup improvements can be achieved by addressing the problem of large cancellation errors in the particle-mesh-Ewald approach,[17] using alternative core functions[52] (that better separate fast and slow interactions and/or push period of numerical cancellation term further), resorting to alternative fast electrostatics methods like multigrid or finite-element approaches,[53] or applying particle-mesh-Ewald-type methods to van der Waals terms to further reduce the nonbonded pairlist.

ACKNOWLEDGMENTS

The work was supported by NSF Award No. ASC-9318159, NIH Award No. R01 GM55164, and a John Simon Guggenheim fellowship to T.S.

APPENDIX A: LINEAR RESONANCE FOR IMPULSE AND EXTRAPOLATION-MULTIPLE-TIME-STEP METHODS AND ESTIMATE OF THRESHOLD γ FOR LANGEVIN DYNAMICS

Following the same notation as Sandu and Schlick,[15] we can develop a more accurate estimate of the resonant spikes in the impulse splitting scheme by analyzing a simple two-class linear system with two force constants $\lambda_1 \gg \lambda_2$, where $\dot{X} = V$ and $\dot{V} = -(\lambda_1 + \lambda_2)X$. The force associated with λ_1 is updated every inner time step $\Delta\tau$ and the force associated with λ_2 is recalculated at every outer time step (which is $k\Delta\tau$ for some integer k). The impulse-multiple-time-step force splitting for VV gives a propagator matrix A_{IV} as defined in the original paper.[15] For most values of k, A_{IV} will have a pair of complex conjugate eigenvalues $x_{1,2}$ as solutions of the equation,

$$x^2 - \mathrm{Tr}(A_{IV})x + \det(A_{IV}) = 0,$$

where $\mathrm{Tr}(A_{IV})$ and $\det(A_{IV})$ denote the trace and determinant of the propagator, respectively. The solutions have the form (we drop the arguments for trace and determinant for simplicity),

$$x_{1,2} = \tfrac{1}{2}\left(\mathrm{Tr} \pm \sqrt{\mathrm{Tr}^2 - 4\ \det}\right).$$

The eigenvalues will approach the real axis and eventually become a real pair for some outer time steps $k\Delta\tau$. The resonance spikes correspond to the extremes of the two real eigenvalues,

$$\frac{\partial x_1}{\partial k} = 0 \quad \text{or} \quad \frac{\partial x_2}{\partial k} = 0.$$

The above two equations can be simplified to give a general formula from which the resonance spikes can be obtained,

$$\left(\frac{\partial\ \det}{\partial k}\right)^2 + \det\left(\frac{\partial\ \mathrm{Tr}}{\partial k}\right)^2 = \mathrm{Tr}\left(\frac{\partial\ \mathrm{Tr}}{\partial k}\right)\left(\frac{\partial\ \det}{\partial k}\right).$$

After algebraic manipulations,[37] we obtain the eigenvalues at resonance spikes as

$$\exp(-\gamma k\Delta\tau/2)\left[\sqrt{1+\left(\frac{k\lambda_2\Delta\tau}{2\sqrt{\lambda_1}}\right)^2} \pm \frac{k\lambda_2\Delta\tau}{2\sqrt{\lambda_1}}\right], \qquad \text{(A1)}$$

where the damping constant γ has a nonzero value for Langevin dynamics or zero for Newtonian dynamics. For stability, γ must be large enough to keep the first spike ($k\Delta\tau \approx T_1/2$) below one. This leads to the following lower bounds for numerical stability:

$$\gamma > \frac{4}{T_1}\ln\left(\frac{\lambda_2\pi}{2\lambda_1} + \sqrt{1+\left(\frac{\lambda_2\pi}{2\lambda_1}\right)^2}\right),$$

for impulse-based multiple-time-step methods. For all variants of extrapolation-multiple-time-step schemes, the first spike is quite similar in magnitude;[15] hence, we can use the estimates for constant extrapolation to derive the lower bound of γ in Langevin dynamics,

$$\gamma \geqslant \frac{4}{T_1}\ln\left(1 + \frac{2\lambda_2}{\lambda_1}\right).$$

Results are shown in Fig. 5 as a function of T_2/T_1, with $T_1 = 11$ fs to mimic biomolecular systems. The corresponding λ_i values for $i = 1,2$ are[15,42] $\lambda_i = (2\pi/T_i)^2$.

APPENDIX B: EQUIVALENCE OF LEAPFROG, VELOCITY, AND POSITION VERLET VARIANTS

Leapfrog, velocity, and position Verlet are all derived from the original discretization formula for advancing positions adopted by Verlet[44] and attributed to Störmer,[45]

$$\mathbf{X}(t+\Delta\tau) = 2\mathbf{X}(t+\Delta\tau) - \mathbf{X}(t-\Delta\tau) + \mathbf{F}(\mathbf{X}(t))(\Delta\tau)^2.$$

The so-called half-step leapfrog[24,43] scheme was proposed to remove numerical roundoff errors resulting from the second-order term $\mathcal{O}((\Delta\tau)^2)$, as well as to give velocity information[54] (Table IV). There is a half-time-step offset between position and velocity updates (hence the name leapfrog). Here we show that they are all equivalent: different formulations of leapfrog will lead to either velocity Verlet or position Verlet.

In the original version of the leapfrog scheme, velocity updates lead each cycle of propagation, so we can call the scheme the *velocity-lead leapfrog* (V-Leap). It is also possible to write the leapfrog scheme with position updates

TABLE IV. Original and modified versions of the velocity-lead (V-Leap) and position-lead leapfrog (P-Leap) Verlet schemes as described in Appendix B. In the modified version of V-Leap, (X_t, V_t) has the same propagation formulas as (X,V) in the original version. The output of X is delayed one time step through the X_t and velocity V is interpolated between two half-time-step updates; this removes the half-time-step offset between position and velocity outputs.

Original V-Leap	**Modified V-Leap**
$X^0 = X_0$	$X_t^0 = X_0$
$F^0 = -\nabla E(X^0)$	$F^0 = -\nabla E(X_t^0)$
$V^0 = V_0 - \frac{\Delta\tau}{2} M^{-1} F^0$	$V_t^0 = V_0 - \frac{\Delta\tau}{2} M^{-1} F^0$
For $n = 0 : NSTEP - 1$	**For** $n = 0 : NSTEP$
$V^{n+1} = V^n + \Delta\tau M^{-1} F^n$	$V_t^{n+1} = V_t^n + \Delta\tau M^{-1} F^n$ (a)
	$X^n = X_t^n$ (b)
	$V^n = \frac{1}{2}(V_t^n + V_t^{n+1})$ (c)
$X^{n+1} = X^n + \Delta\tau V^n$	$X_t^{n+1} = X_t^n + \Delta\tau V_t^{n+1}$ (d)
$F^{n+1} = -\nabla E(X^{n+1})$	$F^{n+1} = -\nabla E(X_t^{n+1})$ (e)
End	**End**
Original P-Leap	**Modified P-Leap**
$V^0 = V_0$	$V_t^0 = V^0$
$X^0 = X_0 - \frac{\Delta\tau}{2} V^0$	$X_t^0 = X_0 - \frac{\Delta\tau}{2} V_t^0$
For $n = 0 : NSTEP - 1$	**For** $n = 0 : NSTEP$
$X^{n+1} = X^n + \Delta\tau V^n$	$X_t^{n+1} = X_t^n + \Delta\tau V_t^n$
$F^{n+1} = -\nabla E(X^{n+1})$	$F^{n+1} = -\nabla E(X_t^{n+1})$
	$X^n = \frac{1}{2}(X_t^n + X_t^{n+1})$
	$V^n = V_t^n$
$V^{n+1} = V^n + \Delta\tau M^{-1} F^{n+1}$	$V_t^{n+1} = V_t^n + \Delta\tau M^{-1} F^{n+1}$
End	**End**

leading the propagation; we call this variant the *position-lead leapfrog* (P-Leap). With extra storage for velocities and positions at different time steps, the original leapfrog scheme (V-Leap, see Table IV) can be rewritten to remove the half-time-step offset. In the modified version, the position/velocity pair $\{X_t, V_t\}$ have the same propagation formulas as $\{X,V\}$ in the original version. The superscripts in Table IV indicate the time intervals from a given initial condition $\{X_0, V_0\}$.

Introducing a force operator $\hat{K}$, where $\hat{K}(X) = -M^{-1}\nabla E(X)$, Eqs. (a), (b), and (c) in Table IV can be written in matrix form as

$$\begin{bmatrix} X^n \\ V^n \end{bmatrix} = \tilde{\mathbf{V}}\left(\frac{\Delta\tau}{2}, \hat{K}\right)\begin{bmatrix} X_t^n \\ V_t^n \end{bmatrix},$$

$$\text{or}\quad \begin{bmatrix} X_t^n \\ V_t^n \end{bmatrix} = \tilde{\mathbf{V}}\left(-\frac{\Delta\tau}{2}, \hat{K}\right)\begin{bmatrix} X^n \\ V^n \end{bmatrix}, \tag{B1}$$

where $\tilde{\mathbf{V}}(\Delta\tau, \hat{K})$ is the velocity propagation operator,

$$\tilde{\mathbf{V}}(\Delta\tau, \hat{K}) = \begin{bmatrix} I & O \\ \Delta\tau\hat{K} & I \end{bmatrix}.$$

Here, I is the identity matrix and O is the zero matrix. It is clear that $\tilde{\mathbf{V}}(t_1, \hat{K})\tilde{\mathbf{V}}(t_2, K) = \tilde{\mathbf{V}}(t_1 + t_2, \hat{K})$, and $\tilde{\mathbf{V}}(0, \hat{K}) = I$. Similarly, for Eqs. (a) and (d) in Table IV, we have

$$\begin{aligned} \begin{bmatrix} X_t^{n+1} \\ V_t^{n+1} \end{bmatrix} &= \tilde{\mathbf{X}}(\Delta\tau)\begin{bmatrix} X_t^n \\ V_t^{n+1} \end{bmatrix} \\ &= \tilde{\mathbf{X}}(\Delta\tau)\tilde{\mathbf{V}}(\Delta\tau, \hat{K})\begin{bmatrix} X_t^n \\ V_t^n \end{bmatrix}, \end{aligned} \tag{B2}$$

where $\tilde{\mathbf{X}}(\Delta\tau)$ is the position propagation operator,

$$\tilde{\mathbf{X}}(\Delta\tau) = \begin{bmatrix} I & \Delta\tau \\ O & I \end{bmatrix}.$$

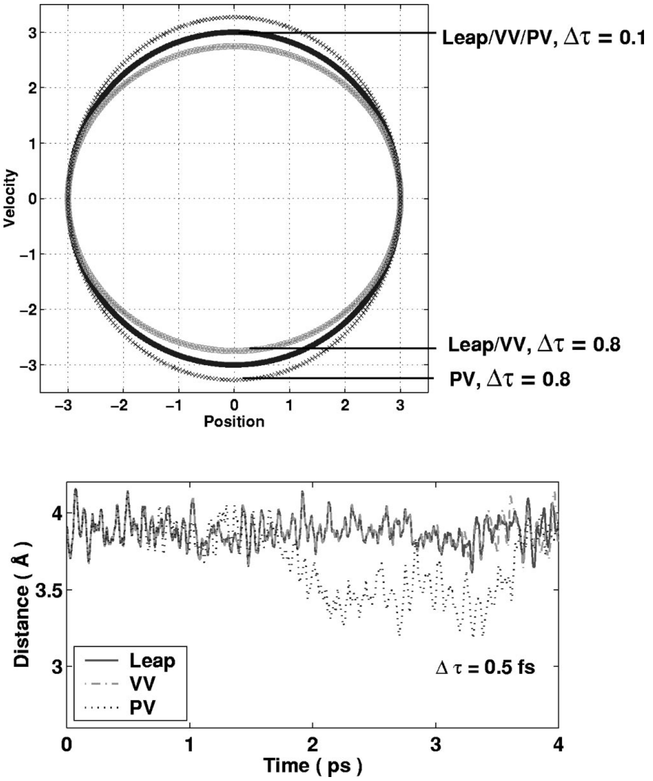

FIG. 12. Comparison of different Verlet schemes (leapfrog, VV, and PV). (Top) linear test case: phase space distributions of a harmonic oscillator. The oscillator has unit mass and unit force constant with an initial position displacement of 3. For a small time-step $(\Delta\tau = 0.1)$, all three Verlet variants generate the correct ensemble as expected from analytical solution (thick middle ring); at larger time step of $\Delta\tau = 0.8$, leapfrog and VV give an ensemble with underscaled velocity (innermost ring), while PV overscales the velocity (outermost ring). (Bottom) nonlinear test case: end-to-end distance between C_β and C_η of the side chain of lysine from a 4 ps simulation with 0.5 fs time step of a single lysine in a large box $(64\times64\times64\ \text{Å}^3)$. All simulations start from the same initial condition. VV and leapfrog yield almost the same trajectory for the first 4 ps, then diverge due to accumulated errors. PV deviates from leapfrog and VV after about 0.6 ps.

Combining Eqs. (B1) and (B2), the propagation matrix from (X^n, V^n) to (X^{n+1}, V^{n+1}) can be expressed as

$$\begin{aligned} \begin{bmatrix} X^{n+1} \\ V^{n+1} \end{bmatrix} &= \tilde{\mathbf{V}}\left(\frac{\Delta\tau}{2}, \hat{K}\right)\begin{bmatrix} X_t^{n+1} \\ V_t^{n+1} \end{bmatrix} \\ &= \tilde{\mathbf{V}}\left(\frac{\Delta\tau}{2}, \hat{K}\right)\tilde{\mathbf{X}}(\Delta\tau)\tilde{\mathbf{V}}(\Delta\tau, \hat{K})\begin{bmatrix} X_t^n \\ V_t^n \end{bmatrix} \\ &= \tilde{\mathbf{V}}\left(\frac{\Delta\tau}{2}, \hat{K}\right)\tilde{\mathbf{X}}(\Delta\tau)\tilde{\mathbf{V}}\left(\frac{\Delta\tau}{2}, \hat{K}\right)\begin{bmatrix} X^n \\ V^n \end{bmatrix}. \end{aligned}$$

This is exactly the same propagator for the VV scheme, and this identity implies that the leapfrog and VV schemes produce identical trajectories for the same initial condition $\{X_0, V_0\}$ in exact arithmetic.

Similarly, it can be shown that the position-lead leapfrog (P-Leap, see Table IV) is equivalent to position Verlet.[37] In practice, the numerical roundoff errors will cause trajectories to diverge after a certain simulation length. Figure 12 shows their similarity for trajectories of a harmonic oscillator as well as a protein model.

[1] T. Schlick, J. Comput. Phys. **151**, 1 (1999) (special volume on Computational Biophysics).
[2] P. Ewald, Ann. Phys. (Leipzig) **64**, 253 (1921).
[3] P. J. Steinbach and B. R. Brooks, J. Comput. Chem. **15**, 667 (1994).
[4] J. Norberg and L. Nilsson, Biophys. J. **79**, 1537 (2000).
[5] T. Darden, D. York, and L. Pedersen, J. Chem. Phys. **98**, 10089 (1993).
[6] W. H. Press, S. A. Teukolsky, W. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, 2nd ed. (Cambridge University Press, Cambridge, 1992), Chap. 12.
[7] J. A. Board, Jr., L. V. Kalé, K. Shulten, R. D. Skeel, and T. Schlick, IEEE Comput. Sci. Eng. **1**, 19 (1994).
[8] Y. Duan and P. Kollman, Science **282**, 740 (1998).
[9] J. Izaguirre, S. Reich, and R. D. Skeel, J. Chem. Phys. **110**, 9853 (1999).
[10] T. Schlick, Structure (London) **9**, R45 (2001).
[11] A. Cheng and Kenneth M. Merz, Jr., J. Phys. Chem. B **103**, 5396 (1999).
[12] P. Procacci, T. Darden, E. Paci, and M. Marchi, J. Comput. Chem. **18**, 1848 (1997).
[13] M. Kawata and M. Mikami, J. Comput. Chem. **21**, 201 (2000).
[14] E. Barth and T. Schlick, J. Chem. Phys. **109**, 1617 (1998).
[15] A. Sandu and T. Schlick, J. Chem. Phys. **81**, 3684 (1984).
[16] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus, J. Comput. Chem. **4**, 187 (1983).
[17] P. Procacci, M. Marchi, and G. J. Martyna, J. Chem. Phys. **108**, 8799 (1998).
[18] P. Procacci, T. Darden, and M. Marchi, J. Phys. Chem. **100**, 10464 (1996).
[19] P. F. Batcho, D. A. Case, and T. Schlick, J. Chem. Phys. **115**, 4003 (2001).
[20] R. Zhou, E. Harder, H. Xu, and B. Berne, J. Chem. Phys. **115**, 2348 (2001).
[21] D. A. Pearlman, D. A. Case, J. W. Caldwell, W. S. Ross, T. E. Cheatham, III, S. DeBolt, D. Ferguson, G. L. Seibel, and P. A. Kollman, Comput. Phys. Commun. **91**, 1 (1995).
[22] H. Berendsen, J. Postma, A. Dinola, and J. Haak, J. Chem. Phys. **81**, 3684 (1984).
[23] D. der Spoel, A. van Buuren, E. Apol, P. M. D. Tieleman, L. Sijbers, B. Hess, K. Feenstra, E. Lindahl, R. van Drunen, and H. Berendsen, Gromacs User Manual version 2.0, Nijenborgh 4, 9747 AG Groningen, The Netherlands. Internet: http://md.chem.rug.nl/~gmx, 1999.
[24] R. W. Hockney and J. W. Eastwood, *Computational Simulation Using Particles* (McGraw–Hill, New York, 1981).
[25] A. Appel, SIAM (Soc. Ind. Appl. Math.) J. Sci. Stat. Comput. **6**, 85 (1985).
[26] J. E. Barnes and P. Hut, Nature (London) **324**, 446 (1986).
[27] L. Greengard and V. Rokhlin, J. Comput. Phys. **73**, 325 (1987).
[28] R. Zhou and B. J. Berne, J. Phys. Chem. **103**, 9444 (1995).
[29] C. Kittel, *Introduction to Solid State Physics* (Wiley, New York, 1971).
[30] S. W. DeLeeuw, J. W. Perram, and E. R. Smith, Proc. R. Soc. London, Ser. A **373**, 27 (1980).
[31] M. Deserno and C. Holm, J. Chem. Phys. **109**, 7678 (1998).
[32] A. Toukmaji and J. A. Board, Comput. Phys. Commun. **95**, 81 (1996).
[33] P. Procacci and M. Marchi, J. Chem. Phys. **104**, 3003 (1996).
[34] P. Procacci and B. J. Berne, J. Chem. Phys. **101**, 2421 (1994).
[35] D. D. Humphreys, R. A. Friesner, and B. J. Berne, J. Phys. Chem. **98**, 6885 (1994).
[36] G. Martyna, M. Tuckerman, D. Tobias, and M. Klein, Mol. Phys. **87**, 1117 (1996).
[37] X. Qian, Ph.D. thesis, New York University, 2002.
[38] M. Tuckerman, B. J. Berne, and G. J. Martyna, J. Chem. Phys. **94**, 6811 (1991).
[39] E. Barth and T. Schlick, J. Phys. Chem. **109**, 1633 (1998).
[40] P. F. Batcho and T. Schlick, J. Comput. Phys. **115**, 4019 (2001).
[41] R. Skeel, G. Zhang, and T. Schlick, SIAM J. Sci. Comput. (USA) **18**, 203 (1997).
[42] T. Schlick, *Molecular Modeling and Simulation: An Interdisciplinary Guide* (Springer-Verlag, New York, 2002), Chaps. 12 and 13 (in press).
[43] R. W. Hockney, Methods Comput. Phys. **9**, 136 (1970).
[44] L. Verlet, Phys. Rev. **159**, 98 (1967).
[45] C. Störmer, Arch. Sci. Phys. Nat. **24**, 5 (1911).
[46] J. Ryckaert, G. Ciccotti, and H. Berendsen, J. Comput. Phys. **23**, 327 (1977).
[47] X. Qian, D. Strahs, and T. Schlick, J. Mol. Biol. **308**, 681 (2001).
[48] M. Watanabe and M. Karplus, J. Chem. Phys. **99**, 8063 (1993).
[49] F. Figueirido, R. Levy, R. Zhou, and B. Berne, J. Chem. Phys. **106**, 9835 (1997).
[50] M. Watanabe and M. Karplus, J. Phys. Chem. **99**, 5680 (1995).
[51] U. Essmann, M. Perera, M. Berkowitz, T. Darden, H. Lee, and L. Pedersen, J. Chem. Phys. **103**, 8577 (1995).
[52] P. F. Batcho and T. Schlick, J. Chem. Phys. **115**, 8312 (2001).
[53] C. Sagui and T. Darden, J. Chem. Phys. **114**, 6578 (2001).
[54] M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids* (Oxford University Press, New York, 1987), Chap. 3.