Overcoming stability limitations in biomolecular dynamics. I. Combining force splitting via extrapolation with Langevin dynamics in LN

Eric Barth[a] and Tamar Schlick[b]
Department of Chemistry and Courant Institute of Mathematical Sciences, New York University and Howard Hughes Medical Institute, 251 Mercer Street, New York, New York 10012

(Received 22 July 1997; accepted 17 February 1998)

We present an efficient new method termed LN for propagating biomolecular dynamics according to the Langevin equation that arose fortuitously upon analysis of the range of harmonic validity of our normal-mode scheme LIN. LN combines force linearization with force splitting techniques and disposes of LIN's computationally intensive minimization (anharmonic correction) component. Unlike the competitive multiple-timestepping (MTS) schemes today—formulated to be symplectic and time-reversible—LN merges the slow and fast forces via extrapolation rather than ''impulses;'' the Langevin heat bath prevents systematic energy drifts. This combination succeeds in achieving more significant speedups than these MTS methods which are limited by resonance artifacts to an outer timestep less than some integer multiple of half the period of the fastest motion (around 4–5 fs for biomolecules). We show that LN achieves very good agreement with small-timestep solutions of the Langevin equation in terms of thermodynamics (energy means and variances), geometry, and dynamics (spectral densities) for two proteins in vacuum and a large water system. Significantly, the frequency of updating the slow forces extends to 48 fs or more, resulting in speedup factors exceeding 10. The implementation of LN in any program that employs force-splitting computations is straightforward, with only partial second-derivative information required, as well as sparse Hessian/vector multiplication routines. The linearization part of LN could even be replaced by direct evaluation of the fast components. The application of LN to biomolecular dynamics is well suited for configurational sampling, thermodynamic, and structural questions. © *1998 American Institute of Physics.* [S0021-9606(98)50220-6]

I. INTRODUCTION

The increasing availability and speed of high-end workstations has made molecular dynamics simulations a powerful desktop resource. Although not yet considered a full partner to experiment for studying molecular configuration and function, the numerically generated sequence of molecular configurations—obeying Newtonian physics—offers insights into molecular flexibility and thermodynamic processes. Besides the uncertainty in force fields, this subordinate status of molecular dynamics to instrumentation can be attributed to the relatively short (e.g., nanosecond) trajectory lengths that can be simulated due to both computer hardware and software limitations.

In biomolecular simulations, computational cost is dominated by the frequent evaluation of the potential energy function and its gradient. Typically, one million steps are required to simulate a nanosecond, with each step entailing several seconds of computing for a large system. With simplifications of the simulation protocol (e.g., reduction of the nonbonded interactions considered), the cost per step can be reduced to a fraction of a second, but a nanosecond simulation of a 20,000-atom system on a 300-teraflop machine still requires several days.[1] Unfortunately, the steps cannot be lengthened arbitrarily, not so much due to loss of accuracy as far as thermodynamics and sampling are concerned, but because of the numerical stability of the integrator.

Stability is limited by the molecular high-frequency vibrational modes, or bond stretching. The fastest period P relevant to biomolecules is around 10 fs (associated with a wavelength of absorption around 3400 cm^{-1} for O–H and N–H stretching, for example). Resolving these fast motions adequately dictates timesteps of 0.5 fs or less ($P/20$). Recent research has shown that these short periods also limit the timesteps that can be used successfully in separating frameworks to update the *slow* forces (i.e., outer timestep). This limitation is removed from the method we describe here, with an analysis of the underlying theory presented in paper II.[2]

Integration schemes for biomolecular dynamics exploit the spatial locality and linear complexity of the fastest components of the force, in contrast to the slow and long-range interactions, which grow in number as the square of the number of atoms. As in other scientific applications of multilevel or multiscale techniques,[3] the temporal and spatial scales of the model are efficiently connected;[4–7] see Ref. 8 for a recent review. Small timesteps ($\Delta\tau$) are used to resolve the fastest vibrational modes, but only the inexpensive local interactions are updated at each small step; the costly long-range forces are updated at appropriately chosen longer (Δt) time intervals.

The Langevin-Implicit/Normal mode scheme (LIN) and multiple timestep (MTS) methods are two such multiscale ap-

[a]Present address: Department of Mathematics, Kalamazoo College, 1200 Academy Street, Kalamazoo, MI 49006.
[b]Corresponding author; electronic mail: schlick@nyu.edu

proaches. LIN resolves the fast motion by linearization of the equations of motion (and obtains the residual motion by implicit integration). MTS schemes update the fast and slow motions at different frequencies. Results to date have shown that LIN permits comparatively long timesteps (15 fs) and reproduces well small-timestep behavior,[9] but the computational expense of each step (due to minimization) yields only modest overall efficiency gains compared to small-timestep methods. Recent experiments show that the popular MTS methods r-Respa and Verlet-I only permit outer timesteps of 5 fs or less for biomolecules based on accuracy criteria and yield a speedup factor of around 5 (in comparison to explicit simulations at timesteps of 0.5 fs);[10,11] this threshold for the outer timestep in these schemes has recently been explained by artificial (i.e., integrator-induced) resonance[12–15] occurring at multiples of half the period of the fastest bond vibrations. A ''resonance'' is a numerical artifact of the integrator appearing at special timesteps related to the periods inherent in the motion; large energy fluctuations or instability are seen at those timesteps.[13] See Ref. 13 for analysis of resonance on a Morse oscillator for the symplectic implicit-midpoint scheme and Ref. 14 for a discussion of nonlinear resonance for a family of symplectic methods for molecular dynamics.

A definition for a ''large timestep'' method has been suggested[16] as a scheme which overcomes this half-period barrier. LIN is one such method, but it is not as competitive in terms of CPU time as are MTS schemes. A mollified impulse method is also being developed[16] to extend the timestep slightly beyond 5 fs. The efficient LN approach, described here for the first time with the force splitting component, overcomes this barrier significantly.

In section II, we briefly describe existing multiscale methods for Newtonian and Langevin dynamics, including Verlet, impulse MTS, BBK, and LIN. These descriptions lay the groundwork for our resonance discussion and the LN results included here. In section III we present LN and discuss advantages and limitations of the former schemes as motivation for the new method. See Ref. 17 for a summary of LN's predecessors and a historical perspective of this algorithmic work.

In section IV, we present the results of LN simulations for several systems and discuss two key issues: (1) performance of LN—measured by agreement with traditional small-timestep methods in terms of thermodynamic, structural, and dynamic properties—as a function of the timestep triplet used; (2) efficiency, measured by overall computational speedup, compared with traditional methods. As assessed in MTS works, efficiency and reliability are measured in comparison to explicit simulations using a 0.5 fs timestep; connections to experiment are made where possible (e.g., spectral density functions and radial distribution functions for water).

We conclude by highlighting LN's ability to overcome severe resonance effects that limit symplectic MTS methods; a linear analysis follows in the companion paper.[2] We show that LN can use an outer timestep such as 50 fs while still giving excellent agreement to a trajectory using a much smaller outer timestep; the computational savings can exceed an order of magnitude.

II. MULTISCALE METHODS

Traditional models for molecular dynamics are guided by the classical equations of motion

$$\dot{X}=V,\ \dot{V}=-\mathbf{M}^{-1}(\nabla E_{\text{fast}}(X)+\nabla E_{\text{mid}}(X)+\nabla E_{\text{slow}}(X)), \tag{1}$$

where X and V are the collective position and velocity vectors, respectively, $\mathbf{M}$ is the diagonal mass matrix, and the dot superscripts denote differentiation with respect to time. The gradient vector (∇E) of the empirical potential energy function E is separated above (for future reference) into fast, medium (or middle), and slow components. For biomolecules, this three-part splitting is made according to the bonded, short-range nonbonded, and long-range nonbonded interactions (detailed below). The associated timesteps $\Delta\tau$, Δt_m, and Δt designate, respectively, the shortest (innermost) (e.g., 0.5 fs), medium (an integer multiple of $\Delta\tau$: $\Delta t_m = k_1\Delta\tau$), and largest (outermost) discretization stepsizes (an integer multiple of Δt_m: $\Delta t = k_2\Delta t_m = k_1k_2\Delta\tau$). The middle timestep ranges from 1 to 3 fs in our implementation and can be thought of as an adjustable value. The ratio $r=\Delta t/\Delta\tau = k_1k_2$ influences the speedup of force-splitting schemes relative to a simulation at the constant, small timestep $\Delta\tau$. In symplectic MTS schemes resonance limits r to less than 10 when $\Delta\tau=0.5$ fs, with asymptotic speedups around 5 (see section IV D).

A. Verlet and impulse multiple timestep methods

The standard (single-timestep) integrator introduced by Verlet[18] completes the updating sweep for positions and velocities by the following triplet:

Velocity Verlet algorithm

$$\begin{aligned} V^{n+1/2} &= V^n - \mathbf{M}^{-1}\frac{\Delta\tau}{2}\nabla E(X^n), \\ X^{n+1} &= X^n + \Delta\tau V^{n+1/2}, \\ V^{n+1} &= V^{n+1/2} - \mathbf{M}^{-1}\frac{\Delta\tau}{2}\nabla E(X^{n+1}). \end{aligned} \tag{2}$$

This second-order method preserves geometric properties (such as time reversibility and phase-space volume) present in the exact solution of these equations for conservative Hamiltonian systems.[19] This symplecticness likely accounts for the favorable energy preservation along computed trajectories. However, for acceptable resolution of the fastest component, the Verlet timestep should be in the range $P/20$ to $P/10$, or $\Delta\tau$ from 0.5 to 1.0 fs, much less than the linear stability limit[8,14] of P/π.

The MTS methods introduced two decades ago[20,21] became increasingly common for biomolecular simulations with the introduction of variants that shared the time-reversal symmetry of the Verlet scheme. Schulten and co-workers[5] introduced ''*Verlet–I*'' and, independently, Tuckerman *et al.*[4] derived the equivalent but more general method ''*r-Respa*'' by applying the Verlet method [system (2)] to the Trotter factorization of the Liouville operator with components corresponding to potential splitting (E_{fast}, E_{mid}, E_{slow}). A sweep over one Δt interval by a Verlet-based MTS method

can thus be written as the following double-nested iteration process:

Impulse MTS algorithm

$$V \leftarrow V - \mathbf{M}^{-1}\,\frac{\Delta t}{2}\nabla E_{\text{slow}}(X)$$

for $j=0,\ k_2-1,$

$$V \leftarrow V - \mathbf{M}^{-1}\,\frac{\Delta t_m}{2}\,\nabla E_{\text{mid}}(X)$$

for $i=0,\ k_1-1,$

$$V \leftarrow V - \mathbf{M}^{-1}\frac{\Delta \tau}{2}\nabla E_{\text{fast}}(X),$$

$$X \leftarrow X + \Delta \tau V, \tag{3}$$

$$V \leftarrow V - \mathbf{M}^{-1}\,\frac{\Delta \tau}{2}\nabla E_{\text{fast}}(X),$$

end

$$V \leftarrow V - \mathbf{M}^{-1}\,\frac{\Delta t_m}{2}\,\nabla E_{\text{mid}}(X),$$

end

$$V \leftarrow V - \mathbf{M}^{-1}\frac{\Delta t}{2}\,\nabla E_{\text{slow}}(X).$$

Note that the application of the slow force components (∇E_{slow}) results in an *impulse*: The velocities are modified by a term proportional to $k_1 k_2 \Delta\tau$—r times larger than the changes made to X and V in the inner loop—only outside of the inner loop (i.e., at the onset and at the end of a sweep covering Δt). This impulse yields undesirable resonance effects in simple oscillator systems;[12] see also the related detailed analysis of Ref. 13. In biomolecular systems, the first such resonance can occur at about $\Delta t = 5$ fs—half the period of the fastest oscillation. Although not explained by resonance in these papers, large energy growth has been reported beyond this threshold for impulse MTS simulations.[10,11] Thus, like the single-timestep Verlet, impulse MTS methods are limited by numerical considerations; even with a small inner timestep, the outer timestep is limited by resonance. The precise limit on the outer timestep depends sensitively on implementational details of the MTS scheme (e.g., number of force classes, switching functions expressions, class-update frequency, and more). Thus, in contrast to the above reports, other studies have used larger outer timesteps in combination with Berendsen thermostats or fast Ewald interactions, for example.

We emphasize that non-impulse MTS methods have been earlier proposed.[5,20–23] These are much less sensitive to resonance but realize an energy drift due to their nonsymplectic nature. Our extrapolation alternative to the impulse formulation in LN does not suffer so severely from these artifacts when coupled with the Langevin heat bath. Consequently, we can increase the outer timestep significantly, while ensuring a stable trajectory. Larger speedup factors are then reaped by economizing on the slow-force computations, as the cost of the fast and medium forces remains essentially the same.

B. Langevin dynamics and the linearization approach of LN's predecessor LIN

Although used by us for numerical (stability) considerations,[6,7,9,24] a stochastic component has also been added to models for biomolecular simulations to eliminate explicit representation of water molecules,[25] enhance sampling,[26–28] treat droplet surface effects,[29,30] and represent hydration shell models in large systems.[31–33] In the simplest form of the phenomenological Langevin equation,[34] the equations of motion in system (1) are modified by friction and random force terms to yield

$$\dot{X}(t) = V(t), \quad \mathbf{M}\dot{V}(t) = -\nabla E(X(t)) - \gamma \mathbf{M} V(t) + R(t), \tag{4a}$$

where γ is the collision parameter. The random-force vector R is a stationary Gaussian process with statistical properties (mean and covariance matrix) given by

$$\langle R(t)\rangle = 0, \quad \langle R(t)R(t')^T\rangle = 2\gamma k_B \mathrm{T}\mathbf{M}\delta(t-t'), \tag{4b}$$

where k_B is the Boltzmann constant, T is the target temperature (italics superscripts ^{T} are reserved for vector/matrix transposes), and δ is the usual Dirac symbol. The generalization of Verlet to Langevin dynamics is typically used in the following form described by Brooks, Brünger and Karplus (''BBK''):[29,35]

BBK **Algorithm**

$$V^{n+1/2} = V^n + \mathbf{M}^{-1}\frac{\Delta\tau}{2}(-\nabla E(X^n) - \gamma \mathbf{M} V^n + R^n),$$

$$X^{n+1} = X^n + \Delta\tau V^{n+1/2}, \tag{5}$$

$$V^{n+1} = V^{n+1/2} + \mathbf{M}^{-1}\frac{\Delta\tau}{2}(-\nabla E(X^{n+1}) - \gamma \mathbf{M} V^{n+1} + R^{n+1}).$$

In this formulation, the third equation above is implicit for V^{n+1}, but the linear dependency allows solution for V^{n+1} in closed form.

LIN, the predecessor of LN,[6,7,9] successfully crosses the half-period timestep barrier with timesteps of 15 fs while producing trajectories which agree closely with standard integrators.[9] Agreement has been assessed by thermodynamic averages and geometric properties within 1% or 2% difference.[9]

The first part of LIN involves the solution of the *linearized* Langevin equation at some reference point X_r:

$$\dot{X} = V, \quad \mathbf{M}\dot{V} = -\nabla E(X_r) - \tilde{\mathbf{H}}(X - X_r) - \gamma \mathbf{M} V + R, \tag{6}$$

where the matrix $\tilde{\mathbf{H}}$ is a sparse approximation to the Hessian of E at X_r.[9] We have used the Hessian resulting from short (e.g., 4.5 Å) cutoffs[9] or the second derivatives coming from the bond-length, bond-angle, dihedral-angle and the 1–4 electrostatic components. Our experience has shown that the second choice is much easier to implement in practice (to exploit sparsity), though the first choice might be preferred if cost were not an issue since the interval over which the linearization is retained can be lengthened.[9]

The solution $X_h(t)$ of the linearized equations of motion—the ''harmonic component''—can be approximated by one of the discretization schemes given in systems (7) or (8) below (described for LN). This numerical solution is an efficient alternative to the costly analytic approach, by normal mode analysis, over time intervals that are not too large, like 15 fs (see computational details in Refs. 6 and 17). While in theory the numeric procedure can become unstable if vibrational modes with negative eigenvalues λ are encountered (corresponding to solutions $\exp(-i\sqrt{\lambda}t)$, where i, here only, is the complex number $\sqrt{-1}$), we have not encountered such problems in practice for reasonable choices of Δt and $\Delta\tau$. In our first suggestion of this numeric approach,[9] we have also described a procedure for approximating the negative eigenvalues and the corresponding eigenvectors (e.g., by Lanczos-based techniques), projecting the corresponding imaginary frequencies, and then solving Eq. (6) by numerical integration. Fortunately, we were not forced to resort to this approach for both LIN and LN with careful parameter choices.

The second part of LIN relies on implicit integration to compute the residual component, $Z(t)$, with a large timestep. Specifically, $Z(t)$ can be determined by solving the new set of equations which Z satisfies. These are determined by using $Z=X-X_h$ and the origin of X and X_h as solutions of systems (4) and (6), respectively. This leads to the equations[6] for Z and its time derivative W, as summarized in the Appendix.

Through a combination of efficient algorithms for the subintegration and minimization processes, we have achieved computational speedup for the LIN outer step of 15 fs, but the factor is modest,[9] about 1.5. Nonetheless, LIN is a true long timestep method.[16]

In the remainder of the paper we present the new method LN which combines the efficiency of impulse MTS methods with the long timestep stability of LIN to successfully overcome the half-period barrier of MTS methods.

III. THE LN METHOD

The idea of LN is to eliminate LIN's implicit integration component and, concomitantly, reduce the interval length over which the harmonic model is followed. This works because anharmonic corrections are very small for a linearization-updating frequency of 5 fs or less.[9,17] Thus, LN approximates the linearized model for the equations of motion every Δt_m (e.g., 1–3 fs) interval, and explicitly integrates the linearized system using an inner timestep $\Delta\tau$ (such as 0.5 fs). This inner timestep parallels the timestep used to update the fast motion in MTS, and the linearization frequency is analogous to the medium timestep used in a three-interval MTS scheme. The subintegration in LN of the linearized model does not require new force evaluations, as in every step of standard molecular dynamics integration. Since, in addition, the timestep of slow-force updates can be lengthened significantly, LN is computationally competitive.

LN begins with the linear approximation to the Langevin equation at some reference position X_r (e.g., the previous position, X^n, or a midpoint, $X^{n+1/2}$). System (6) is then solved numerically, at a small ''inner'' timestep $\Delta\tau$, as mentioned above for LIN (see note above on the theoretical caveat). For the explicit integration of system (6), we have explored two possibilities. The first uses X^n as a reference point, in combination with the *second-order partitioned Runge-Kutta method* (''Lobatto IIIa,b''),[36] which reduces to the velocity Verlet method [system (2)] when $\gamma=0$. This yields the following iteration process for $\{X^{n+1},V^{n+1}\}$ from the initial conditions $X^0=X^n$, $V^0=V^n$:

$$
\begin{aligned}
V^{i+1/2} &= V^i+\frac{\Delta\tau}{2}\,\mathbf{M}^{-1}[-\nabla E(X_r)-\tilde{\mathbf{H}}(X_r)(X^i-X_r)\\
&\qquad -\gamma\mathbf{M}V^{i+1/2}+R^i],\\
X^{i+1} &= X^i+\Delta\tau V^{i+1/2},\\
V^{i+1} &= V^{i+1/2}+\frac{\Delta\tau}{2}\,\mathbf{M}^{-1}[-\nabla E(X_r)-\tilde{\mathbf{H}}(X_r)\\
&\qquad \times(X^{i+1}-X_r)-\gamma\mathbf{M}V^{i+1/2}+R^{i+1}].
\end{aligned}
\tag{7}
$$

In slight contrast to the BBK method of system (5), the third equation above is explicit, but the first equation is linearly implicit for $V^{i+1/2}$.

The second subintegration scheme uses position Verlet rather than velocity Verlet, and defines the inner iteration process of LN by

$$
\begin{aligned}
X^{i+1/2} &= X^i+\frac{\Delta\tau}{2}V^i,\\
V^{i+1} &= V^i+\Delta\tau\mathbf{M}^{-1}[-\nabla E(X_r)-\tilde{\mathbf{H}}(X^{i+1/2}-X_r)\\
&\qquad -\gamma\mathbf{M}V^{i+1}+R^i],\\
X^{i+1} &= X^{i+1/2}+\frac{\Delta\tau}{2}V^{i+1}.
\end{aligned}
\tag{8}
$$

Note the Hessian/vector products in the first and third equations of system (7) and the second equation of system (8). The random force is updated according to equation (4b) at every $\Delta\tau$ substep, so there is no problem of thermal equilibrium as for larger timesteps.[7]

Thus, the skeletal LN procedure is a dual timestep scheme ($\{\Delta\tau,\Delta t_m\}$) which consists of two practical tasks: (a) constructing the Hessian $\tilde{\mathbf{H}}$ in system (6) every Δt_m interval, and (b) solving system (6), where R is given by equation (4b), at the timestep $\Delta\tau$ by procedures (7) or (8). When a force-splitting procedure is also applied to LN, a value $\Delta t > \Delta t_m$ is used to update the slow forces less often than the linearized model.

Our experience has shown that a sparse $\tilde{\mathbf{H}}$ resulting from 4.5 Å cutoffs evaluated every 5 fs gives very similar results in comparison to explicit trajectories at 0.5 fs.[9] We also reported[8] that a sparser $\tilde{\mathbf{H}}$ including only bonded interactions (i.e., bond-length, bond-angle and dihedral-angle) can be used together with a 3 fs timestep for similar computational gains, as shown in Table I, which increase with system size. Detailed comparisons are shown in the next section.

The length of the parameter Δt_m is limited by two factors: the time interval over which the linearized forces are adequate, and the fundamental timescale of the interactions

TABLE I. Timings for LN without (top) and with (bottom) force splitting. The reference timestep in BBK is $\Delta\tau=0.5$ fs and the sparse $\tilde{\mathbf{H}}$ of LN includes bond-length, bond-angle, dihedral-angle and 1–4 electrostatic terms. For the top part of the table, $T_{\nabla E}$, $T_{\tilde{\mathbf{H}}}$, and $T_{\tilde{\mathbf{H}}d}$ denote, respectively, the time for one gradient evaluation, sparse Hessian evaluation, and one sparse-Hessian/vector multiplication. The percentages that follow these quantities represent the CPU percentages required for that task during an entire trajectory. The sparsity of $\tilde{\mathbf{H}}$ is the ratio of the number of nonzero entries in the upper triangle (diagonals included) to the total number of elements in the upper triangle [$N(N+1)/2$ entries for an $N\times N$ matrix]. The interval $\Delta t(=\Delta t_m)$ in the top part of the table is 3 fs except for water where it is 1 fs. In the lower part, times are recorded during the 60 ps production simulations, and reflect force splitting overheads. Thus, evaluating each component of the force is slightly more expensive than in the ideal case, reflected in the top portion of the table. Here we show the evaluation times for the three parts of the forces: fast (same terms associated with $\tilde{\mathbf{H}}$), middle (within 6 Å radius) and slow (outside the 6 Å spherical region). Then for BBK and each LN variant, we show the percentage of total CPU time spent on each such component (the subscript in the $\%_{\text{fast}}$ column refers to the percentage for $\tilde{\mathbf{H}}$ evaluation): With force splitting the time spent on the slow components decreases, but the percentage of time spent on the fast and medium forces increases. The values of Δt_m are as above, and $\Delta t=k_2\Delta t_m$ where $k_2=3$, 24, and 48 for the LN variants. For the reference water simulation with 12 Å cutoffs, $T_{\nabla E}$ in the upper portion includes the cost per step of constructing the nonbonded pair list; in the lower portion the corresponding $T_{\nabla E_{\text{slow}}}$ value (6.28) includes time for evaluating interactions within the 12 Å cutoff region (2.78 s) and time for constructing the nonbonded pair list (3.5 s); for LN 3, where the list is updated every 8 Δt steps, the cost of the long-range forces is taken to be $2.78+(3.5/8)=3.22$ s. All timings are taken from the simulations described in the text, run in serial mode on a 195 MHz SGI R10000 Power Challenge computer.

A. LN 1 (without force splitting)

	Atoms/ Variables	$\tilde{\mathbf{H}}$ sparsity	$T_{\nabla E}$	$T_{\tilde{\mathbf{H}}}$	$T_{\tilde{\mathbf{H}}d}$	BBK CPU/Δt	LN CPU/Δt	LN Speedup
Dipeptide	22/66	0.459	4.7e-4 (22%)	1.2e-3 (57%)	4.2e-5 (18%)	2.8e-3	2.1e-3	1.3
BPTI	904/2712	0.012	0.38 (77%)	0.11 (18%)	0.002 (5%)	2.35	0.531	4.4
Lysozyme	2030/6090	0.005	1.98 (81%)	0.45 (16%)	0.005 (3%)	12.6	2.53	5.0
Water	12393/37179	0.0003	53.5 (99%)	0.07 (0.3%)	0.12 (0.7%)	106.9	53.6	2.0
Water [cutoff]	12393/37179	0.0003	3.0 (91%)	0.07 (2%)	0.12 (7%)	6.6	3.6	1.8

B. LN with force splitting

	$T_{\nabla E_{\text{fast}}}$	$T_{\tilde{\mathbf{H}}}$	$T_{\nabla E_{\text{mid}}}$	$T_{\nabla E_{\text{slow}}}$	LN 3			LN 24			LN 48			LN Speedup		
					$\%_{\text{fast}}$	$\%_{\text{med}}$	$\%_{\text{slow}}$	$\%_{\text{fast}}$	$\%_{\text{med}}$	$\%_{\text{slow}}$	$\%_{\text{fast}}$	$\%_{\text{med}}$	$\%_{\text{slow}}$	LN 3	LN 24	LN 48
BPTI	0.01	0.11	0.05	0.43	$24_{(15)}$	33	43	$38_{(24)}$	53	9	$40_{(25)}$	55	5	7.4	11.7	12.6
Lysozyme	0.02	0.45	0.12	2.2	$15_{(9)}$	32	53	$28_{(16)}$	60	12	$30_{(17)}$	64	6	9.3	17.2	18.6
Water	0.02	0.07	0.39	53.5	$1_{(0)}$	2	97	$7_{(3)}$	14	79	$12_{(4)}$	23	65	5.4	32.6	48.0
Water [cutoff]	0.02	0.07	0.39	6.28	$17_{(14)}$	22	61	$32_{(8)}$	41	27	$37_{(9)}$	47	16	3.4	5.4	7.1

not included in the approximate linearization. In large systems, the long-range interactions—excluded from the linearization—have a fundamental timescale much longer than the interval over which the linearized forces are adequate. Thus they can be updated less frequently than the approximate linearization. This observation suggests that the LN idea can be naturally extended by continuing the splitting process: The bonded interactions are treated by the approximate linearization, the local nonbonded interactions are treated by constant extrapolation over timesteps of length Δt_m, and the nonlocal interactions are treated by constant extrapolation over much longer intervals Δt (i.e., $\Delta t=k_2\Delta t_m$ for some integer $k_2>1$). Thus, like the impulse MTS methods, the force $\nabla E(X_r)$ is evaluated once per Δt step but, unlike these schemes, LN applies this force *at each inner timestep* $\Delta\tau$. This constant extrapolation over the outer timestep does not produce an impulse. The splitting of the nonbonded interaction into short and long-range parts can be accomplished using a smooth switching function,[23] discussed in section IV.

Formally, the LN method with the above force splitting for the triplet protocol $\{\Delta\tau,\Delta t_m,\Delta t\}$ is given as follows:

LN **algorithm**

$$X_r=X+\frac{\Delta t_m}{2}V,$$

$$F_s=-\nabla E_{\text{slow}}(X_r)$$

for $j=1,\ k_2$

$$X_r=X+\frac{\Delta t_m}{2}V,$$

$$\tilde{\mathbf{H}}=\tilde{\mathbf{H}}(X_r),$$

$$F_m=-\nabla E_{\text{mid}}(X_r)-\nabla E_{\text{fast}}(X_r),$$

$$F=F_m+F_s$$

for $i=1,\ k_1$

evaluate R

$$X\leftarrow X+\frac{\Delta\tau}{2}V,$$

$$V\leftarrow(V+\mathbf{M}^{-1}\Delta\tau(F-\tilde{\mathbf{H}}[X-X_r]+R))/(1+\gamma\Delta\tau),$$

$$X\leftarrow X+\frac{\Delta\tau}{2}V$$

end

end

IV. RESULTS

We use for LN the sparse Hessian matrix $\tilde{\mathbf{H}}$ assembled from all bonded interactions (bond lengths, bond and dihedral angles) as well as the 1–4 electrostatic interactions (i.e., those among atom pairs separated by three bonds). The inner timestep for LN is $\Delta\tau=0.5$ fs, and the same value is used in the reference simulations of BBK. The middle timestep in LN

is $\Delta t_m = 3$ fs except for water, where it is 1 fs; this timestep can be considered an adjustable quantity when force splitting is used. When force splitting is not used (as for the small model amino acid system), this middle timestep is also the outer timestep ($\Delta t_m = \Delta t,\ k_2 = 1$). In our force-splitting notation below, this LN variant is designated as ''LN 1.''

In the presence of force splitting in LN, we adopt the notation LN 2, LN 3, etc., to indicate the frequency (k_2) for updating the slow forces relative to Δt_m. Namely, at these integral multiples of Δt_m, the nonbonded pairwise interactions at separations $r > r_0$ are calculated, and the nonbonded pair list is updated for local ($r < r_0$) interactions. In this work, we use the value $r_0 = 6$ Å. $\tilde{\mathbf{H}}$ is updated each Δt_m along with the local nonbonded interactions associated with atom pairs at separations $r < r_0$. Thus, for example, the LN 3 scheme updates the local nonbonded interactions and $\tilde{\mathbf{H}}$ every Δt_m but interactions outside of the $r_0 = 6$ Å range every $3\Delta t_m$.

The local interactions need not be truncated abruptly at r_0; rather, we can use a potential switching function[23] over a buffer region of length b as follows:

$$S(r) = \begin{cases} 1, & r < r_0 - b \\ 1 + R^2(2R-3), & r_0 - b \leq r < r_0, \\ 0, & r_0 < r \end{cases} \tag{9}$$

where $R = (r - (r_0 - b))/b$. The short and long-range forces F_{short} and F_{long} for each atom pair are then defined from the full force F by

$$F_{\text{short}} = FS(r), \quad F_{\text{long}} = F - F_{\text{short}}.$$

For the proteins, we used atomic-based cutoffs with a switching function using $b = 3$ Å, and for the water droplet we used group-based cutoffs with no switching (i.e., $b = R = 0$). We use the unit dielectric constant and include *all* nonbonded interactions for the modeled systems, except when stated otherwise.

The second subintegration scheme [equation (8)] is used in all LN simulations with a collision frequency of $\gamma = 20$ or 5 ps^{-1}. Our motivation is to use as small a γ as possible to weigh the stochastic terms minimally (relative to the inertial terms) but at the same time guarantee numerical stability of the method. Clearly, if γ is too small, LN is unstable. We found the midpoint version of equation (8) to tolerate better lower γ; there are also theoretical justifications.[2] Since the gradient and Hessian of the potential energy are computed at an intermediary point $X^{n+1/2}$, for reporting purposes we compute the bond and bond-angle energies at each X^n point of the LN trajectory as well. The added cost is negligible.

All simulations were run with CHARMM[37] version 24b1, modified to include our integration modules, with the CHARMM 22 all-atom parameter set.[38] (Our routines are currently being integrated into CHARMM version 25a3.) The bath temperature was set to T = 300 K and, for fair comparison, the same starting position, velocity vector, and sequence of random numbers were used in the compared LN and BBK trajectories. The equilibration (by BBK) and production lengths are described for each system examined. The correlation function and spectral densities are computed by the correl routines in CHARMM. Computations were performed in serial mode on a 195 MHz SGI Power Challenge computer at New York University.

TABLE II. Averages (mean and variance) for the blocked alanine residue computed over 3 ns trajectories produced by BBK and LN with $\gamma = 20\ \text{ps}^{-1}$ and $\Delta\tau = 0.5$ fs, $\Delta t = \Delta t_m = 3$ fs.

	BBK		LN	
E[a]	21.5	4.74	21.5	4.77
E_k	19.7	3.42	19.7	3.42
E_p	1.78	3.28	1.79	3.30
T[b]	300.1	52.1	300.6	52.2
E_{bond}	6.73	2.07	6.71	2.06
E_{angle}	8.91	2.38	8.86	2.36
E_{tor}	4.71	1.02	4.73	1.03
E_{vdw}	0.12	1.09	0.14	1.12
E_{elec}	−20.4	2.29	−20.4	2.32
ϕ[c]	−107.0	30.3	−106.6	30.0
ψ	117.5	48.5	117.5	48.8
$r_{C_\alpha C}$[d]	1.53	0.034	1.53	0.034
θ_{CNC_α}	123.3	3.28	123.3	3.28
$\theta_{NC_\alpha C}$	110.8	4.12	110.8	4.10
$\theta_{C_\alpha CN}$	117.0	2.76	117.0	2.77
CPU	1.0		0.75	

[a]Energy is given in kcal/mol for the total, kinetic, potential, bond length, bond angle, torsion angle, van der Waals, and electrostatic components.
[b]Temperature is given in degrees.
[c]Angles (torsions and bond angles) are measured in degrees.
[d]Bond lengths are given in Å.

A. Blocked alanine residue model

We first examine data from LN and BBK Langevin simulations at $\gamma = 20\ \text{ps}^{-1}$ for a small blocked-alanine model (N-Acetyl Alanyl N′-Methyl Amide), with 22 atoms.[26] All atoms are represented explicitly. Data were collected over 3 ns (following 160 ps of equilibration), and trajectory snapshots were recorded every 60 fs.

The excellent agreement in averages and variances of the energy components (total, kinetic, and potential) and time-averaged properties of the dihedral angles ϕ and ψ, as well as selected bond lengths and angles, can be seen from Table II. The results obtained with both methods are very close. This is especially good for the stochastic Langevin simulations.[39] The angular variances from both simulations are about 30° and 49° for ϕ and ψ, respectively.

This good agreement of the BBK and LN trajectories can also be seen from Figure 1, which compares the ensemble-generated distributions for a representative bond length ($r_{C_\alpha\text{–N}}$), a representative bond angle ($\theta_{\text{N–}C_\alpha\text{–C}}$), and the two dihedral angles. The matching of the ϕ and ψ distributions, in particular, indicates that the overall motion is essentially the same. This can also be observed from the time evolution of the distance between the dipeptide oxygen and hydrogen atoms that form a hydrogen bond (data not shown).

For reference, the CPU times required to cover 3 ns are 47 and 36 min for BBK and LN, respectively. This timing does not include any force splitting for LN.

B. BPTI and lysozyme

Results of LN without (LN 1) and with force splitting (LN 3, 6, 12, 24, 48, and 96) versus BBK are shown in Tables III

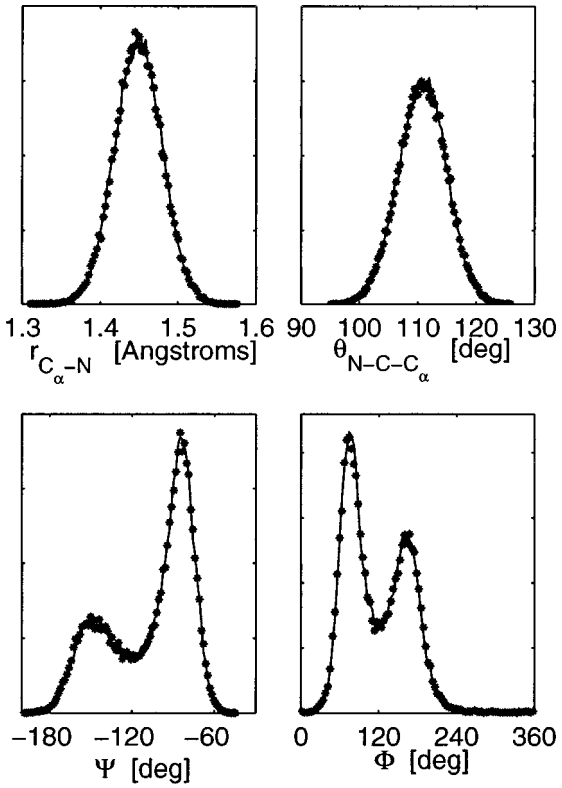

FIG. 1. Distributions for a representative bond length C_α–N (r); a representative bond angle N–C–C_α (θ); and the two torsion angles ϕ,ψ for BBK (***) and LN (—) for a blocked alanine model over 3 ns simulations. Here $\Delta\tau=0.5$ fs (the timestep of BBK and the inner timestep of LN), $\Delta t_m=3.0$ fs, and $\gamma=20\ \mathrm{ps}^{-1}$. Force splitting is not used for this small system, so $\Delta t=\Delta t_m$.

and IV for the proteins BPTI (bovine pancreatic trypsin inhibitor) and human lysozyme, respectively. For the initial BPTI structures, we use for BPTI those described in Barth *et al.*,[40] with four internal water molecules. The starting structure was taken from the Brookhaven Protein Data Bank[41,42] from the file pdb4pti.ent.[43] After the positions of hydrogen atoms were generated,[44] the coordinates were energy-minimized with 500 steps of the Adopted Basis Newton-Raphson method in CHARMM,[37] subject to isotropic harmonic constraints on heavy atoms.[45] The unconstrained system was then heated from 0 to 300 K by 20 K increments during 5 ps (5000 steps of 1 fs), and equilibrated at 300 K for a further 5 ps. The final coordinates from this procedure, together with a random sample of velocities from a Maxwell–Boltzmann distribution at 300 K, were used to start the production runs. For lysozyme, we start from the crystallographic data[46] (1LZ1 in the Brookhaven database[41,42]). After the placement of hydrogen atoms, approximately 10,000 steps of ABNR minimization was performed in CHARMM (gradient tolerance 10^{-8}), followed by 30 ps of equilibration by Langevin dynamics at 300 K.

As for the blocked amino acid above, we report ensemble averages and variances for the total, kinetic, and potential energy, as well as the energy components (bond, angle, torsion, van der Waals, electrostatic) and the kinetic temperature. The last row shows the total CPU time for LN and the speedup over BBK (recall that the CPU time reflects inclusion of all nonbonded interaction—no cutoffs).

The data for both proteins show very good agreement in the mean values reported (first column of each LN entry), with an error not greater than 0.5% for the total energy mean when LN is compared to BBK values. For the other means reported, errors are typically less than 1%, with up to 4% differences exhibited by the van der Waals term (4% for BPTI and 2% for lysozyme).

The LN variances (second column of each LN entry), show larger differences when compared to BBK variance values but, in the worst case (electrostatics), these errors are large percentagewise because the BBK variance is a small percentage of the BBK mean (see percentage tables in Ref. 17). That is, when the BBK reference variance is used in absolute (i.e., numerical) terms (e.g., 16 kcal/mol for the electrostatic energy of BPTI) the LN value can be relatively large (e.g., 21 kcal/mol for LN 96, reflecting a 33% error when the two variances are directly compared); however, the original BBK variance reflects only 0.82% fluctuations from the BBK mean; this increases to only 1.08% in the worst LN case. Similarly, for lysozyme, the electrostatic energy variance in BBK is only 0.54% of the BBK mean and becomes 0.65% of the BBK mean for LN 96 (a 22% deviation from the numerical value of 27 kcal/mol). The LN variances of the total energy and potential energy increase monotonically

TABLE III. Langevin dynamics averages (mean and variance) for BPTI over 60 ps with BBK ($\Delta\tau=0.5$ fs) versus LN ($\Delta\tau=0.5$ fs, $\Delta t_m=3$ fs, and $\Delta t=k_2\Delta t_m$ where k_2 ranges from 1 for LN 1 to 96 for LN 96) at $\gamma=20\ \mathrm{ps}^{-1}$.

	BBK		LN 1		LN 3		LN 6		LN 12		LN 24		LN 48		LN 96	
E[a]	1620.2	32.4	1626.8	32.9	1625.6	32.9	1625.0	33.0	1626.0	33.2	1625.0	33.4	1626.2	33.8	1625.8	34.1
E_k	809.0	22.3	812.5	22.5	811.9	22.4	812.1	22.4	812.4	22.4	812.5	24.2	812.6	22.4	812.6	22.5
E_p	811.1	23.0	814.5	23.3	813.7	23.3	812.9	23.6	813.7	23.9	812.6	24.2	813.7	24.9	813.3	25.4
E_{bond}	322.4	14.6	322.1	14.6	322.1	14.6	322.2	14.6	322.3	14.6	322.6	14.7	322.5	14.7	322.3	14.7
E_{angle}	456.2	16.0	454.7	16.0	454.4	15.9	453.8	15.9	453.8	15.9	457.3	15.9	457.5	15.9	457.3	16.1
E_{tor}	353.6	8.93	355.5	8.94	355.1	8.91	354.7	8.94	354.5	8.94	352.2	9.05	352.5	8.96	354.4	8.93
E_{vdw}	−119.0	12.9	−117.3	13.1	−117.2	13.1	−117.5	13.1	−117.5	13.1	−115.1	13.2	−114.5	13.3	−117.0	14.7
E_{elec}	−1958.4	16.0	−1958.0	17.5	−1958.1	17.5	−1957.7	17.9	−1956.8	18.4	−1961.6	19.8	−1961.5	20.6	−1960.1	21.2
T	300.2	8.28	301.5	8.33	301.3	8.33	301.4	8.32	301.5	8.33	301.5	8.34	301.6	8.32	301.6	8.34
CPU[b]	14.0	(1.0)	3.5	(4.0)	1.9	(7.4)	1.5	(9.3)	1.3	(10.8)	1.2	(11.7)	1.11	(12.6)	1.08	(13.0)

[a]Energy [in kcal/mol] is given for the total, kinetic, potential (with respect to the initial values −1664.96 corresponding to a local minimum near the initial configuration), bond length, bond angle, torsion angle, van der Waals, and electrostatic components.
[b]Time is given in hours, with speedup shown in parentheses.

TABLE IV. Langevin dynamics averages (mean and variance) for lysozyme simulations over 60 ps with BBK ($\Delta\tau=0.5$ fs) and LN ($\Delta\tau=0.5$ fs, $\Delta t_m=3$ fs, Δt variable) at $\gamma=20\ \mathrm{ps}^{-1}$.

	BBK		LN		LN 3		LN 6		LN 12		LN 24		LN 48		LN 96	
E[a]	3605.2	46.4	3620.4	47.3	3617.1	47.9	3621.4	47.7	3620.7	48.2	3622.8	48.7	3623.3	49.0	3621.7	49.6
E_k	1814.9	32.8	1822.6	33.0	1822.1	33.0	1823.2	33.0	1823.8	33.0	1824.1	33.0	1824.0	33.0	1823.8	32.9
E_p	1790.3	33.6	1797.9	34.2	1795.0	34.7	1798.2	34.7	1796.9	35.4	1798.7	36.1	1799.4	36.8	1797.8	38.0
E_{bond}	714.6	21.5	714.0	21.5	714.1	21.5	714.3	21.5	713.9	21.4	714.1	21.4	714.2	21.4	714.3	21.4
E_{angle}	1010.3	23.8	1007.1	23.8	1006.1	23.8	1006.7	23.8	1005.3	23.7	1006.6	23.7	1006.7	23.7	1007.5	23.8
E_{tor}	719.4	12.7	723.7	12.9	723.6	13.0	724.2	12.9	724.6	12.9	723.8	13.0	724.3	13.0	722.8	12.9
E_{vdw}	−426.7	20.2	−423.3	20.6	−421.4	21.5	−422.0	20.5	−419.1	20.5	−422.7	20.6	−422.0	20.7	−420.4	20.3
E_{elec}	−5080.2	27.2	−5078.9	27.8	−5082.5	29.1	−5079.6	28.5	−5083.4	28.8	−5078.7	29.9	−5079.3	31.0	−5081.9	33.2
T	299.9	5.4	301.2	5.5	301.1	5.4	301.3	5.4	301.4	5.5	301.4	5.4	301.4	5.5	301.4	5.4
CPU[b]	72.4	(1.0)	14.9	(4.9)	7.8	(9.3)	5.7	(12.7)	4.7	(15.4)	4.2	(17.2)	3.9	(18.6)	3.8	(19.1)

[a]See Table III legend. Here the potential energy is given with respect to the initial value −4637.85 kcal/mol, corresponding to a local minimum near the initial configuration.
[b]CPU time is shown in hours; LN speedup is shown in parentheses.

with Δt, exhibiting up to 14% error (van der Waals energy for BPTI in LN 96) in direct comparison to the BBK variance value, but typically much less. The van der Waals variance for BPTI obtained by BBK is 11% of the mean value; the LN 96 error of 14% corresponds to a variance of 12% of the BBK mean. Thus, increasing the interval between slow-force updates does not affect the trajectory means significantly but yields somewhat larger variances in these quantities (e.g., electrostatic energy). Still, upon careful examination, the behavior is satisfactory.

To examine configurational behavior in time for these proteins, we also show in Figures 2–4 the root mean square (rms) fluctuations for various quantities: (a) total rms from the starting structure, (b) rms of the backbone C_α atoms (58 for BPTI, 130 for lysozyme), and (c,d) rms fluctuations of the ψ angles along the protein backbone. For both proteins, we also performed a BBK simulation with 12 Å cutoffs for comparison to the LN trajectories as k_2, and hence the outermost timestep, increases.

First, with respect to the total rms fluctuations of the proteins from the initial (equilibrated) structure, we note that the BBK trajectory with cutoffs diverges from the other curves after around 35 ps. Most of the LN variants follow the pattern of BBK without cutoffs well, but the LN 96 version shows poorer agreement than the others. This difference is more notable for BPTI, but still the trend is not a diverging pattern as is seen for the cutoff trajectory.

The rms plots of each C_α atom in the proteins (Figure 3) again show good agreement between the LN variants and BBK, as well as a substantial difference from the cutoff trajectory. The same good agreement can also be seen in the rms fluctuations of each ψ (Figure 4) with LN 96 showing poorer agreement. The ϕ data (not shown) exhibit the same trends. Recall that in LN 96 the slow forces are only updated every 288 fs! These results suggest that LN 24 or LN 48 present more reasonable limits, especially since the asymptotic speedup is nearly reached (see Table I).

Dynamic analyses of the trajectories are performed via the cosine Fourier transforms of the velocity autocorrelation function[34] $C_{VV}(t)$, where

$$C_{VV}=\langle V(t)^T V(0)\rangle/\langle V(0)^T V(0)\rangle. \tag{10}$$

Here $V(t)$ represents the discrete approximation to V at time t, and the brackets denote ensemble averages. The transformation of this real-valued function into the frequency domain produces the spectral density function

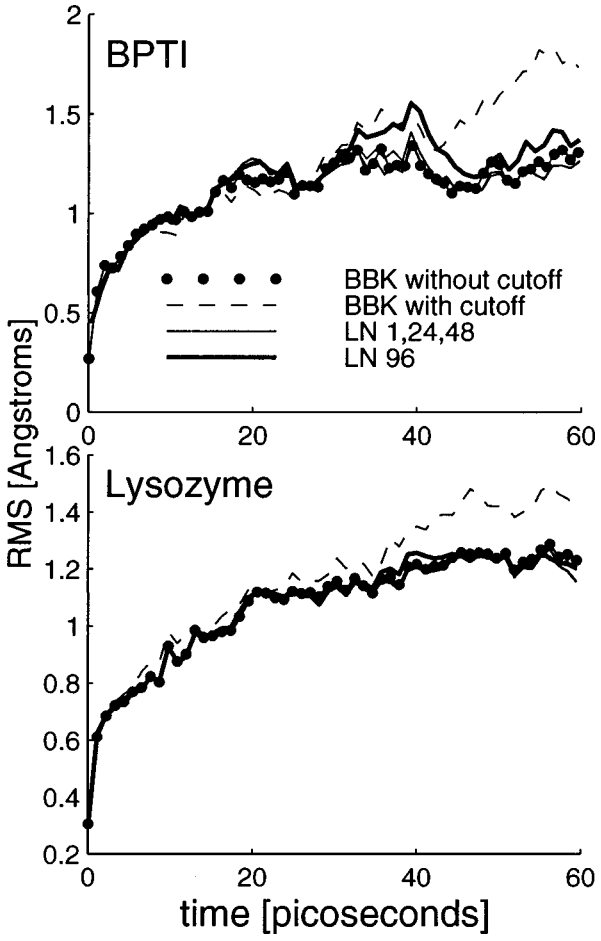

FIG. 2. The root-mean-square (rms) deviations from the initial (equilibrated) structure of BPTI (top) and lysozyme (bottom) taken from 60 ps Langevin simulations with $\gamma=20\ \mathrm{ps}^{-1}$: BBK without cutoffs (●●●), BBK with a 12 Å cutoff (––), LN 1,24,48 (—) and LN 96 (——). As before, $\Delta\tau=0.5$ fs, $\Delta t_m=3$ fs, and $\Delta t=k_2\Delta t_m$ where k_2 is the number following LN.

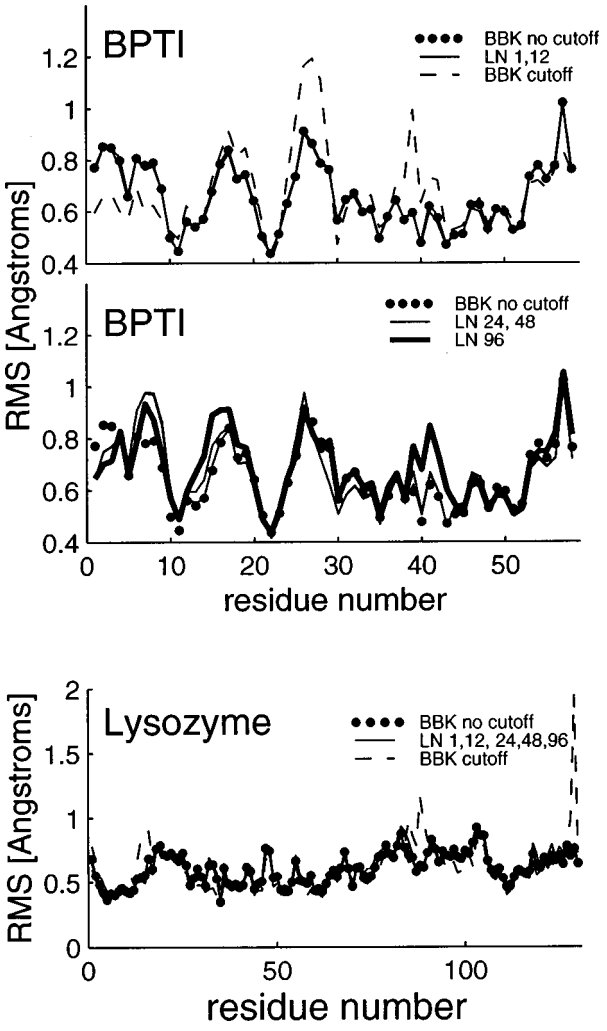

FIG. 3. C_α RMS fluctuations for BPTI and lysozyme over 60 ps Langevin simulations. For BPTI, the top panel compares LN 1 and LN 12 results to BBK, without (●●●) and with (––) cutoffs (at 12 Å), and the center panel shows LN 24, 48 and 96 versus BBK. The lower box compares BBK to LN for lysozyme.

$$\hat{C}_{VV}(\omega) = 2\int_0^\infty dt\; C_{VV}(t)\cos\,\omega t, \tag{11}$$

where $\omega = 2\pi c\nu$ (c = speed of light) and ν is the wave number.

The spectral density functions for BPTI and lysozyme (Figure 5) are shown for two values of γ (5 and 20 ps^{-1}) as obtained from trajectories of BBK, LN 1, LN 96, and Verlet ($\gamma = 0$). The excellent agreement between all the LN and BBK spectra is striking. We note spectral peaks corresponding to O–H stretches around 3300 cm^{-1} and C–H stretches around 3000 cm^{-1}. The regions around 1400 cm^{-1} and between 600 and 1200 cm^{-1} are associated with various bending nodes and also heavy-atom bond vibrations such as C—C and C═O.

Interesting from these spectra is the dependence on γ, that is the dependence of dynamics on the governing model rather than the scheme. The Verlet ($\gamma = 0$) spectra show a sharper peak for the C–H stretch around 3000 cm^{-1}, as well as sharper patterns for the peak area 1400 cm^{-1} and the region below it. The Langevin trajectory with the smaller γ (5 ps^{-1}) agrees more closely with Verlet than the larger γ used (20 ps^{-1}). We also notice a rough agreement between the $\gamma = 0$ and $\gamma = 5$ ps^{-1} spectra. The Langevin modes appear more smoothed out in comparison to the resolved Newtonian frequency pattern, with splitting of some of the more notable Verlet peaks.

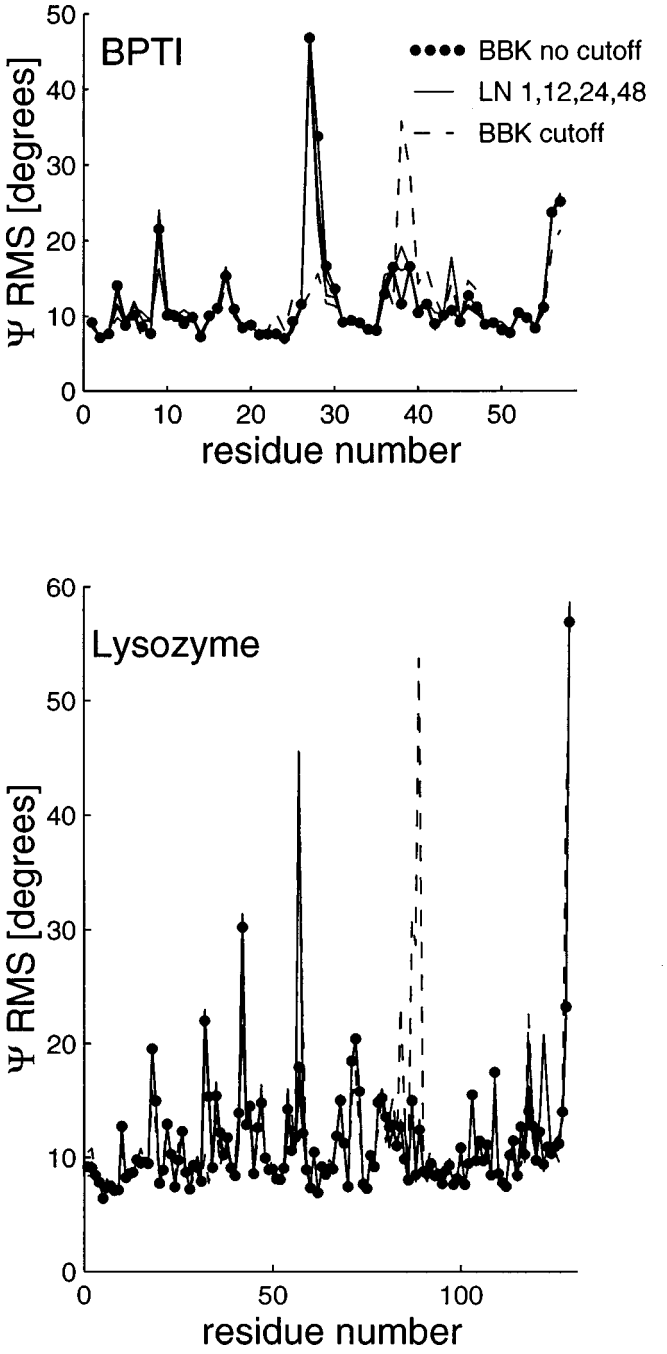

FIG. 4. Root-mean-square fluctuations of the backbone torsion angle ψ for BPTI and lysozyme over 60 ps Langevin simulations at $\gamma = 20$ ps^{-1} as obtained by BBK, LN, and BBK with a 12 Å cutoff.

Besides velocity autocorrelation functions corresponding to all atoms, we also calculated these spectra for BPTI from hydrogen atoms only (data not shown); the difference between Langevin and Newtonian modes is smaller for both γ.

The trends in Langevin modes have been investigated by a number of authors as a function of the damping parameter

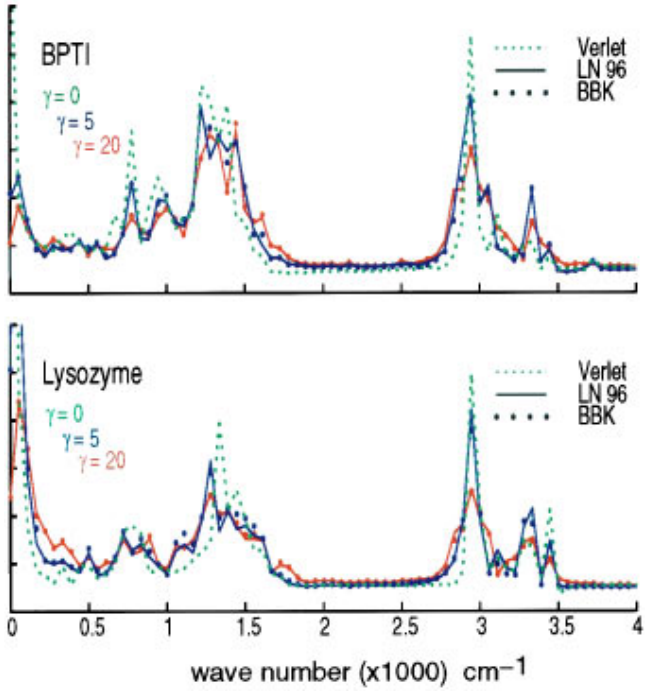

FIG. 5. BPTI and lysozyme spectral densities derived from the cosine Fourier transforms of the velocity autocorrelation function for all atoms from BBK and LN Langevin trajectories of length 6 ps for two choices of collision frequency: $\gamma=5$ and 20 ps^{-1}. Here $\Delta\tau=0.5$ fs, $\Delta t_m=3.0$ fs, and $\Delta t=\Delta t_m$ for LN 1 and $\Delta t=96\Delta t_m=288$ fs for LN 96. Spectral densities for the Verlet method ($\gamma=0$) are also shown for reference ($\cdots$).

(see Case[47] for a review and references cited therein) with the goal of understanding the effect of solvent viscosity on biomolecular motion. Essentially, it is found that the low-frequency vibrational modes become overdamped in the Langevin treatment and that the decay of various correlation functions is smoother for Langevin than Newtonian motion components.

With regard to speedup, the LN 3 scheme gives a speedup factor of 8 for BPTI (2712 variables) and 9 for lysozyme (6090 variables). Performance improves as system size increases because of the larger fraction of long-range interactions. As the outermost timestep increases, LN speedup approaches 13 for BPTI and 19 for lysozyme (Tables III and IV). Performance improves here since savings in the time-consuming slow force are realized, and the asymptotic behavior stems from the increased importance of CPU time for the fast and medium force components (see also Table I). We discuss CPU issues in more detail below.

C. Water droplet

Next we examine in Table V and Figure 6 results for a water droplet of radius 31 Å (with all nonbonded interactions considered). This system size (4131 molecules, 37,179 variables) is large enough to envelop lysozyme. We use CHARMM's quartic spherical boundary potential to maintain the appropriate density of the system and eliminate boundary effects on the primary region of interest.[48] A flexible, modified version of the TIP3P[49] water potential is used, in which a van der Waals term for the hydrogens prevents unusually close contacts with other charged atoms. Here we compare Langevin trajectories with collision parameter $\gamma=50$ ps^{-1}.

We found for this system that smaller values of Δt_m (1 or 2 fs) work better than 3 fs in terms of agreement to BBK. This can be explained by the high degree of anharmonicity governing the water droplet system and the fast librational modes in water. Recall that our results for water used no switching function and group-based cutoffs. Indeed, we noticed increased sensitivity to the stability of the LN simulations to the choice of short-range cutoffs. The good agreement with the reference BBK trajectories can also be seen from Figure 6, which shows for the different LN protocols the errors in various energy components relative to BBK. Results are given for long-range forces updated every $k_2\Delta t_m$ for integers k_2 ranging from 1 to 6. The bond energy component reveals the largest errors for this system, here at most 2% with LN 3 and only 1% with LN 6. The best agreement with the small-timestep method in terms of energies and variances is obtained with LN 6 ($\Delta t_m=1$ fs, $\Delta t=6$ fs) formulation. The speedup of LN 6 with $\Delta t_m=1$ fs (10.2) is only slightly less than LN 3 with $\Delta t_m=2$ fs (10.7), making the case for reducing Δt_m strong. Thus, the outer timestep of LN dominates the CPU performance for larger systems. This is because the

TABLE V. Langevin dynamics averages (mean and variance) for the water droplet simulations over 0.4 ps with all nonbonded interactions included: BBK ($\Delta t=0.5$ fs) versus LN ($\Delta\tau=0.5$ fs, $\Delta t_m=1$ and 2 fs, and $\Delta t=\Delta t_m$, $3\Delta t_m$, and $6\Delta t_m$) with the sparse $\tilde{\mathbf{H}}$ constructed for bond-length and bond-angle terms, $\gamma=50$ ps^{-1}.

	BBK $\Delta t=0.5$ fs		LN $\Delta t=1$ fs		LN $\Delta t=2$ fs		LN 3 $\Delta t=2$ fs		LN 6 $\Delta t=1$ fs	
E[a]	−59094.6	771.9	−59113.1	797.8	−58599.6	732.9	−58602.3	734.8	−59111.2	795.3
E_k	11133.8	232.7	11128.4	251.0	11381.4	212.2	11384.3	211.7	11131.3	250.4
E_p	−70228.4	664.3	−70241.6.0	672.2	−69981.0	644.4	−69986.5	646.5	−70242.5.0	669.7
E_{bond}	3838.0	86.8	3788.7	84.5	3900.6	75.2	3902.0	75.5	3786.3	84.4
E_{angle}	4105.1	74.4	4098.4	74.3	4099.3	75.0	4099.2	74.3	4098.6	73.8
E_{vdw}	6900.1	222.0	6894.2	220.4	6877.6	215.0	6878.6	214.3	6893.8	218.2
E_{elec}	−85065.0	849.5	−85016.2	855.3	−84851.8	832.8	−84859.7	834.6	−85014.6	851.3
T[b]	301.4	6.30	301.3	6.80	308.1	5.74	308.2	5.73	301.3	6.77
CPU[c]	17.6 (1.0)		8.81 (2.0)		4.45 (4.0)		1.64 (10.7)		1.72 (10.2)	

[a]Energy is given in kcal/mol for: total, kinetic, potential, bond-length, bond-angle, torsion, van der Waals, and electrostatic terms.
[b]Temperature is given in degrees.
[c]CPU time is shown in hours, with speedup of LN in parentheses.

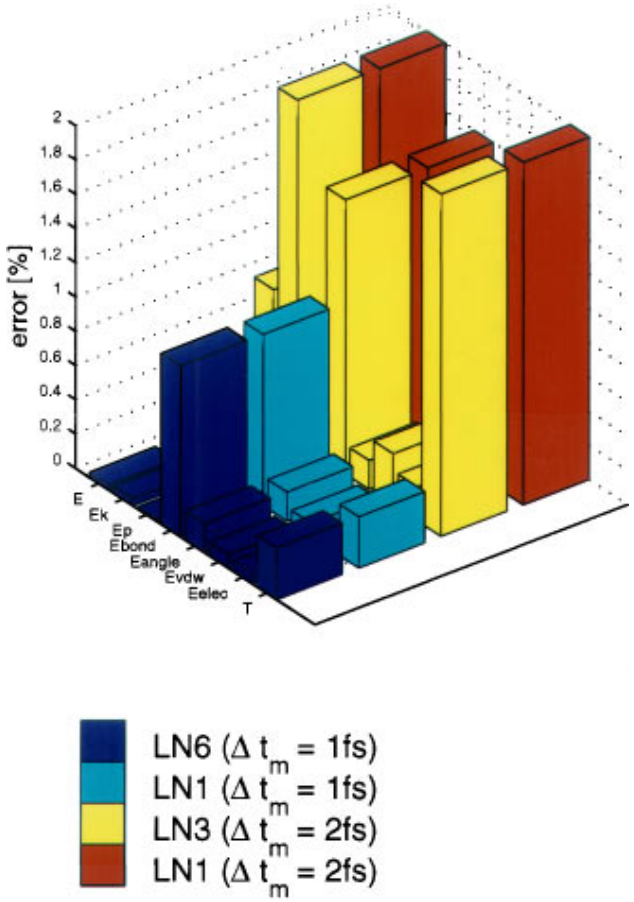

FIG. 6. Error in LN energies for the water simulations relative to BBK. The LN error is shown for the different energy components (see Table V) for different LN protocols propagated over 0.4 ps at $\gamma=50\ \mathrm{ps}^{-1}$ without any cutoffs enforced.

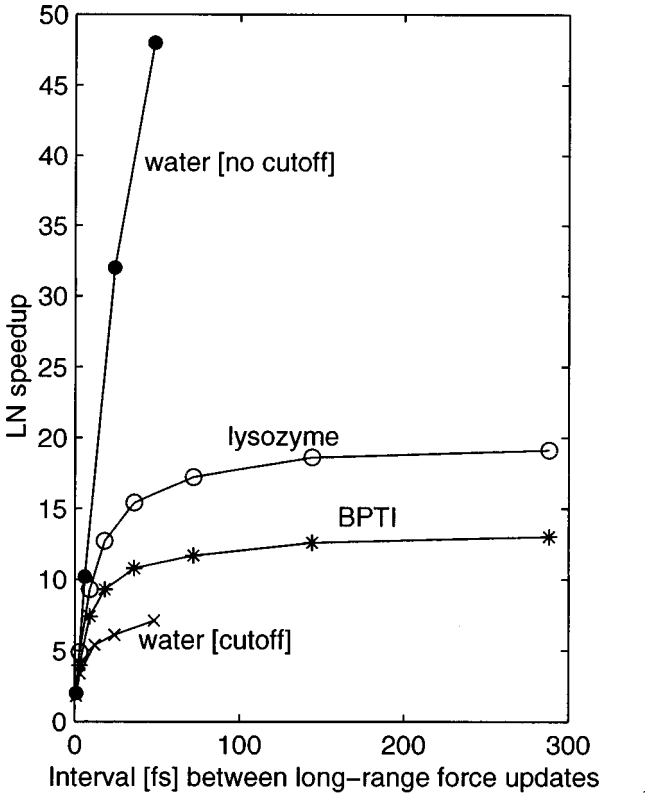

FIG. 7. LN speedup compared to BBK simulations with $\Delta\tau=0.5$ fs. Shown are the ratios of CPU times from 60 ps simulations for BBK and LN with $\Delta t_m=3.0$ fs, for BPTI and lysozyme and $\Delta t_m=1$ fs for water. In all cases, $\Delta t=k_2\Delta t_m$ where k_2 ranges from 1 to 96. The Langevin γ values (in ps^{-1}) are 20 for the proteins and the water system with cutoffs, and 50 for the water system without cutoffs. The data for the water system is based on our estimate of 73 days for a 60 ps simulation without cutoffs.

short-range forces require a small percentage of the work for the long-range components.

While long BBK simulations of the water system without cutoffs are not generally feasible (we estimate that a 60 ps trajectory would require 73 days), we have generated 60 ps trajectories using LN. For example, LN with $\Delta t=48$ fs required 36.7 h, a speedup of 48. Some timing results are shown in Figure 7. From this figure we see that smaller outer timesteps Δt are used for water than for the proteins considered above. The outer timestep is limited by processes in the physical system (e.g., fast librations for water); this is reflected in the need to update the pair list associated with the force-splitting cutoff distance more frequently. We have had preliminary success in extending Δt by updating the force-splitting pair list more frequently than the long-range nonbonded force evaluations. With this practice, however, there is the danger of including interactions in *neither* the long nor short-range treatments, or in both *simultaneously*. That is, during the interval over which the long-range forces are kept constant, the separation of certain atom pairs may increase (or decrease) sufficiently to change the status from short to long-range (or vice versa). In the first case, the interaction is no longer considered in the short-range treatment and is not included in the long-range treatment until the next long-range force update. In the second situation, both the older long-range and the newer short-range forces include contributions from the given atom pair. We continue to study this issue.

To examine the performance of LN with a more practical water model, we also simulated the dynamics of the water system with $\Delta t_m=1$ fs and 12 Å nonbonded cutoffs for both BBK and LN at $\gamma=20\ \mathrm{ps}^{-1}$. The smaller γ value is preferred if a better agreement of the spectral densities to Verlet is of concern. For large values of k_2 (e.g., 24 or 48), the nonbonded pair list associated with the 12 Å cutoff is updated every k_2 LN steps, along with the recalculation of the nonbonded forces. For smaller values (such as $k_2=1$ or 3) the nonbonded cutoff pair list can be updated less frequently. In the latter situation, we update the nonbonded pair list every 24 fs (i.e., every 8 outer steps of LN 3). We found again that this updating limits the size of k_2. However, the computational efficiency of LN is not compromised here, due to asymptotic speedup considerations considered in the next section. Results are shown in Table VI: The agreement of BBK to LN is very good; significantly the speedup of LN 48 is 7.1.

To compare the structural properties of water trajectories produced by LN, we have run 5 ps simulations for a box containing 216 TIP3P water molecules. We found the large system with the quartic boundary potential inappropriate for this purpose due to an accumulation of excessive pressure in

TABLE VI. Langevin dynamics averages (mean and variance) for the water system over 5 ps with nonbonded interactions truncated at 12 Å: BBK ($\Delta\tau = 0.5$ fs) versus LN ($\Delta\tau = 0.5$ fs, $\Delta t_m = 1$ fs, and $\Delta t = k_2 \Delta t_m$ where $k_2 = 1, 3, 12, 24, 48$) at $\gamma = 20\ \text{ps}^{-1}$.

	BBK		LN 1		LN 3		LN 12		LN 24		LN 48	
E[a]	−67245.8	3122.6	−67187.9	3115.5	−67202.8	3135.5	−67125.0	3160.5	−66976.4	3110.7	−66952.6	3109.1
E_k	11141.3	89.6	11239.7	92.5	11249.2	92.2	11276.8	95.3	11294.5	91.9	11284.8	91.7
E_p	−78387.1	3099.4	−78427.6	3085.3	−78452.0	3114.5	−78401.8	3132.6	−78270.9	3091.9	−78237.5	3082.3
E_{bond}	3905.1	80.0	3876.0	75.8	3880.4	78.2	2872.3	76.6	3878.1	77.2	3868.7	73.9
E_{angle}	4809.3	276.1	4801.3	268.8	4807.0	275.3	4820.3	289.8	4813.7	263.3	4808.6	274.8
E_{vdw}	8012.3	273.7	8009.5	277.2	8006.6	275.2	7995.1	273.4	8016.8	278.9	8007.1	282.4
E_{elec}	−95113.8	3630.8	−95114.4	3606.4	−95146.0	3639.4	−95089.4	3679.0	−94979.5	3618.3	−94921.9	3606.1
T	301.6	2.43	304.3	2.51	304.5	2.47	305.3	2.58	305.7	2.49	305.5	2.48
CPU[b]	9.2 (1.0)		5.0 (1.8)		2.7 (3.4)		1.7 (5.4)		1.5 (6.1)		1.3 (7.1)	

[a]Energy [in kcal/mol] is given for the total, kinetic, potential, bond length, bond angle, van der Waals, and electrostatic components.
[b]Time is given in hours, with speedup shown in parentheses.

the system's interior. The equilibrated structure was obtained from the CHARMM test file test/data/tip216.crd[37] and used for LN and BBK simulations. Radial distribution functions from these simulations are compared in Figure 8. These distributions agree well with each other and with those reported in other simulations that developed the TIP3P model[49] for good agreement with experiment; some flattening beyond the first peak in the computed O–O radial distribution functions was originally attributed to this water model.[49]

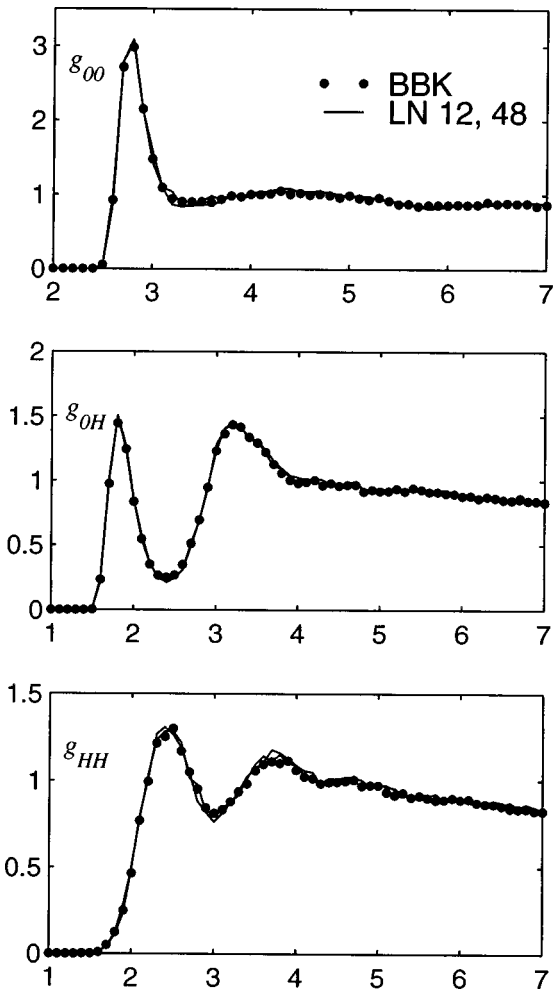

FIG. 8. Water radial distribution functions $[g(r)]$ for O–O, O–H, and H–H interactions (g_{OO}, g_{OH}, and g_{HH}) calculated from 5 ps simulations with BBK (●●), and LN 12, 48 with $\Delta t_m = 1.0$ fs, at $\gamma = 20\ \text{ps}^{-1}$.

D. CPU time

The two parts of Table I show the CPU timings of BBK versus LN without (part A) and with (part B) force splitting in LN. In part A, for each system we show the sparsity of $\tilde{\mathbf{H}}$, as measured by the ratio of nonzero elements in the upper triangle (diagonals included) to the total number of elements in the upper triangle [$N(N+1)/2$ entries for an $N \times N$ matrix]. The value of $T_{\nabla E}$ gives the cost (in seconds) of evaluating the full gradient, and $T_{\tilde{\mathbf{H}}}$ specifies the evaluation cost of the sparse Hessian. $T_{\tilde{\mathbf{H}}d}$ indicates the cost of one multiplication of $\tilde{\mathbf{H}}$ by a vector. The cost of the linearization phase of LN (covering the interval Δt_m) amounts to k sparse-Hessian/vector products. Following the CPU values, the CPU percentage of that task (e.g., gradient evaluation) with respect to the entire computational time is also given. In the last three columns, the CPU costs per Δt interval ($= \Delta t_m$ in part A) are also given for BBK and LN, followed by the ratio of the two (speedup).

As system size increases (lysozyme has 2.5 times more atoms than BPTI), the number of entries in the approximate Hessian $\tilde{\mathbf{H}}$ grows linearly, and so does the computational cost of integrating the linearized equations of motion ($\tilde{\mathbf{H}}$/vector products). In principle, the cost of evaluating the Hessian should also increase linearly with the number of entries, but the current CHARMM implementation of the 1–4 electrostatic terms does not exploit the increased sparsity.

Note that the sparsity of our $\tilde{\mathbf{H}}$ is less than 5% for systems with more than 6000 atoms. One calculation of $\tilde{\mathbf{H}}$ requires about one fourth the CPU of the gradient calculation. This fact, combined with the cheapness of evaluating the linearized forces—about 24% of the total time for BPTI, 19% for lysozyme, and only 1% for the large water system without cutoffs (9% for water with a 12 Å cutoff)—explains why LN 1 offers overall speedups.

The small CPU percentages for the sparse Hessian evaluation and the linearization in LN 1 also imply that the majority of the work comes from gradient computations. Indeed, the percentage of CPU spent on gradient computation is about

80% for lysozyme and 99% for water (91% with cutoffs) in LN 1. This suggests the benefit of force splitting in LN, in the spirit of MTS methods, for additional speedup. For the three-stage MTS method described in system (3), the speedup factors depend on the ratio $r=\Delta t/\Delta\tau$ and the cost of the full gradient (∇E) relative to the gradient of the fast (∇E_{fast}) and the medium (∇E_{mid}) forces. Similarly in LN with force splitting, where the fast forces are treated by linearization, speedup depends on the cost of ∇E relative to the cost of ∇E_{mid} and the cost of evaluating the approximate Hessian $\tilde{\mathbf{H}}$ and performing the sparse Hessian/vector products.

The costs of the fast and medium gradient parts, as well as the linearization components, are given in part B of Table I. Note that for the proteins, more than half of the work for the fast components involves $\tilde{\mathbf{H}}$ evaluation. Using these values, we can estimate the CPU cost of LN relative to BBK and the impulse MTS scheme, as follows. The CPU cost of one Δt step of LN is given by

$$T^{\text{LN}}=T_{\nabla E}+k_2(T_{\nabla E_{\text{mid}}}+T_{\tilde{\mathbf{H}}}+k_1T_{\tilde{\mathbf{H}}d}),$$

where $T_{\tilde{\mathbf{H}}}$ is the cost of evaluating the sparse Hessian, and $T_{\tilde{\mathbf{H}}d}$ is the cost of a sparse Hessian vector product. The cost of BBK over the same (Δt) interval is $T^{\text{BBK}}=rT_{\nabla E}$. The LN speedup over BBK is given by the ratio

$$\begin{aligned}\text{LN speedup}&=k_2k_1T_{\nabla E}/[T_{\nabla E}+k_2(T_{\nabla E_{\text{mid}}}+T_{\tilde{\mathbf{H}}}+k_1T_{\tilde{\mathbf{H}}d})]\\&=k_1T_{\nabla E}/[T_{\nabla E}/k_2+(T_{\nabla E_{\text{mid}}}+T_{\tilde{\mathbf{H}}}+k_1T_{\tilde{\mathbf{H}}d})].\end{aligned}\tag{12}$$

From equation (12) we see that as the long-range update parameter k_2 grows, the LN speedup approaches the asymptotic limit

Asymptotic LN speedup

$$=k_1T_{\nabla E}/(T_{\nabla E_{\text{mid}}}+T_{\tilde{\mathbf{H}}}+k_1T_{\tilde{\mathbf{H}}d}).\tag{13}$$

Using the data from Table I, where $k_1=6$, we can calculate the asymptotic speedup for BPTI to be $(6\times0.38)/(0.05+0.11+6\times0.002)\approx13.3$. The analogous calculation for lysozyme gives 19.8 as the approximate asymptotic LN speedup. It can be seen from Table I (part B) and Figure 7 that LN rapidly approaches the asymptotic speedup as the long-range force update frequency, k_2, increases.

Similarly, the CPU cost of the impulse MTS method [system (3)] is given by

$$T^{\text{MTS}}=T_{\nabla E}+k_2(T_{\nabla E_{\text{mid}}}+k_1T_{\nabla E_{\text{fast}}}),$$

and the speedup over Verlet (or BBK) is

Impulse MTS speedup

$$\begin{aligned}&=k_2k_1T_{\nabla E}/[T_{\nabla E}+k_2(T_{\nabla E_{\text{mid}}}+k_1T_{\nabla E_{\text{fast}}})]\\&=k_1T_{\nabla E}/[T_{\nabla E}/k_2+(T_{\nabla E_{\text{mid}}}+k_1T_{\nabla E_{\text{fast}}})].\end{aligned}\tag{14}$$

In contrast to LN, the stability of the impulse method discussed above and in the companion paper[2] restricts the parameters k_1 and k_2 to values such that $k_1k_2\Delta\tau<5$ fs. Hence the notion of asymptotic speedup does not apply in this case. Using the BPTI data from Table I, and the typical[11] values $k_1=4$ and $k_2=2$ and $\Delta\tau=0.5$ fs, the MTS speedup over BBK is close to 6, possibly higher than realized in practice due to omission of some overhead calculations.

Assuming that the timestep treatment $\Delta t_m=k_1\Delta\tau$ is the same in both methods, we can approximate the LN speedup over MTS by taking the ratio of the costs per Δt_m of each method

$$\text{LN speedup over MTS}\approx\left(\frac{k_2^{\text{LN}}}{k_2^{\text{MTS}}}\right)\left(\frac{T_{\nabla E}+k_2^{\text{MTS}}(T_{\nabla E_{\text{mid}}}+k_1T_{\nabla E_{\text{fast}}})}{T_{\nabla E}+k_2^{\text{LN}}(T_{\nabla E_{\text{mid}}}+T_{\tilde{\mathbf{H}}}+k_1T_{\tilde{\mathbf{H}}d})}\right),\tag{15}$$

where k_2^{MTS} and k_2^{LN} correspond to the long-range force update frequencies in the respective schemes. In fact, we see from eq. (15) that the speedup is given by the product of the ratio $(k_2^{\text{LN}}/k_2^{\text{MTS}})$ and the ratio of the costs for treating the medium and fast forces in the two methods. Thus, the speedup advantage of LN over MTS comes largely from the method's success in using a larger k_2, and hence larger Δt value for the slow-force updating frequency. This computational advantage is amplified by the second term in equation (15) when the linearization treatment offers cost savings over direct evaluation of the fast and medium force components. This expression also suggests that LN's linearization approach can be replaced by direct calculation of the fast forces when linearization is not advantageous in terms of computational speed.

E. Comparison of LN to MTS at increasing Δt

The success of LN in overcoming the half-period resonance due to its extrapolative technique can be seen from Figure 9 shown for BPTI and lysozyme. The results for water (not shown) reveal the same trend. The impulse-MTS version for Langevin dynamics has been adapted and implemented in CHARMM for this purpose. The plots show the average bond energy, and associated variance, as a function of the updating frequency of the long-range (slow) forces. We use $\Delta t_m=1$ fs in all cases, so the abscissa also corresponds to the time in units of femtoseconds. The total simulation times are short (see captions) to reduce computation time. We note that the presence of stochasticity renders the impulse method immune to the half-period barrier, but not at the period (10 fs) of the fastest motion, where large energetic fluctuations

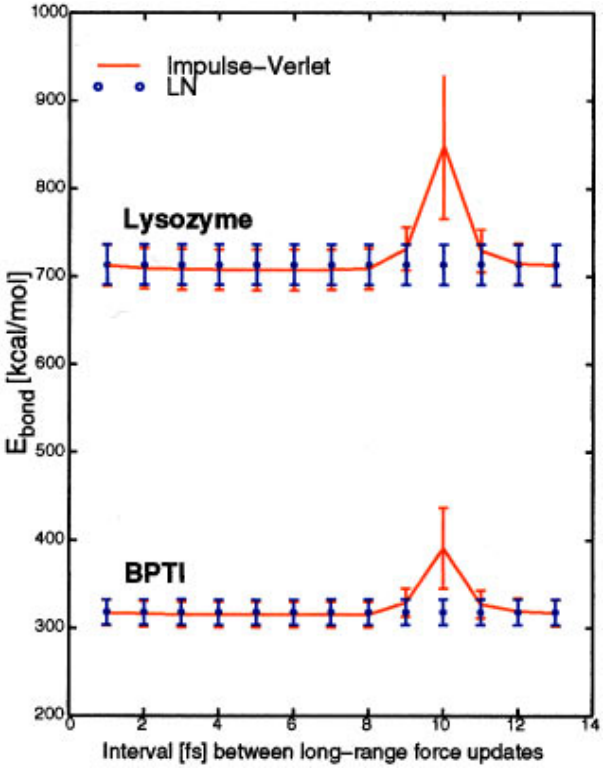

FIG. 9. Impulse-MTS vs LN for Langevin simulations of BPTI and lysozyme over 2 ps ($\gamma=20$ ps^{-1}). Average bond energy and variance are shown for the two methods: Verlet-I/r-Respa (or impulse-MTS) generalized for Langevin dynamics and LN with extrapolation of the long-range nonbonded forces. The abscissa gives the interval (in fs) between updates of the long-range forces, defined as interactions outside of a 6 Å radius. In all cases, bond and angle forces are updated every $\Delta\tau=0.5$ fs, and local (<6 Å) forces are updated every $\Delta t_m=1.0$ fs.

result. Our linear analysis[2] shows expected artifacts at multiples of half the period for a two-class force-splitting scheme.

No such average-energy increase due to resonance is observed for LN. We found that the energy increase effect for impulse-MTS can be significantly diminished by increasing γ, but our reasoning for minimal relative weight of the random forces argues against this approach.

V. DISCUSSION AND CONCLUSION

Traditional time discretization schemes for molecular dynamics—Verlet and its variants—have been popular for 30 years. This is due to their simplicity and reliability. The suitability of these methods for propagating molecular motion is largely attributable to their preservation of abstract properties (symplecticness, phase-space volume, time reversibility) present in the exact solution of the Newtonian equations of motion. The disadvantage of these methods is that the length of timestep, and hence the efficiency of the overall simulation, is limited by numerical stability of the method.

Furthermore, the notion of symplecticness becomes more blurred as the timestep is increased, since all theoretical concepts (e.g., the closeness of the symplectic trajectory to that of a nearby Hamiltonian[19]) are true only in the limit of very small timesteps. The timesteps used in molecular dynamics are, in fact, relatively large. In our experience, many differential-equation packages that use adaptive timesteps do not permit the typical 1 fs timesteps used in molecular dynamics (only one tenth the fastest period), due to violation of error tolerance parameters. Indeed, Deuflhard *et al.* recently showed that only timesteps less than 0.01 fs yield convergence of the end-to-end distance of a butane molecule![50] (see also Ref. 17).

The Verlet method has been generalized in a number of ways for increased efficiency in biomolecular simulations. Success has been achieved by adding constraints via SHAKE (speedup of 2–4),[51] and by using MTS methods such as r-Respa and Verlet-I (speedup of about 5).[10,11] In each such case, the time discretization shares the structure-preserving properties of Verlet, as well as the inherent numerical stability issues which limit the timestep length.

With the introduction of stochasticity in Langevin dynamics, a number of numerical integration schemes become available which are not applicable for long-time simulations in the time-reversible regime of Newtonian dynamics. The exchange of Hamiltonian dynamics for stochastic dynamics can guarantee better numerical behavior, but the resulting detailed dynamics are not the same.

Here we have described the LN method, which differs from those mentioned above in that forces are applied frequently, but updated more or less rarely, based on the fundamental timescales involved; between updates, the forces are held constant. Such extrapolation methods appear quite suitable for Langevin dynamics simulations in combination with small γ, to stabilize the simulations but weigh the inertial terms maximally. Moreover, this extrapolative approach of LN does not suffer dramatically from the resonance limitations of the impulse treatments.[2] This results in speedup factors over current MTS schemes, and thus longer times can be followed for the same amount of computing time.

The LN scheme uses the triplet $\{\Delta\tau,\Delta t_m,\Delta t\}$, where $\Delta\tau$ corresponds to the subintegration time of the linearized equations of motion, Δt_m is the frequency of updating the harmonic approximation, and Δt is the interval between slow-force updates. In each inner timestep, sparse-Hessian/vector multiplications are required; at integral multiples of Δt_m, the local Hessian is updated along with the fast and ''medium'' forces (the latter defined here within a 6 Å radius); and, every $\Delta t=k_2\Delta t_m$ interval, the long-range forces are recalculated. We have used $\Delta\tau=0.5$ fs, $\Delta t_m=1$ fs (for water) and 3 fs for the proteins, and $\Delta t=k_2\Delta t_m$ where k_2 extends from 1 to 96. The nonbonded pair list is updated every Δt, but this might improve (see note in text). Special care was needed in the implementation of cutoffs and switching function parameters. We have shown through detailed examinations of trajectories for two proteins in vacuum and a water droplet system that LN gives very good agreement to explicit Langevin trajectories performed at the inner timestep $\Delta\tau$. This agreement was assessed in terms of ensemble-generated averages and means for the energy components, geometric quantities, and rms fluctuations from the initial structure, and of each backbone C_α and ψ angle from its trajectory mean.

The correctness of the resulting LN dynamic properties was demonstrated by the agreement of the spectral density

function of LN to that obtained for a small-timestep Langevin trajectory. Speedup factors depend on the problem size (see tables), but factors exceeding 10 arise when k_2 is 6 or greater in our examples.

A comparison to Newtonian dynamics demonstrated that as γ is decreased, the smoothed Langevin spectra resemble more closely the sharper Newtonian spectra. In particular, the agreement between the LN spectra at $\gamma=5\ \mathrm{ps}^{-1}$ with that of Verlet ($\gamma=0$) is not bad. Thus, a value smaller than $20\ \mathrm{ps}^{-1}$ may be preferred if such considerations are important. However, decreasing γ increases the probability that a trajectory will become unstable at some point because the stochastic terms may be too weak to restore equilibrium. We have encountered such a case for $\gamma=5\ \mathrm{ps}^{-1}$ in LN 48 with $\Delta t_m=3$ fs (LN 96 was stable for this protocol), but this could occur more generally. In such occurrences, γ should be increased or Δt_m decreased (doing either stabilized the computation above). See paper II[2] for a discussion of the effect of γ on LN's stability.

The general issue of Langevin versus Newtonian dynamics depends strongly on the applications in mind. For thermodynamic and structural questions, the efficiency of the LN method may be preferred, while for detailed dynamic questions, symplectic integration of the Newtonian equations may be important. Given the highly approximate nature of the governing molecular mechanics force fields and the enormous spatial and temporal scales involved in biomolecular motion, the LN method may be well suited for thermodynamic and sampling questions. Small-timestep dynamics simulations can always be performed in tandem once an interesting region of conformation space is identified.

Based on our analyses, we recommend $k_2=24$ or 48 as an upper bound. The asymptotic speedup is nearly reached for these values (Figure 7). Similarly, the $\Delta t_m=1$ fs used for water is likely to be the recommended value for solvated biomolecules, although slightly larger values can be considered. Water's fast librational modes appear to force this smaller value for LN as well as for MTS schemes. If this smaller value is used, speedup factors will be smaller than those we showed for the proteins in vacuum, but the system sizes will be larger.

The LN method can also be incorporated in the context of Monte Carlo schemes for enhanced sampling. For added efficiency and decreased emphasis on dynamics, the LN protocol may be pushed further, for example by using a smaller radius than 6 Å for defining the medium forces, increasing Δt_m, and also incorporating nonbonded cutoffs at a large value (e.g., 15 Å).

The implementation of LN in any program that already incorporates force splitting is straightforward. Besides decomposing the forces, such programs must carefully turn off the slow forces [i.e., by means of a gradual switching function such as that given in equation (9)] and update the local nonbonded pair list every time the slow forces are recalculated.

In addition to this splitting, LN requires a routine to assemble the sparse Hessian approximation from the local terms (bond lengths, bond angles, dihedral angles, and nearby electrostatics), and a subprogram to efficiently multiply this sparse Hessian by a vector. In CHARMM, the sparse Hessian multiplication routine is provided as part of the DIMB diagonalization module. A combination of LN with fast electrostatic treatments—multipoles or Ewald summations—is possible in the same way as for MTS schemes. However, as found in MTS schemes with fast summations,[10] the efficiency of the integrator, relative to traditional small-timesteps methods, decreases somewhat due to faster force routines.

It is also possible to substitute LN's linearization approach with direct calculation of the fast forces. The linearization is not crucial for LN's stability and efficiency; rather it is the combination of stochasticity with extrapolation that avoids notable resonances and offers speedup factors exceeding 5; see also the companion paper.[2] Thus, the cost benefit of this substitution (straightforward force-splitting instead of linearization plus force splitting) should be determined on the basis of the cost distribution of the associated components for the target system [see Table I, part B and equation (15)]. Note that with our linearization treatment for the proteins, more than half of the work in the fast components involves $\tilde{\mathbf{H}}$ evaluation.

In essence, the stochastic extrapolative approach of LN invites many variations and allows molecular dynamics practitioners to balance desired accuracy, as measured by the resemblance to Newtonian dynamics, with computational efficiency.

Note added in proof: We have since implemented a direct-force version of LN, and the results are similar to those described here with the linearization. These results, as well as LN performance on solvated biomolecular systems, will be reported separately (Sandu and Schlick).

ACKNOWLEDGMENTS

This work is dedicated to the memory of Edward R. Friedman, who provided the essential computer visualization support, and much more, over many years of devoted service at NYU. Support from the National Institutes of Health and the National Science Foundation is also gratefully acknowledged. We are grateful to Bruce Berne for suggesting the spectral density analyses and thank Bernie Brooks for valuable discussions. T.S. is an investigator of the Howard Hughes Medical Institute.

APPENDIX: LIN DESCRIPTION

After the linearized equations (6) are solved by system (7) or (8) for the next X_h, X_h^{n+1}, the residual component $Z(t)$ and its time derivative $W(t)$ are obtained from

$$\dot{Z}(t)=W(t),$$

$$\mathbf{M}\dot{W}(t)=-\nabla E(X_h+Z(t))-\gamma\mathbf{M}W(t)+\nabla E(X_r(t))+\tilde{\mathbf{H}}(X_h-X_r). \tag{A1}$$

The initial conditions for system (A1) are $Z(0)=0$ and $W(0)=0$.

To solve this system for the residual components, LIN applies the second-order midpoint scheme. Following alge-

braic manipulations,[6,7] we obtain a nonlinear system $\nabla\Phi(Y)=0$, where $Y=(X+X^n)/2$. This system can be solved by reformulating the problem as minimization for the ''dynamics'' function Φ:

$$\Phi(Y)=2(1+(\gamma\Delta t/2))(Y-Y_0^n)^T\mathbf{M}(Y-Y_0^n) + (\Delta t)^2E(Y), \tag{A2}$$

where

$$Y_0^n=Y_h+(\Delta t)^2/[4(1+(\gamma\Delta t)/2)]\mathbf{M}^{-1}[\nabla E(X_r) + \mathbf{H}_h(Y_h-X_r)],$$
$$Y_h=(X_h^{n+1}+X^n)/2. \tag{A3}$$

Thus, each correction step of LIN for the anharmonic residual requires nonlinear minimization of Φ. This is the cost of the large timesteps possible in LIN. This minimization subproblem can be accomplished efficiently using our truncated Newton package[52–54] with an initial approximate minimizer of Φ of X_h^{n+1} or $(X_h^{n+1}+X^n)/2$. The new coordinate and velocity vectors for timestep $n+1$ are then obtained from the relations

$$X^{n+1}=2Y-X^n, \quad V^{n+1}=V_h^{n+1}+2(X^{n+1}-X_h^{n+1})/\Delta t. \tag{A4}$$

[1] H. J. C. Berendsen, Public address, Second International Symposium of Algorithms for Macromolecular Modelling, May 21–24, 1997, Berlin.
[2] E. Barth and T. Schlick, J. Chem. Phys. **109**, 1633 (1998).
[3] T. Schlick and A. Brandt, IEEE Comp. Sci. Eng. **3**, 78 (1996).
[4] M. E. Tuckerman, B. J. Berne, and G. J. Martyna, J. Chem. Phys. **97**, 1990 (1992).
[5] H. Grubmüller, H. Heller, A. Windemuth, and K. Schulten, Mol. Simul. **6**, 121 (1991).
[6] G. Zhang and T. Schlick, J. Comput. Chem. **14**, 1212 (1993).
[7] G. Zhang and T. Schlick, J. Chem. Phys. **101**, 4995 (1994).
[8] T. Schlick, E. Barth, and M. Mandziuk, Annu. Rev. Biophys. Biomol. Struct. **26**, 179 (1997).
[9] *Computer Simulation of Biomolecular Systems: Theoretical and Experimental Applications*, edited by E. Barth, M. Mandziuk, and T. Schlick. In W. F. van Gunsteren, P. K. Weiner, and A. J. Wilkinson, Vol. III, Chap. 4 (ESCOM, Leiden, The Netherlands, 1997), pp. 97–121.
[10] R. Zhou and B. J. Berne, J. Chem. Phys. **103**, 9444 (1995).
[11] M. Watanabe and M. Karplus, J. Phys. Chem. **99**, 5680 (1995).
[12] J. J. Biesiadecki and R. D. Skeel, J. Comput. Phys. **109**, 318 (1993).
[13] M. Mandziuk and T. Schlick, Chem. Phys. Lett. **237**, 525 (1995).
[14] T. Schlick, M. Mandziuk, R. D. Skeel, and K. Srinivas, J. Comput. Phys. **139**, 1 (1998).
[15] T. Bishop, R. D. Skeel, and K. Schulten, J. Comput. Chem. **18**, 1785 (1997).
[16] B. García-Archilla, J. M. Sanz-Serna, and R. D. Skeel, SIAM J. Sci. Comp. (to be published); Also Tech. Rept. 1996/7, Dep. Math. Applic. Comput., Univ. Valladolid, Valladolid, Spain.
[17] T. Schlick, in *Proceedings of the Second International Symposium on Algorithms for Macromolecular Modelling, Konard-Zuse Zentrum, Berlin, May 21–24, 1997*, Springer-Verlag Lecture Notes in Computational Science and Engineering (Springer, Berlin, 1998).
[18] L. Verlet, Phys. Rev. **159**, 98 (1967).
[19] J. M. Sanz-Serna and M. P. Calvo, *Numerical Hamiltonian Problems* (Chapman and Hall, London, 1994).
[20] W. B. Streett, D. J. Tildesley, and G. Saville, Mol. Phys. **35**, 639 (1978).
[21] W. B. Streett, D. J. Tildesley, and G. Saville, in *Computer Modeling of Matter*, edited by Peter Lykos, Vol. 86, ACS Symposium Series (ACS, Washington, D. C., 1978), pp. 144–158.
[22] R. D. Swindoll and J. M. Haile, J. Chem. Phys. **53**, 289 (1984).
[23] M. E. Tuckerman, B. J. Berne, and G. J. Martyna, J. Chem. Phys. **94**, 6811 (1991).
[24] C. S. Peskin and T. Schlick, Commun. Pure Appl. Math. **42**, 1001 (1989).
[25] R. W. Pastor, in *The Molecular Dynamics of Liquid Crystals*, edited by G. R. Luckhurst and C. A. Veracini (Kluwer Academic, Dordrecht, The Netherlands, 1994).
[26] R. J. Loncharich, B. R. Brooks, and R. W. Pastor, Biopolymers **32**, 523 (1992).
[27] P. Derreumaux and T. Schlick, Proteins: Struct., Funct., Genet. **21**, 282 (1995).
[28] M. H. Hao, M. R. Pincus, S. Rackovsky, and H. A. Scheraga, Biochemistry **32**, 9614 (1993).
[29] A. Brünger, C. L. Brooks III, and M. Karplus, Chem. Phys. Lett. **105**, 495 (1982).
[30] T. Simonson, Chem. Phys. Lett. **250**, 450 (1996).
[31] D. Beglov and B. Roux, J. Chem. Phys. **100**, 9050 (1994).
[32] D. Beglov and B. Roux, Biopolymers **35**, 171 (1994).
[33] D. Beglov and B. Roux, J. Chem. Phys. **103**, 360 (1995).
[34] D. A. McQuarrie, *Statistical Mechanics* (Harper and Row, New York, 1976), Chaps. 20–21.
[35] R. W. Pastor, B. R. Brooks, and A. Szabo, Mol. Phys. **65**, 1409 (1988).
[36] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems*, Vol. 14, Springer Series in Computational Mathematics (Springer, New York, 1991).
[37] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus, J. Comput. Chem. **4**, 187 (1983).
[38] A. D MacKerell, Jr., D. Bashford, M. Bellott, R. L. Dunbrack, Jr., J. Evanseck, M. J. Field, S. Fischer, J. Gao, H. Guo, S. Ha, D. Joseph, L. Kuchnir, K. Kuczera, F. T. K. Lau, C. Mattos, S. Michnick, T. Ngo, D. T. Nguyen, B. Prodhom, W. E. Reiher III, B. Roux, M. Schlenkrich, J. Smith, R. Stote, J. Straub, M. Watanabe, J. Wiorkiewicz-Kuczera, D. Yin, and M. Karplus, J. Phys. Chem. (in press).
[39] B. Mishra and T. Schlick, J. Chem. Phys. **105**, 299 (1996).
[40] E. Barth, K. Kuczera, B. Leimkuhler, and R. D. Skeel, J. Comput. Chem. **16**, 1192 (1995).
[41] F. C. Bernstein, T. F. Koetzle, G. J. B. Williams, E. F. Meyer, Jr., M. D. Brice, J. R. Rodgers, O. Kennard, T. Shimanouchi, and M. Tasumi, J. Mol. Biol. **112**, 535 (1977).
[42] E. E. Abola, F. C. Bernstein, S. H. Bryant, T. F. Koetzle, and J. Weng, in *Crystallographic Databases—Information Content, Software Systems, Scientific Applications*, edited by F. H. Allen, G. Bergerhoff, and R. Sievers (Data Commission of the International Union of Crystallography, Bonn, 1987), pp. 107–132.
[43] R. Huber, D. Kukla, A. Ruehlmann, O. Epp, H. Formanek, J. Deisenhofer, and W. Steigemann, Brookhaven Protein Data Bank, 1982.
[44] A. T. Brünger and M. Karplus, Proteins: Struct., Funct., Genet. **4**, 148 (1988).
[45] B. R. Bruccoleri and M. Karplus, J. Comput. Chem. **7**, 165 (1986).
[46] P. J. Artymiuk and C. C. F. Blake, J. Mol. Biol. **152**, 737 (1981).
[47] D. A. Case, Curr. Opin. Struct. Biol. **4**, 385 (1994).
[48] C. L. Brooks III and M. Karplus, J. Chem. Phys. **79**, 6312 (1983).
[49] W. Jorgensen, J. Chandrasekar, J. Madura, R. Impey, and M. Klein, J. Chem. Phys. **79**, 926 (1983).
[50] P. Deuflhard, M. Dellnitz, O. Junge, and Ch. Schütte, Technical Report SC 96-45, Konrad-Zuse-Zentrum für Informationstechnik Berlin, Takustraße 7, D-14195, Berlin-Dahlem, December 1996.
[51] W. F. van Gunsteren and M. Karplus, Macromolecules **15**, 1528 (1982).
[52] T. Schlick and M. L. Overton, J. Comput. Chem. **8**, 1025 (1987).
[53] T. Schlick and A. Fogelson, ACM Trans. Math. Softw. **14**, 46 (1992).
[54] P. Derreumaux, G. Zhang, B. Brooks, and T. Schlick, J. Comput. Chem. **15**, 532 (1994).