Application of a Multiple Time Step Algorithm to Biomolecular Systems

Ailan Cheng* and Kenneth M. Merz, Jr.*

Department of Chemistry, 152 Davey Laboratory, The Pennsylvania State University, University Park, Pennsylvania 16802

Received: January 20, 1999; In Final Form: April 21, 1999

Herein we describe the implementation of the multiple time step method, in conjunction with a reversible integrator and the Nosé−Hoover chain method for temperature and pressure control, into ROAR 1.0. We have extensively tested the MTS method on three systems: an antifreeze peptide in water, the organic solvent dimethylformamide (DMF), and a dimyristoylphosphatidylcholine (DMPC)-based lipid bilayer. From these test simulations, we observe that the MTS method was capable of producing stable trajectories even when a long time step (e.g., 8 fs) is used, while the SHAKE method was unable to do so. The SHAKE method also disturbs the bond vibrational motion while MTS algorithm does not when the time step was smaller than 5 fs. We also observe that we can conservatively obtain a 2.5-fold speed-up using the MTS method over a SHAKE simulation using a 1 fs time step. Overall, the MTS method gives a solid speed-up over the traditional SHAKE method, while simultaneously giving much more stable trajectories.

Introduction

Computer simulation of biomolecular systems is a rather computationally intensive undertaking even when performed on modern serial or parallel supercomputers. Due to the great complexity of biological systems, the dynamics of a wide range of time scales is present and need to be sampled to generate an accurate trajectory. Generally, the dynamics of principal concern, such as those associated with typical biological transformations (e.g. catalysis), occur at a relatively long time scale. However, the fastest motions, such as bond stretching and angle bending, are the limiting factors that govern the size of the time step used in a molecular dynamics simulation study. The simulation time step has to be small enough so that those fast degrees of freedom are evolved properly, thereby ensuring numerical stability. In simulations using quantum mechanical methods, even higher frequency electronic motions are involved, which put even further constraints on the allowable time step. In addition, due to the large size of biomolecules and the associated conformational space that needs to be sampled, being able to sample as much of phase space as possible using a given amount of computational resources is of critical importance.

Several types of time-saving schemes have been proposed to deal with this problem. The SHAKE algorithm[1] was devised to constrain bonds to their equilibrium bond lengths, thereby removing the fast motions associated with bond stretching. A similar scheme can be used to constrain the bond angles. However, this procedure is rather complicated, especially for complex systems such as biomolecules. In addition, it was shown that the dynamics was adversely affected by the angle constraints.[2,3] Due to the iterative nature of the SHAKE algorithm, simulations using the SHAKE algorithm do not, in general, run efficiently on parallel computers, which are now the platforms of choice when carrying out simulations of large biomolecules.

Several multiple time step algorithms have been developed to deal with the wide range of time scales involved in the dynamics of heterogeneous systems.[4−10] In complex systems the forces can be separated into several components according to their characteristic time scales. One can then integrate the equations of motion involving the fast components with a small time step and for the slow ones a larger time step can be used. The method of Teleman and Jönnson[5] fixes the slow part while updating the fast components. This method has been shown to be unstable due to the loss of accuracy.[10] The method by Swindoll and Haile[4] requires the evaluation of the second derivatives of the force, which is expensive for complex systems.

Tuckerman et al. proposed the reference system propagator algorithm for systems where different time scales are present.[6−9] This algorithm makes no approximation related to the separation of the time scales, and hence, it is an exact algorithm. This methods was shown to improve the simulation speed by 5−10-fold for some simple atomic systems. Finally, Martyna et al. have combined the reversible multiple time step integrator with the Nosé−Hoover chain extended system methods for temperature and pressure control.[11]

The motions involved in biomolecular systems range from fast bond vibrations to slow long-range van der Waals interactions. The force field used in the simulation includes the contributions due to bond, bond angle, dihedral angle, hydrogen-bonding, and nonbonded van der Waals and electrostatic interactions.[12−15] In a simulation the most time-consuming calculation is the evaluation of the slow nonbonded forces. The frequency of bond vibration and angle bending is generally very high, while the dynamics due to the nonbonded interactions is relatively slow. Thus, it is possible to separate the forces into several components, thereby taking advantage of the multiple time step method to improve the speed of a MD simulations. This method has been applied successfully to the study of several macromolecular systems.[16−19]

Herein, we apply the multiple time step method to the study of biopolymers. Our application differs from the previous ones in the following aspects. We have combined the Nosé−Hoover chain method with the explicit reversible multiple time step method. It has been shown that Nosé−Hoover chain method samples phase space more extensively and controls temperature and pressure very well.[20,21] The trajectories generated also

10.1021/jp990231w CCC: $18.00
Published on Web 05/29/1999

correspond to an existing ensemble and there exists a conserved quantity which can be used to monitor the performance of the algorithm conveniently.[21] Moreover, the explicit reversible integrator in conjunction with the reference systems propagator algorithm (RESPA) has been shown to increase the stability and accuracy of the simulation significantly.[6−8] We have implemented the multiple time step algorithm into ROAR 1.0[22] using explicitly reversible integrators combined with the Nosé−Hoover chain temperature and pressure controlling schemes.[21] ROAR 1.0 can be used for both classical MD simulations and QM/MM coupled potential simulations. Below we examine the performance of the algorithm on several biomolecular systems in detail.

Multiple Time Step Integrator: Implementation

Here, we briefly described the multiple time step algorithm. The reversible reference system propagator algorithm formulated by Tuckerman et al.[10] is based on the Trotter factorization of Liouville operator.[23] A reversible Nosé−Hoover chain (NHC) MD integrator can be generated in the following way: Let Γ be the phase space vector; then the Liouville operator is

$$iL = \dot{\Gamma}\cdot\nabla_{\Gamma} \tag{1}$$

The state of the system at time dt can be generated by applying the propagator to the phase space vector at $t = 0$, e.g.,

$$\Gamma(\mathrm{d}t) = \mathrm{e}^{iL\ \mathrm{d}t}\Gamma(0) \tag{2}$$

Using eq 1 and the NHC equations of motion,[11,20] one can obtain the operator L and then generate the integrator using Trotter factorization. For the constant temperature algorithm, the phase space vector $\Gamma = (r,p,\xi,p_{\xi})$ and the Liouville operator iL can be derived using eq 1. Here iL can be separated into three parts,

$$iL = iL_1 + iL_2 + iL_{\mathrm{nhc}} \tag{3}$$

where

$$iL_1 = \sum_{i=1}^{N} \frac{F_i(t)}{m_i}\cdot\nabla_{v_i} \tag{4}$$

$$iL_2 = \sum_{i=1}^{N} v_i\cdot\nabla_{r_i} \tag{5}$$

$$iL_{\mathrm{nhc}} = \text{all thermostat related} \tag{6}$$

The operator associated with the thermostat variables is rather complicated and the interested reader may consult the original work for details.[11,20] Here, using Trotter factorization the propagator can be written as

$$\mathrm{e}^{iL\ \mathrm{d}t} = \mathrm{e}^{iL_{\mathrm{nhc}}(\mathrm{d}t/2)}\mathrm{e}^{iL_1(\mathrm{d}t/2)}\mathrm{e}^{iL_2\ \mathrm{d}t}\mathrm{e}^{iL_1(\mathrm{d}t/2)}\mathrm{e}^{iL_{\mathrm{nhc}}(\mathrm{d}t/2)} \tag{7}$$

For the multiple time step implementation, the forces are separated into several components and here we show an example of a three time scale derivation where the forces are separated into fast, intermediate and slow components, i.e.

$$F_i = F_i^{\mathrm{f}} + F_i^{\mathrm{m}} + F_i^{\mathrm{s}} \tag{8}$$

and correspondingly

$$iL_1 = iL_1^{\mathrm{f}} + iL_1^{\mathrm{m}} + iL^{\mathrm{s}}{}_1 \tag{9}$$

Let dt, dt/n, and dt/nm be the time steps used to evolve the slowest, the intermediate, and fastest motions, where n and m are nonzero integers; eq 7 can be written as

$$\mathrm{e}^{iL\ \mathrm{d}t} = \mathrm{e}^{iL_{\mathrm{nhc}}(\mathrm{d}t/2)}\mathrm{e}^{iL_1^{s}(\mathrm{d}t/2)}(P_m)^n\mathrm{e}^{iL_1^{s}(\mathrm{d}t/2)}\mathrm{e}^{iL_{\mathrm{nhc}}(\mathrm{d}t/2)} \tag{10}$$

where

$$P_{\mathrm{m}} = \mathrm{e}^{iL_1^{m}(\mathrm{d}t/2n)}(P_f)^m\mathrm{e}^{iL_1^{m}(\mathrm{d}t/2n)}$$

and

$$P_{\mathrm{f}} = \mathrm{e}^{iL_1^{f}(\mathrm{d}t/2nm)}\mathrm{e}^{iL_2(\mathrm{d}t/nm)}\mathrm{e}^{iL_1^{f}(\mathrm{d}t/2nm)}$$

The operator P_{f} is applied at the dt/nm time interval, while P_{m} is applied at the dt/n time interval. Finally, the rest of the operators in eq 10 are updated at the dt time interval. By applying this propagator on the coordinates, velocities, and extended variables, one can generate the modified velocity Verlet integrator for the multiple time step algorithm. The implementation procedure for a constant temperature simulation is similar to that of NHC algorithm outlined in our previous work.[21]

In eq 10, we assume that the motion associated with the L_{nhc} operator is slow compared to the other forces; thus, it is only updated with the slow force. This may not be always the case. Therefore, our implementation is designed to be flexible enough to allow the user to choose the time scale appropriate for the corresponding motions of the system of interest.

The AMBER force field for biomolecules[14,15] is used in this work. The potential energy of the system can be expressed as

$$U = U_{\mathrm{bond}} + U_{\mathrm{angle}} + U_{\mathrm{dihedral}} + U_{\mathrm{LJ}} + U_{\mathrm{el}} + U_{\mathrm{H-bond}} \tag{11}$$

where

$$U_{\mathrm{bond}} = \sum_{\mathrm{bonds}} K_r(r-r_{\mathrm{eq}})^2 \tag{12}$$

$$U_{\mathrm{angle}} = \sum_{\mathrm{angles}} K_\theta(\theta-\theta_{\mathrm{eq}})^2 \tag{13}$$

$$U_{\mathrm{dihedral}} = \sum_{\mathrm{dihedrals}}\sum_{n}\frac{V_n}{2}(1+\cos(n\phi-\gamma)) \tag{14}$$

$$U_{\mathrm{LJ}} = \sum_{i<j} 4\epsilon_{ij}\left(\frac{\sigma_{ij}^{12}}{r_{ij}^{12}} - \frac{\sigma_{ij}^{6}}{r_{ij}^{6}}\right) + \frac{1}{E_{\mathrm{scale}}^{1,4}}\sum_{i<j} 4\epsilon_{ij}\left(\frac{\sigma_{ij}^{12}}{r_{ij}^{12}} - \frac{\sigma_{ij}^{6}}{r_{ij}^{6}}\right) \tag{15}$$

$$U_{\mathrm{el}} = \sum_{i<j}\frac{q_iq_j}{r_{ij}} + \frac{1}{E_{\mathrm{scale}}^{1,4}}\sum_{i<j}\frac{q_iq_j}{r_{ij}} \tag{16}$$

$$U_{\mathrm{H-bond}} = \sum_{i<j}\left(\frac{C_{ij}}{r_{ij}^{12}} - \frac{D_{ij}}{r_{ij}^{10}}\right) \tag{17}$$

The first term, U_{bond}, describes the potential of a bond connecting two atoms by a harmonic spring of strength K_r and an equilibrium bond length r_{eq}. The second term, U_{angle}, represents the energy of an angle formed by three atoms which are connected by a harmonic spring, and K_θ and θ_{eq} are the force constant and equilibrium angle, respectively. The third term, U_{dihedral}, is the potential for a dihedral angle, ϕ, formed

by four bonded atoms expressed by a truncated Fourier series. U_{LJ} and U_{el} are the van der Waals and electrostatic interactions between two nonbonded atoms and between 1 and 4 pairs of atoms on the same molecule. U_{H-bond} is the 10-12 hydrogen bond potential used in the AMBER force field.[14] In this work, the electrostatic interactions are treated using the traditional Ewald sum method;[24] thus, the potentials are represented by a real space term and a reciprocal space term

$$E(\text{elec}) = E(\text{real}) + E(\text{reciprocal}) - E(\text{self}) - E(\text{excl}) \quad (18)$$

where the real space term is

$$E(\text{real}) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left(\sum_{|n|=0}^{\infty} q_i q_j \frac{\text{erfc}(\alpha|r_{ij}+n|}{|r_{ij}+n|}\right) \quad (19)$$

and the reciprocal space term is

$$E(\text{reciprocal}) = \frac{1}{2\pi V}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k\neq 0} q_i q_j \left(\frac{4\pi^2}{k^2}\right)\exp\left(\frac{-k^2}{4\alpha^2}\right)\times \exp(ik\cdot r_{ij}) \quad (20)$$

This term includes all the interactions due to the bonded atoms and these interactions should be excluded from the total electrostatic interactions by subtracting E(excl) from the total potential.

$$E(\text{excl}) = \sum_{i=1}^{N}\sum_{j\in N_i^{\text{excl}}}^{N}\left(q_i q_j \frac{\text{erf}(\alpha|r_{ij}|}{|r_{ij}|}\right) \quad (21)$$

Finally, the correction term due to the self-energy is

$$E(\text{self}) = \frac{\alpha}{\sqrt{\pi}}\sum_{i=1}^{N} q_i^2 \quad (22)$$

As mentioned previously, the time scales associated with biomolecular systems are rather complex. For the bonded atoms a wide range of time scales are involved; typically, the bond stretching frequency is very high. A recent study showed that the bond stretching for hydrogen atoms range from 3000 cm^{-1} for aliphatic groups to 3800 cm^{-1} for hydroxyl group.[25] The angle vibration frequency is around 500 cm^{-1} and the dynamics of torsional motions are around 200 cm^{-1}. The nonbonded interactions are relatively slow compared to the bonded interactions. However, due to the diversity of chemical groups present in many biomolecules, even the characteristic frequency of the van der Waals interactions spans a wide spectrum. For example, polar and charged groups involved in hydrogen-bonding interactions often have faster dynamics than other inert or hydrophobic groups (e.g., $-CH_3$).

Here, we briefly analyze the characteristic frequencies of some nonbonded interactions contained in the AMBER force field. The nonbonded van der Waals interactions modeled as Lennard-Jones (12-6) potential for a pair of atoms is given by

$$U_{LJ} = 4\epsilon\left(\frac{\sigma^{12}}{r^{12}} - \frac{\sigma^6}{r^6}\right) \quad (23)$$

One can expand this potential about its energy minimum by a harmonic oscillator and the characteristic time scale of the harmonic oscillator is proportional to

$$t = \sqrt{\frac{m\sigma^2}{\epsilon}} \quad (24)$$

For interactions of two nonidentical atoms, the time scale of the motion of each atom is related to its mass. Thus

$$\frac{t_1}{t_2} = \left(\frac{m_1}{m_2}\right)^{1/2} \quad (25)$$

TABLE 1: Time Scales for the Lennard-Jones Interactions in the AMBER Force Field[a]

atom	ϵ (kcal/mol)	σ (Å)	t (ps)
carbon	0.06	3.21	2.22
polar hydrogen	0.02	1.78	0.62
other hydrogen	0.01	2.74	1.34
oxygen	0.20	2.85	1.25
nitrogen	0.16	3.12	1.43

[a] Parm91 in particular.

Using the previous equations we can evaluate the time scales for the Lennard-Jones term used in the AMBER force field (see Table 1).[14] The time scales involved are much slower than that of the bonded interactions. The nonbonded interactions are the most time-consuming components in the force calculation. Therefore, the multiple time step method is very suitable under these circumstances because we will be able to update the most expensive part of the energy and force calculation much less frequently than the faster part of the calculation (i.e., bonds, angles, etc.). In addition to the time scale differences of the different components in the force filed, the forces due to the nonbonded nearest neighbors vary significantly faster than those due to atoms which are further away. Thus, one can also separate the nonbonded interactions into long- and short-range components.

In this work, the forces were separated into three time scales: fast, intermediate, and slow. There are many ways to separate the forces; thus, our program is designed so that the user can select the separation scheme according to the system under investigation. When $n = 1$ or $m = 1$, it will reduce to a two-time-scale algorithm (see eq 10). When $n = m = 1$, all forces will be updated at the same time interval, and therefore, the multiple time step method, in fact, becomes a standard integrator. Herein, we show examples of several types of separation approaches which are listed in Table 2. The first and the second separation schemes were based on the expression of the force field in eq 11. For the electrostatic interactions using the Ewald sum, the reciprocal space part is mostly long-range interactions, while the real space part involves short-range interactions. Therefore, in MTS 2 the short-range electrostatic interactions and van der Waals interactions (since those involved in hydrogen bonds are rather fast) were grouped into the intermediate time scale. For the third separation method (MTS 3) the nonbonded interactions were separated according to the distances between a pair of atoms as well; i.e., van der Waals interactions and the real space part of the Ewald sum within a 7 Å cutoff were placed in the intermediate time scale, while those beyond 7 Å were placed in the slow time scales in order to further speed-up the simulation.

Simulation Details

We have implemented the multiple time step method combined with the Nosé−Hoover chain/rRESPA algorithm into

TABLE 2: Three Force Separation Schemes Used in This Work

time scale	components of force
	MTS 1
fast	bonds
intermediate	angles, torsions
slow	nonbonded van der Waals and electrostatic (i.e., all components of Ewald sum)
	MTS 2
fast	bonds
intermediate	angles, torsions, vdW, and real space part of Ewald sum
slow	long-range electrostatic (i.e., reciprocal part of Ewald sum)
	MTS 3
fast	bonds
intermediate	angles, torsions, vdW, and real space part of Ewald sum within 7 Å
slow	long-range electrostatic (i.e. reciprocal part of Ewald sum) and vdW and real space part of Ewald sum beyond 7 Å

ROAR 1.0[22] (part of the AMBER suite of programs[26]). The code was tested on several biomolecular systems including the winter flounder antifreeze protein solvated by water,[27] a dimyristoylphosphatidylcholine (DMPC)-based lipid bilayer,[28−31] and an organic solvent dimethylformamide (DMF).[32] The AMBER force field[15] for proteins and TIP3P[33] model for water were used in the antifreeze peptide/water simulation, while the united atom models were used for the DMPC[28−31] bilayer and a neat DMF simulations, respectively. The initial configurations for the DMPC bilayer and antifreeze peptide systems were taken from our previous simulations on these systems.[27−31] The DMF simulation started with a random configuration and with velocities assigned randomly according to the Maxwell−Boltzmann distribution. The system was equilibrated for 200 ps using ROAR 1.0. Then the last configuration is taken to be the initial configuration for the testing of MTS and SHAKE algorithms.

Within the Nosé−Hoover chain extended system method, there exists a conserved quantity, the extended Hamiltonian H, which can be used to monitor the performance of the algorithm.[20,21] Conservation of the Hamiltonian reflects the stability of the integrator. Here we define

$$\Delta H = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{H(t_i) - H(0)}{H(0)}\right| \tag{26}$$

where $H(0)$ is the initial value at $t = 0$ and N is the total number of time steps. ΔH measures the degree of conservation and since this number is usually very small we have employed a logarithmic scale $\log(\Delta H)$ for convenience.

We compared the energy conservation and the CPU time of several sets of simulations to assess the accuracy and the speed of the multiple time step method relative to "standard" simulations performed using the SHAKE algorithm. We also compared several thermodynamic properties, such as the average temperature, potential energy, and pressure in the simulations. There are many factors affecting the energy conservation and speed of a given simulation. For example, the speed of a simulation is closely related to the number of the reciprocal vectors used in the Ewald calculation and the cutoff distance for the nonbonded energies. In the Ewald sum method, the convergence of the reciprocal and real space electrostatic energies depends on the parameter α in eqs 19 and 20. A large α will lead to a sharp distribution of charges and a fast convergence of the real space energy. However, more terms are needed in the k-space sum to ensure the convergence of the reciprocal part. Therefore, an intermediate value was chosen to ensure that both terms converge at an optimum computation time. With the value used here ($\alpha = 0.3$), the error in the real space energy ($\mathrm{erf}(\alpha r)/r$) was estimated to be 10^{-5} at the cutoff distance $r = 10$ Å. The k-space energy was found to converge reasonably well when 512 k-vectors were used.

Since nonbonded interactions represent the most time-consuming calculation in a classical MD simulation, a neighbor list for each atom or molecule was generated at the beginning of the simulation, in order to avoid checking all possible pairs at every time step. Constructing the list is almost as costly as evaluating the force on each atom, thus, one updates the list at some fixed time interval (e.g., 10−20 fs). Usually, when generating the neighbor list buffer region, δr is added to ensure that the exchange (i.e., moving in or out of the boundary of the cutoff region) of atoms or molecules do not adversely affect the outcome of the energy and force calculation. In addition, a sharp cutoff can cause energy drift in the simulation, so we have used a switching function $S(r)$ on the force. The switching function is the same as that used by Tuckerman.[9]

$$S(r) = \begin{cases} 1 & r < r_c - \lambda \\ 1 + R^2(2R - 3) & r_c - \lambda < r < r_c \\ 0 & r > r_c \end{cases} \tag{27}$$

where $R = [r - (r_c - \lambda)]/\lambda$ and the force is tapered to zero over a distance λ, which was chosen to be 0.4 Å. For MTS 3 (see Table 2) one will have more than one cutoff distance such as, rcut(1), rcut(2), rcut(3), etc. and the switching function was applied to each of the cutoff distances. The neighbor list for region j includes all the pairs between rcut($j-1$) − δr and rcut(j) + δr.

Results and Discussion

The first test system was a small 37-residue antifreeze peptide solvated by 1130 water molecules with periodic boundary condition.[27] The all-atom model was used in this study. The simulation was carried out for 20 ps at 300 K. The number of k-vectors is 24 × 8 × 8 and the nonbonded cutoff distance was 10 Å. Table 3 gives the time steps used for the fast, intermediate, and slow time scales together with CPU time usage and energy conservation parameters. Figure 1 shows the energy conservation (top panel) and the speed-up factor (lower panel), which is given as the ratio of CPU time relative to that for SHAKE simulation using a 1 fs time step. The simulations using SHAKE, not unexpectedly, scale almost linearly with the size of the time step. The energy conservation is reasonable for up to 4 fs; however, it becomes unstable once a time step of 5 fs is used. Furthermore, the average temperature deviates quite significantly away from the desired value especially for the simulations using a large time step (see Table 4). Since the nonbonded interactions are the most time-consuming components in the force calculation, it is better to be able to update them as infrequently as

TABLE 3: AFP/Water Simulations of 20 ps Performed on an Origin200[a]

run	dt	dt/n	dt/nm	CPU	log(ΔH)	CPU ratio
			MTS1			
1	1.0	1.0	0.25	33.72	−2.61	1.02
2	2.0	1.0	0.25	17.64	−2.54	1.95
			MTS 2			
0	0.5	0.5	0.25	57.51	−2.50	0.60
1	1.0	1.0	0.25	31.48	−2.60	1.09
2	2.0	1.0	0.25	20.47	−2.63	1.66
3	3.0	1.0	0.25	16.83	−2.65	2.05
4	4.0	1.0	0.25	14.78	−2.66	2.33
5	5.0	1.0	0.25	13.41	−0.67	2.57
6	6.0	1.0	0.25	13.77	−1.10	2.50
8	8.0	1.0	0.25	12.14	−0.30	2.84
			MTS 3			
0	0.5	0.5	0.5	71.69	−2.77	0.48
1	1.0	1.0	0.5	37.38	−2.81	0.92
2	2.0	1.0	0.5	20.65	−2.41	1.67
3	3.0	1.0	0.5	16.09	−2.82	2.14
4	4.0	1.0	0.5	13.35	−2.08	2.58
5	5.0	1.0	0.5	11.30	−0.71	3.05
6	6.0	1.0	0.5	11.82	−1.00	2.92
8	8.0	1.0	0.5	9.44	−0.30	3.65
			SHAKE			
1	1.0			34.46	−2.47	1.00
2	2.0			17.42	−2.69	1.98
3	3.0			12.02	−2.52	2.87
4	4.0			9.21	−2.01	3.74

[a] For this set of simulations the nonbonded neighbor list was updated every 10 fs. The CPU ratio is relative to a SHAKE simulation using a 1 fs time step. [b] dt, dt/n, and dt/nm are the time steps used to evolve the slow, intermediate and fast motions in the multiple time step simulations. In the simulation using SHAKE all the forces are updated at the same time interval dt. [c] CPU hours used to perform the simulation on the specified platform. [d] log(ΔH) measures the energy conservation as defined in eq 26 in the text. [e] CPU ratio, defined relative to the simulations using SHAKE at 1 fs, gives the speed-up factors. The higher the value the faster the simulation.

possible which is the case for MTS 1. However, MTS 1 only works for time step ≤ 2 fs. It seems that the van der Waals and electrostatic interactions associated with the polar hydrogen atoms evolve at faster time scales and, therefore, need to be updated more frequently. When all the nonbonded interactions are updated in the intermediate scales (MTS 2), the simulation was stable even for an 8 fs time step, although the energy conservation is less satisfactory for the larger time step than that for the smaller ones. The maximum speed-up is 3 for this set of simulations and the optimum is when a time step of 2−4 fs is used (best balance between speed and accuracy). The increased energy drift for time steps greater than 4 fs is related to the cancellation error of the electrostatic interactions for the bonded atoms.[34] When the Ewald sum was used, the intramolecular electrostatic interactions between bonded atoms (see eq 21) vary rapidly; thus, they were upgraded in the short time scale and yet were computed exactly. However, the reciprocal space energy (see eq 20) were truncated at a finite cutoff (i.e., only a finite number of k-vectors were included). Therefore, the electrostatic energies due to the bonded atoms were not completely canceled; thus, this may lead to instabilities, especially when a multiple time step method is used.

One should keep in mind that nonbonded interactions due to atoms far away vary more slowly when compared to those due to the nearest neighbors. Therefore, we tried another force separation scheme (MTS 3). In this case, only the van der Waals and real space part of the electrostatic interaction within a 7 Å cutoff were updated in the intermediate time interval. The rest

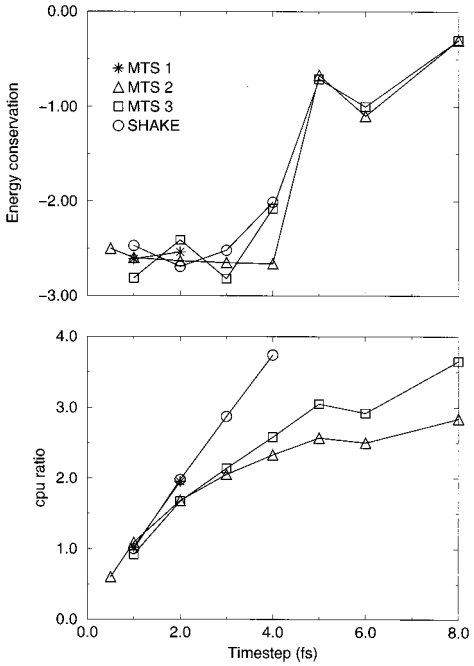

Figure 1. Energy conservation (upper panel) and the speed-up factor (lower panel) as a function of the time step used for a series of simulations of the antifreeze peptide solvated by water. The energy conservation factor is defined in the text (see eq 26), and the speed-up factor is defined relative to the CPU time used for performing a SHAKE simulation using a 1 fs time step. In this set of simulations, the neighbor lists were updated every 10 fs.

TABLE 4: Average Total, Solute and Solvent Temperature (K), Potential Energy, and Pressure for AFP/Water System[a]

run	T (K)	T (solu)	T (solv)	u (kcal/mol)	P (bar)
			MTS 1		
1	300	300	300	−12221	1776
2	300	300	300	−12178	6661
			MTS 2		
0	300	300	300	−12238	523
1	300	300	300	−12232	1699
2	300	300	300	−12232	1812
3	300	300	300	−12231	1626
4	300	300	300	−12254	1456
5	300	299	300	−13588	−417
6	300	300	300	−13223	1024
8	300	299	300	−13674	−1174
			MTS 3		
1	300	300	300	−12217	1492
2	300	300	300	−12228	1494
3	300	300	300	−12223	1412
4	300	301	300	−12248	1234
5	300	299	300	−13452	−888
6	300	300	300	−13174	731
8	300	299	300	−13606	−1681
			SHAKE		
1	297	300	297	−12523	22
2	288	299	287	−12621	58
3	273	298	269	−12830	87
4	250	298	242	−13149	163

[a] Simulation details given in Table 3.

of the nonbonded interactions were evolved at the slow time scale. A buffer region of $\delta r = 1$ Å thick was used to construct the neighbor list which is updated every 10 fs. The simulation of MTS 3 shows, expectedly, that the CPU time used scales better than that of MTS 2 method. One can achieve a factor of 2.5 speed-up at d$t = 4$ fs. Table 4 shows the average temperature of the solute and solvent together with the potential energy and

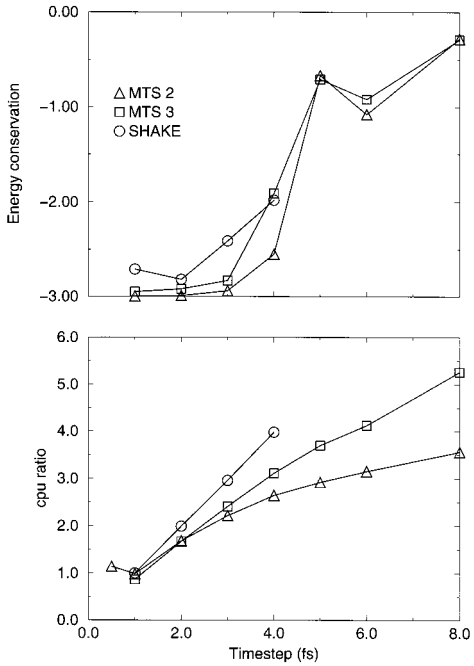

Figure 2. Same as in Figure 1, but for a set of simulations where the neighbor lists were updated every time step.

average pressure. We observe that the MTS methods control the simulation temperature much better than the SHAKE method which was not able to maintain the desired temperature for time steps 3 and 4 fs.

In the original AMBER module SANDER[26] and ROAR 1.0,[22] the buffer region option is not available, and hence, to achieve the same order of accuracy obtained here one needs to update the neighbor list every time step. This is, of course, quite costly. Typically, MD simulations are performed where the list is updated every 10−20 fs if one is not concerned about a slight energy drift during the simulation. Herein, we carried out another set of simulations where the neighbor list is updated every time step to compare the results with the above set of simulations. The simulation results, shown in Figure 2 and in Table 5, are very similar to what we obtained in the previous set, but the energy conservation was slightly better. However, the CPU time is, as expected, significantly greater. For example, the simulation using SHAKE and 1 fs time step takes 42.74 h.

We also performed test simulations using the MTS 2 scheme, but set $dt/nm = 0.5$, and we did not notice a significant difference in terms of energy conservation and CPU time required. Another set of simulations were also performed using the MTS 2 separation scheme, but the angles were evolved as a fast motion. Again, the simulation results resemble the sets discussed above. Thus, it appears that this system is tolerant of a number of settings for the time steps used.

The second set of test simulations were performed for a box of DMF molecules.[32] The simulation was performed using 216 united atom DMF molecules with periodic boundary conditions and a temperature of 300 K. The cutoff distance for the nonbonded interactions was 10 Å and the number of *k*-vectors was 512. Coordinates covering 10 ps were collected for later data analysis. We performed three sets of simulations: two using the multiple time step method and the other using SHAKE to constrain all the bonds.

Figure 3 shows the $\log(\Delta H)$ (top panel) as a function of the time step used in the simulations and the ratio of CPU time is given in the lower panel. The ratio is relative to the SHAKE simulation using a 1 fs time step (see Table 6). It is clear that

TABLE 5: Average Total, Solute, and Solvent Temperature (K), Potential Energy, and Pressure for the AFP/Water System[a]

run	*T* (K)	*T* (solu)	*T* (solv)	*u* (kcal/mol)	*P* (bar)
			MTS 2		
0	300	300	300	−12250	589
1	300	300	300	−12232	1731
2	300	300	300	−12242	1639
3	300	300	300	−12222	1637
4	300	300	300	−12242	1549
5	300	299	300	−13584	−291
6	300	300	300	−13246	917
8	300	300	300	−13677	−1374
			MTS 3		
0	300	300	300	−12249	430
1	300	300	300	−12201	1521
2	300	300	300	−12202	1485
3	300	300	300	−12212	1448
4	300	300	300	−12249	1142
5	300	299	300	−13441	−889
6	300	300	300	−13190	658
8	300	300	300	−13615	−1547
			SHAKE		
1	297	300	297	−12537	96
2	288	299	287	−12629	47
3	273	299	269	−12848	158
4	250	298	242	−13169	184

[a] In this set of simulations the neighbor lists were updated every time step.

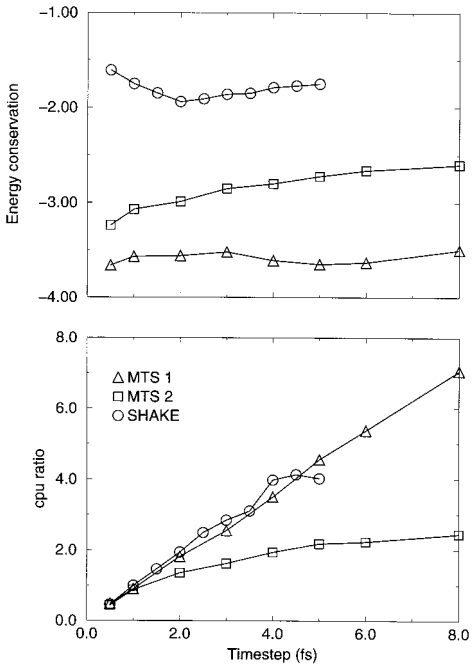

Figure 3. Same as in Figure 1, but for a set of simulations on a box of DMF molecules. In this set of simulations, the neighbor lists were updated every 10 fs.

the multiple time step method is better able to conserve the energy than is the SHAKE method. Here, both the SHAKE and MTS 1 methods scale almost linearly with the size of time step used. Since there were no explicit hydrogen atoms present in the simulation, the MTS 1 scheme is an efficient method. Nevertheless, we carried out the same simulation using the MTS 2 scheme where all forces due to van der Waals interaction and that of the real space part of Ewald sum were evaluated in the intermediate time interval (meaning the nonbonded forces were updated multiple times depending on the length of the long time step). In this case, one can only achieve a speed-up factor of

TABLE 6: DMF Simulations of 10 ps Performed on an Origin200[a]

run	d*t*	d*t*/*n*	d*t*/*nm*	CPU	log(Δ*H*)	CPU ratio
			MTS 1			
0	0.5	0.5	0.25	9.21	−3.66	0.49
1	1.0	1.0	0.25	4.99	−3.57	0.90
2	2.0	1.0	0.25	2.48	−3.56	1.82
3	3.0	1.0	0.25	1.77	−3.52	2.55
4	4.0	1.0	0.25	1.29	−3.61	3.50
5	5.0	1.0	0.25	0.99	−3.65	4.56
6	6.0	1.0	0.25	0.84	−3.63	5.37
8	8.0	1.0	0.25	0.64	−3.50	7.05
			MTS 2			
0	0.5	0.5	0.25	10.23	−3.24	0.44
1	1.0	1.0	0.25	5.13	−3.07	0.88
2	2.0	1.0	0.25	3.33	−2.99	1.35
3	3.0	1.0	0.25	2.79	−2.85	1.62
4	4.0	1.0	0.25	2.33	−2.80	1.94
5	5.0	1.0	0.25	2.07	−2.72	2.18
6	6.0	1.0	0.25	2.02	−2.66	2.23
8	8.0	1.0	0.25	1.84	−2.60	2.45
			SHAKE			
1	0.5			9.60	−1.61	0.47
2	1.0			4.51	−1.75	1.00
3	1.5			3.08	−1.85	1.46
4	2.0			2.33	−1.94	1.94
5	2.5			1.81	−1.91	2.49
6	3.0			1.58	−1.86	2.85
7	3.5			1.45	−1.85	3.11
8	4.0			1.13	−1.79	3.99
9	4.5			1.09	−1.77	4.14
10	5.0			1.12	−1.75	4.03

[a] The CPU ratio is relative to a SHAKE simulation using a 1 fs time step.

2.5 at 8 fs. It is not surprising that nonbonded interactions, the most time-consuming components in the force calculation, limited the scaling of the CPU time for the MTS 2 scheme. For DMF we are able to achieve a remarkable speed-up while still maintaining good temperature control and conserving the conserved quantity for the NHC method.

The third set of simulations was carried out on a DMPC-based lipid bilayer using a united atom model for the lipids. The system included 32 lipids with 861 water molecules. The simulation was carried out at 300 K using $16 \times 8 \times 8$ *k*-vectors. All nonbonded interactions beyond 12 Å were neglected. The coordinates covering 10 ps were collected for further data analysis. Tables 7 and 8 summarize the data obtained for these simulations and Figure 4 shows the energy conservation and ratio of CPU time for this set of simulations. The simulations using SHAKE, where all bonds were constrained to the corresponding equilibrium values, were able to go to completion even at a time step of 8 fs, although the energy conservation was not as good as that observed for the smaller time steps. Moreover, the temperature drifted significantly away from the desired value for time steps greater than 3 fs. Similar to AFP/water simulations, when MTS 1 scheme is used, the simulation was stable for time step ≤ 2 fs (the results are not shown here). For MTS 2, where all the nonbonded interactions were updated in the intermediate scale, one only obtains a speed-up of 2 which is similar to what was observed for the previous systems. The performance was improved when MTS 3 scheme was used (a speed-up of 3 was obtained). Moreover, the energy conservation for all MTS simulations was significantly better than for the SHAKE simulations.

In addition to the static thermodynamic properties, we also examined and compared the dynamics of AFP/water system using MTS 3 scheme and SHAKE algorithm. In order to do this, we estimated several types of correlation functions and their corresponding Fourier transformation. The velocity autocorrelation functions are defined as

$$C_v = \frac{\langle \sum_{i=1}^{N} v_i(t) \cdot v_i(0) \rangle}{\langle \sum_{i=1}^{N} v_i^2(0) \rangle} \tag{28}$$

Here N is the number of atoms in the system. The bond and

TABLE 7: DMPC-Based Lipid Bilayer Simulations of 10 ps Each Performed on an Origin200[a]

run	d*t*	d*t*/*n*	d*t*/*nm*	CPU	log(Δ*H*)	CPU ratio
			MTS 2			
1	1.0	1.0	0.25	18.48	−3.20	1.00
2	2.0	1.0	0.25	11.76	−3.15	1.57
3	3.0	1.0	0.25	9.93	−3.16	1.86
4	4.0	1.0	0.25	8.70	−3.16	2.12
5	5.0	1.0	0.25	8.10	−1.26	2.28
6	6.0	1.0	0.25	7.83	−1.67	2.36
8	8.0	1.0	0.25	7.02	0.07	2.63
			MTS 3			
1	1.0	1.0	0.25	25.18	−3.06	0.73
2	2.0	1.0	0.25	13.12	−2.79	1.40
3	3.0	1.0	0.25	9.51	−2.91	1.94
4	4.0	1.0	0.25	7.30	−2.98	2.53
5	5.0	1.0	0.25	6.25	−1.17	2.95
6	6.0	1.0	0.25	5.61	−1.64	3.29
8	8.0	1.0	0.25	4.48	0.09	4.12
			SHAKE			
1	1.0			18.46	−2.04	1.00
2	2.0			9.29	−2.25	1.99
3	3.0			6.33	−2.40	2.92
4	4.0			4.62	−2.34	4.00
5	5.0			3.73	−2.41	4.95
6	6.0			3.24	−1.64	5.70
8	8.0			2.34	−0.75	7.90

[a] The CPU ratio is relative to a SHAKE simulation using a 1 fs time step.

TABLE 8: Average Total, Solute, and Solvent Temperature (K), Potential Energy, and Pressure for the DMPC-Based Lipid Bilayer Simulations

run	*T* (K)	*T* (solu)	*T* (solv)	*u* (kcal/mol)	*P* (bar)
			MTS 2		
1	315	315	315	−14812	1895
2	315	315	315	−14963	1787
3	315	315	315	−14936	1903
4	315	316	315	−14958	1574
5	315	315	315	−15754	514
6	315	315	315	−15651	2210
8	316	315	316	−15570	460
			MTS 3		
1	315	315	315	−14960	1858
2	315	315	315	−15056	1870
3	315	315	315	−15044	1858
4	315	315	315	−15048	1575
5	315	315	315	−15829	371
6	315	315	315	−15728	2298
8	316	315	316	−15649	445
			SHAKE		
1	312	315	311	−15237	824
2	303	314	297	−15366	847
3	288	313	273	−15593	857
4	265	313	237	−15891	907
5	232	312	187	−16263	1160
6	190	311	121	−16654	1465
8	124	307	19	−17147	2250

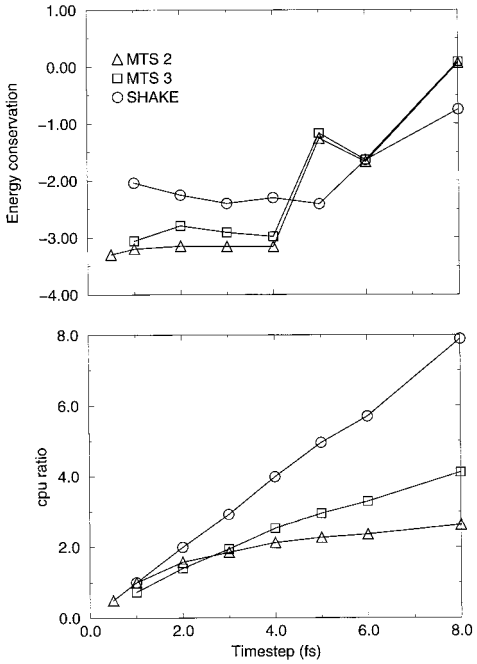

Figure 4. Same as Figure 1, but for a set of simulations of a dimyristoylphosphatidylcholine (DMPC)-based lipid bilayer. In this set of simulations, the neighbor lists were updated every 10 fs.

angle fluctuation correlation functions are defined as

$$C_\alpha = \frac{\langle \sum_{i=1}^{n} (\alpha_i(t) - \alpha_i^{av}) \cdot (\alpha_i(0) - \alpha^{av}{}_i) \rangle}{\langle \sum_{i=1}^{n} (\alpha_i(t) - \alpha_i^{av})^2 \rangle} \qquad (29)$$

where α is the bond length or torsion angle of interest and n is the total of number of bonds or angles. The superscript av stands for the average value of that variable.

The velocity autocorrelation function for the entire system gives a complete spectrum of the dynamics involved in the system. Figure 5 shows the spectrum for seven MTS simulations (MTS 3 scheme). The regions below 2000 cm^{-1} are almost identical for all simulations using a time step of up to 6 fs. The high-frequency region (i.e., from 2000 to 4000 cm^{-1}) were the same for simulations using a time step of up to 4 fs; however, clear differences are observed when the time step was greater than 5 fs. This region represents the stretching motions of bonds involving hydrogen atoms. The bond stretching motion can further be described using bond fluctuation correlation functions. We examined the spectrum of bond fluctuations for several peptide bonds and two X−H bonds: they were C_α−N, C_α−C_β, C_α-C, and N−C, backbone carbonyl C═O, N−H, and C_α−H_α. We found that the calculated frequencies for the former five bond types involving heavy atoms were nearly identical for all simulations. Those for the latter two bond types involving hydrogen atoms were also identical for simulations when a time step of 1−4 fs was used. The spectrum obtained in this way was also consistent with that estimated from the velocity autocorrelation function. Our calculated frequencies were also in good agreement with the experimental values. For example, the calculated frequencies for C═O, C_α−H_α, and N−H were 1650, 2930, and 3330 cm^{-1}, respectively, and they are all in good agreement with the experimental ranges of 1600−1690, 2850−2960, and 3310−3500 cm^{-1}, respectively. Figure 6 shows

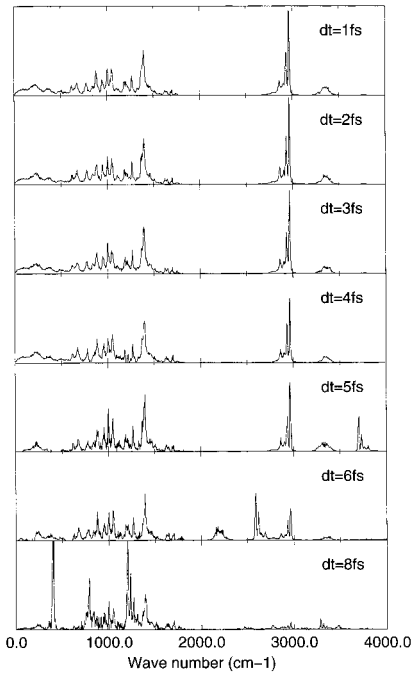

Figure 5. Fourier transformation of the velocity autocorrelation function for all peptide atoms of AFP/water simulations using the MTS method (MTS 3 scheme). Only the relative density is of interest here; thus, all the spectra in this paper are in arbitrary units. When all the plots have the same scale, the scale is not explicitly identified.

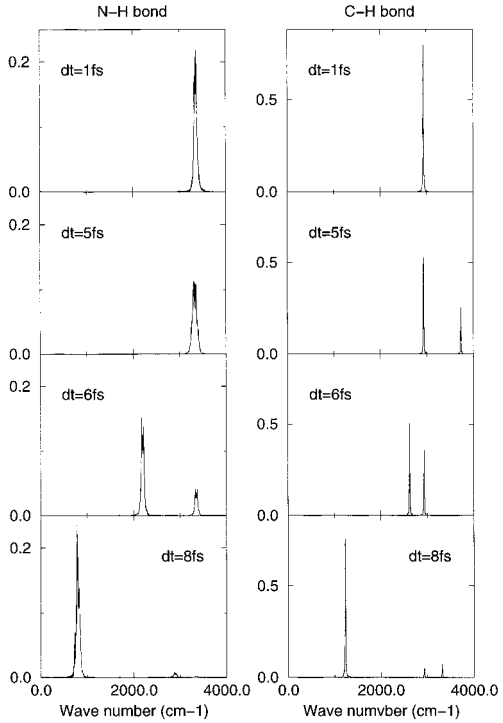

Figure 6. Fourier transformation of the bond fluctuation correlation function for the N−H (left column) and C_α−H_α (right) bonds of AFP/water simulations using the MTS 3 method.

the spectrum for the C_α−H_α and N−H bond correlation functions. When the time step was greater than 4 fs, the dynamics of these two bonds involving hydrogen atoms were clearly disrupted. For example, the spectrum of the backbone N−H bond stretching splits when a 6 fs time step was used and the major component shifts to 2100 cm^{-1}. Furthermore, when an 8 fs time step was used it shifts to 800 cm^{-1}. Obviously, the time step is too long for the high-frequency motion to be

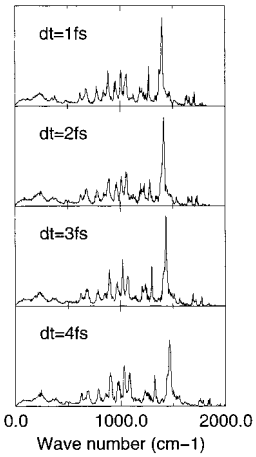

Figure 7. Fourier transformation of the velocity autocorrelation function for all peptide atoms for AFP/water simulations using the SHAKE method.

faithfully captured. Similarly, the spectrum of the $C_\alpha-H_\alpha$ bond also starts to split when a 5 fs time step was used, while at 8 fs it further shifts to a unrealistically low frequency of 1200 cm^{-1}. Hence, it appears that the stretching motion of bonds involving hydrogen atoms will be disturbed when a larger time step ($\geq$5 fs) is used in MTS simulations. As discussed previously, the reciprocal space energy also contains some short-range interactions, which may involve relatively fast dynamics, especially for the hydrogen-bonding interactions for polar hydrogen atoms. Therefore, in the simulation using a larger time step, the short-range motions in the reciprocal space components were not properly integrated, which led to unrealistic dynamics for the hydrogen atoms.

A comparison with SHAKE in the high-frequency region was not possible since all the bonds involving hydrogen atoms were constrained to their equilibrium values. Nonetheless, Figure 7 shows the complete spectrum for the four simulations using the SHAKE algorithm (*i.e.*, below 2000 cm^{-1}). The modes greater than 900 cm^{-1} begin to slowly shift toward higher frequencies as the size of the time step increases from 1 to 4 fs. The motion in this frequency range is due to bond stretching involving bonds between two heavy atoms and this type of motion becomes significantly affected as the time step increases (resulting in the upward shift of the spectrum). Thus, we conclude that all bond vibrational motions will be disturbed when SHAKE is used with an ever increasing time step.

We also examined the backbone torsion angle fluctuations as shown in Figure 8 for the simulation using MTS and in Figure 9 for that using SHAKE. Overall, we find that these torsion motions were not significantly disturbed when either the MTS or SHAKE method was used. Most of the frequencies associated with torsional motions are below 200 cm^{-1}. Similarly, the side chain torsion fluctuations involving heavy atoms did not show any dependence on the method or the size of time step used.

Conclusions

We have implemented the multiple time step (MTS) method into ROAR 1.0 and have tested the algorithm on several biomolecular systems. The simulation results show that for a typical biomolecular the MTS method can produce a stable trajectory even when a time step of 4 fs was used. When a distance-dependent separation of the nonbonded interactions was used, we achieved a conservative 2.5-fold speed-up over a 1 fs SHAKE simulation. We also found that the MTS method controls the temperature better than does SHAKE for all the

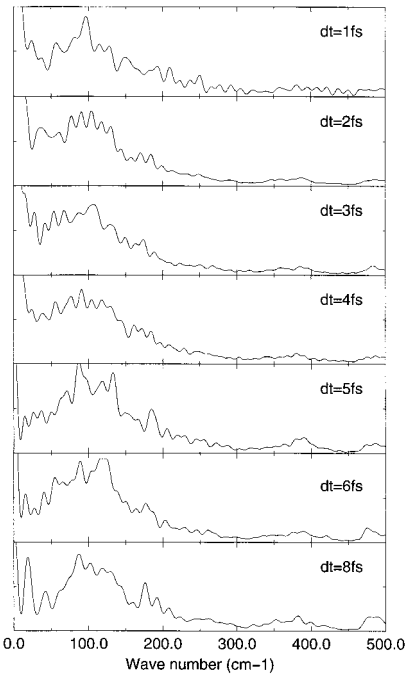

Figure 8. Fourier transformation of the fluctuation correlation function for the backbone torsion angle, ϕ(C−N−C_α−C), of AFP solvated by using the MTS 3 algorithm.

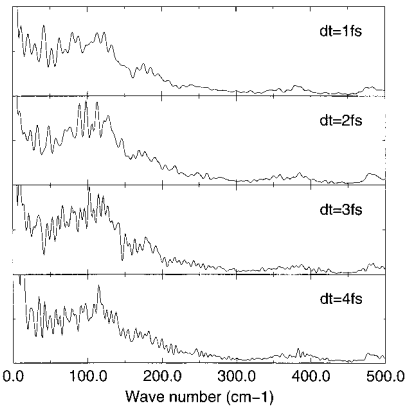

Figure 9. Same as in Figure 8, but for simulations using the SHAKE algorithm.

systems studied. Even for a large time step of 8 fs, the MTS simulations came to completion, though, the energy drifted somewhat. For special situations (e.g., DMF), one can achieve almost linear scaling using the MTS method, as can be done with the SHAKE algorithm, but the MTS method gives much better temperature control and energy conservation. The SHAKE algorithm tends to disturb all the bond vibrational motions even when a time step of 2−4 fs was used. On the other hand, the MTS method only disturbs the high-frequency vibrational motions of bonds involving hydrogen atoms when the time step is greater than 4 fs. Finally, we note that parallelization of the MTS method should be straightforward and should further enhance the performance of this approach relative to SHAKE. Overall, we found that the multiple time step method was a significant improvement over the SHAKE algorithm and will allow one to carry out much longer MD simulations at a reduced CPU cost.

Acknowledgment. We thank the DOE (DE-FGO2-96-ER62270) and Oxford Molecular for supporting this research. We also thank the Pittsburgh Supercomputer Center, the San

Diego Supercomputer Center, the National Center for Supercomputer Applications, and the Cornell Theory Center for generous allocations of supercomputer time.

References and Notes

(1) Ryckaert, J. P.; Ciccotti, G.; Berendsen, H. J. C. *J. Comput. Phys.* **1977**, *23*, 327−341.
(2) van Gunsteren, W. F.; Karplus, M. *Macromolecules* **1982**, *15*, 1528−1544.
(3) Toxvaerd, S. *J. Chem. Phys.* **1987**, *87*, 6140−6143.
(4) Swindoll, R. D.; Haile, J. M. *J. Comput. Phys.* **1984**, *53*, 289−298.
(5) Teleman, O.; Jönnson, B. *J. Comput. Chem.* **1986**, *7*, 58−66.
(6) Tuckerman, M. E.; Martyna, G. J.; Berne, B. J. *J. Chem. Phys.* **1990**, *93*, 1287−1291.
(7) Tuckerman, M. E.; Berne, B. J.; Rossi, A. *J. Chem. Phys.* **1991**, *94*, 1465−1469.
(8) Tuckerman, M. E.; Berne, B. J. *J. Chem. Phys.* **1991**, *95*, 8362−8364.
(9) Tuckerman, M. E.; Berne, B. J.; Martyna, G. J. *J. Chem. Phys.* **1991**, *94*, 6811−6815.
(10) Tuckerman, M.; Berne, B. J.; Martyna, G. J. *J. Chem. Phys.* **1992**, *97*, 1990−2001.
(11) Martyna, G. J.; Tuckerman, M. E.; Tobias, D. J.; Klein, M. L. *Mol. Phys.* **1995**, *87*, 1117−1157.
(12) Brooks, B. R.; Bruccoleri, R. E.; Olafson, B. D.; States, D. J.; Swaminathan, S.; Karplus, M. *J. Comput. Chem.* **1983**, *4*, 187−217.
(13) Jorgensen, W. L.; Tirado-Rives, J. *J. Am. Chem. Soc.* **1988**, *110*, 1655−1666.
(14) Weiner, S. J.; Kollman, P. A.; Case, D. A.; Singh, U. C.; Ghio, C.; Alagona, G.; S. Profeta, J.; Weiner, P. *J. Am. Chem. Soc.* **1984**, *106*, 765−784.
(15) Cornell, W. D.; Cieplak, P.; Bayly, C. I.; Gould, I. R.; Merz, K. M., Jr.; Ferguson, D. M.; Spellmeyer, D. C.; Fox, T.; Caldwell, J. W.; Kollman, P. A. *J. Am. Chem. Soc.* **1995**, *117*, 5179−5197.
(16) Watanabe, M.; Karplus, M. *J. Chem. Phys.* **1993**, *99*, 8063−8074.
(17) Humphreys, D. D.; Friesner, R. A.; Berne, B. J. *J. Phys. Chem.* **1994**, *98*, 6885−6892.
(18) Watanabe, M.; Karplus, M. *J. Phys. Chem.* **1995**, *99*, 5680−5697.
(19) Procacci, P.; Marchi, M. *J. Chem. Phys.* **1995**, *104*, 3003−3012.
(20) Martyna, G. J.; Klein, M. L.; Tuckerman, M. E. *J. Chem. Phys.* **1992**, *97*, 2635−2643.
(21) Cheng, A.; Merz, K. M., Jr. *J. Phys. Chem.* **1996**, *100*, 1927−1937.
(22) Cheng, A.; Stanton, R. S.; Vincent, J. J.; Damodaran, K. V.; Dixon, S. L.; Hartsough, D. S.; Best, S. A.; Merz, K. M., Jr. ROAR 1.0, 1997.
(23) Trotter, H. F. *Proc. Am. Math. Soc.* **1959**, *10*, 545−551.
(24) Allen, M. P.; Tildesley, D. J. *Computer Simulation of Liquids*; Clarendon Press: Oxford, UK, 1987.
(25) Mazur, A. K. *J. Phys. Chem. B* **1998**, *102*, 473−479.
(26) Case, D. A.; Pearlman, D. A.; Caldwell, J. W.; III, T. E. C.; Ross, W. S.; Simmerling, C.; Darden, T.; Merz, K. M., Jr.; Stanton, R. V.; Cheng, A.; Vincent, J. J.; Crowley, M.; Ferguson, D. M.; Radmer, R.; Seibel, G. L.; Singh, U. C.; Weiner, P.; Kollman, P. A. AMBER 5, 1997.
(27) Cheng, A.; Merz, K. M., Jr. *Biophys. J.* **1997**, *73*, 2851−2873.
(28) Damodaran, K. V.; Merz, K. M., Jr. *Reviews in Computational Chemistry*; VCH Publishers: New York, 1994; Volume V.
(29) Damodaran, K. V.; Merz, K. M., Jr. *Langmuir* **1993**, *9*, 1179−1183.
(30) Damodaran, K. V.; Merz, K. M., Jr. *Biophys. J.* **1994**, *66*, 1076−1087.
(31) Damodaran, K. V.; Merz, K. M., Jr. *Computer Simulation of Lipid Systems*; Damodaran, K. V., Merz, K. M., Jr., Eds.; VCH Publishers: New York, 1994; Vol. 5, pp 269−298.
(32) Jorgensen, W. L.; Swenson, C. J. *J. Am. Chem. Soc.* **1985**, *107*, 569−578.
(33) Jorgensen, W. L.; Chandrasekhar, J.; Madura, J. D.; Impey, R. W.; Klein, M. L. *J. Chem. Phys.* **1983**, *79*, 926−935.
(34) Procacci, P.; Marchi, M.; Martyna, G. J. *J. Chem. Phys.* **1998**, *108*, 8799−8803.