TARGETED MOLLIFIED IMPULSE – A MULTISCALE STOCHASTIC INTEGRATOR FOR LONG MOLECULAR DYNAMICS SIMULATIONS*

QUN MA † AND JESÚS A. IZAGUIRRE‡

Abstract. Molecular dynamics (MD) is widely used in simulations of biomolecular systems such as DNA and proteins, systems which are multiscale in nature. However, current time stepping integrators are not able to address the time scale problems. Multiscale integrators, in which the presence of "fast" modes does not affect the time integration of "slow" modes, are pressingly needed in light of the fast growing biological data generated from the many genome sequencing projects. In this paper, we present a new multiple time stepping (MTS) multiscale integrator with stochasticity built in for constant temperature molecular dynamics simulations, called Targeted Mollified Impulse method (TM). TM combines the Mollified Impulse method, which is a stabler version of Verlet-I/r-RESPA (**r**eversible **RE**ference **S**ystem **P**ropagator **A**lgorithm), and a self-consistent dissipative leapfrog integrator commonly used in dissipative particle dynamics. TM introduces the Langevin coupling in a targeted manner to stabilize the MTS integrator such that the total linear momentum is conserved and less randomness in slower modes is imposed. Numerical experiments of simple model problems provide evidence that TM samples from the canonical ensemble. Possible applications include kinetics calculations such as conformational transition rates, computation of structural quantities from a canonical ensemble, and approximation of dynamical quantities from a canonical ensemble. We present results for the last two by showing that both the radial distribution functions and the self-diffusion coefficient are correctly computed from the simulations of flexible TIP3P waters using TM with outermost time step of 16 fs and innermost time step of 2 fs. Compared to leapfrog with time step of 1 fs, the implementation of TM achieves a six-fold computational speedup, whereas Verlet-I/r-RESPA with outer time step of 4 fs and inner time step of 1 fs only achieves a three-fold speedup. We also show how to generalize the method for the simulation of biological macromolecules.

Key words. long molecular dynamics simulations, multiple time stepping, Verlet-I/r-RESPA, mollified impulse method, self-consistent dissipative leapfrog, targeted Langevin stabilization

AMS subject classifications. 34D04, 65L05, 70F04, 70F08

1. Introduction.

Biological systems are multiscale in nature. For example, the dynamics of proteins contain motions over vastly different time scales, from atomic vibrations in the order of femtoseconds to collective motions that may occur in the order of milliseconds or even seconds, a span of fifteen orders of magnitude [11, pp. 19, 20]. To efficiently simulate such systems, we need multiscale numerical methods, where the presence of fast scales does not affect the time integration of slow scales, so that long time dynamics simulations of macro biomolecular systems can be done reasonably quickly using moderate computing resources. In this paper, we introduce a new integrator for molecular dynamics (MD), termed Targeted MOLLY (TM) [51]. TM allows very large time steps for time integration of slow forces while still making it possible to compute dynamical properties correctly with the right choice of method parameters.

1.1. Motivation: Multiscale Protein Simulations.

The advance in genome sequencing projects in recent years has resulted in a rapid increase of protein sequence data which far exceeds the known protein structures. Thus, predicting the 3D native structures of proteins from the known amino acid sequence, *i.e.*, protein folding, has become pressing in structural

*The work of Q. Ma and J. A. Izaguirre was supported in part by NSF Grant BIOCOMPLEXITY-IBN-0083653, NSF CAREER award ACI-0135195 and a grant from the University of Notre Dame. A preliminary form of this paper has been published in the ACM Symposium on Applied Computing with many of the details omitted due to size limitations, cf. [51]. Here we give a full presentation of the method including implementation details for reproducibility.

†qma1@cse.nd.edu, Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556-0309, USA

‡izaguirr@cse.nd.edu, Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556-0309, USA

genomics and computational biology. Though it is plausible to use MD simulations to study the folding of proteins [15, 43–46], currently available technologies are limited in their ability to tackle the time scale gaps. For example, it took over 100 days on a CRAY T3D/T3E supercomputer using all 256 CPUs to perform a one microsecond MD simulation of a small 36-residue protein called the villin headpiece subdomain in 1998 by UCSF researchers only to observe the early stages of folding [15]. In nature, however, the folding from totally denatured structure to the native structure takes tens of microseconds or even longer even for small proteins. For large proteins, folding may take several seconds. The use of multiscale integrators will greatly facilitate the process of protein simulations.

1.2. Introducing Stochasticity: Targeted MOLLY. Due to the chaotic nature of MD, only statistical correctness of the trajectories from long MD simulations is important, cf. [67] and [41, p. 354], as the computed trajectories are overwhelmed by the effect of finite time step and finite precision. For a limited time interval the numerical solution of MD simulations is very near the exact solution of a modified Hamiltonian when using symplectic integrators [70]. However, from examination of the numerical trajectories of one-dimensional systems, it is not clear whether this result extends to very long time intervals [68].

Given that multiple time stepping symplectic integrators suffer from severe instabilities that limit the largest time step possible, as is explained below, one frequently needs to introduce some form of damping when using large time steps in order to stabilize the integrator. Introducing stochasticity into MD simulations is the least harmful way to restore energy lost through damping. Stochasticity may be introduced in the initial conditions, the boundary conditions, or the integrator itself [77, p. 299]. We introduce the stochasticity in the integrator itself in a controlled or *targeted* manner as described below, in order to minimize damage to the dynamics introduced by stochasticity.

Examples of successful non-symplectic integrators include LN [5] and LM [32], which allow larger time steps in MD by introducing stochasticity to stabilize the time integration. TM is a new multiscale, multiple time stepping (MTS) integrator that uses in its outermost level the Mollified Impulse method (MOLLY) [21, 22, 31–34], a stabler version of Verlet-I/r-RESPA, and in the innermost level the self-consistent dissipative leapfrog (SCD-leapfrog) commonly used in Dissipative Particle Dynamics [9, 17, 23, 40, 58, 59]. TM uses targeted Langevin coupling in the SCD-leapfrog, *i.e.*, introducing a random force and a dissipative force only for interacting pairs of atoms associated with the fastest normal modes in the system. The linear momentum of the system is conserved in TM.

TM has been introduced in [51] with many details omitted due to size limitations. Here we give a full presentation of the method including implementation details for reproducibility of our results.

1.3. Other Multiscale Approaches. Multiscale approaches for MD fall into two fundamentally different categories: multiscale integrators, of which this work is an example, and coarsening. Coarsening relies on model reduction, which needs careful validation and is harder to control in a precise manner.

Other examples of multiscale integrators are: (i) the asynchronous variational integrator approach of Marsden and co-workers, which produces symplectic integrators that avoid resonances by advancing different degrees of freedom asynchronously in time. Although it is a very promising method, it has not yet been implemented and validated for MD, cf. [37, 47]; (ii) the nonsymplectic time reversible integrator of Hairer and Lubich [27, 28]; and (iii) the nonsymplectic reversible averaging integrator of Reich and Leimkuhler for the special case of separable slow and fast variables [42].

Two notable examples of coarsened model approaches are: (i) Bai and Brandt's coarsened Monte Carlo, which coarsens simple polymer chains and performs Monte Carlo sim-

ulations at each spatial level to verify that the coarsened model reproduces the probability distribution function of interest [2]. It is not obvious how to produce an automatic coarsening procedure for biological macromolecules though; (ii) Balaeff and Schulten's hybrid continuum/MD models, of an elastic rod model of a DNA loop in the *lac* operon [3]. These coarsening approaches are less general though very powerful.

1.4. Applications of Targeted MOLLY. TM naturally samples the canonical ensemble. Three possible applications of TM are: (i) long-time kinetics for statistically accurate trajectories for long time limit of either a microcanonical or canonical distribution; (ii) short-time kinetics for accurate trajectories for short time simulations; and (iii) thermodynamics, such as structural and free energy calculations.

First, TM is suitable for long-time simulations of chaotic or mixing Hamiltonian systems due to its computational efficiency through large time steps. Thus one can generate multiple trajectories using TM to calculate kinetics quantities such as the transition rates for structural changes. One can also use TM to facilitate the study of ensemble dynamics in protein folding, *e.g.*, folding@home [66]: Initial conditions are chosen from some prescribed distribution and a large number of trajectories can be calculated with TM. If the governing equations of motion for the MD system are stochastic due to stochastic boundary conditions or implicit solvent, the long time limit is likely a canonical distribution; on the other hand, if the governing equations of motion are deterministic, the long time limit is likely a microcanonical distribution. In the latter case, the use of artificial Langevin coupling is speculative, since the stochastic artifacts may obscure delicate features of the dynamics, such as the decay rate of correlation functions. Such features might require a smaller time step in any case. However, in the case of large number of atoms, N, the canonical ensemble is approximately the same as the microcanonical. More precisely, the long time distribution for Langevin coupling differs from constant energy by only $\mathcal{O}(1/N)$, cf. [1, Section 2.3].

Second, TM can be used for the calculation of short-time dynamical quantities from a canonical ensemble. Ideally, one should use Newtonian dynamics with initial conditions drawn from an isobaric–isothermal ensemble. In practice, dynamical data is gathered with temperature control still on. Targeted Langevin reduces the dynamical artifacts by only targeting the fastest interactions and conserving linear momentum. We use TM to compute the self-diffusion coefficient from simulations of model systems of TIP3P water [35] with flexible bonds and angles. We are able to use 16 fs for the outer time step and 2 fs for the inner time step. The baseline for comparison is the leapfrog method with time step of 1 fs.

Third, TM can be used to compute structural quantities such as the radial distribution functions from a canonical ensemble. The targeted Langevin coupling allows these quantities to be computed more accurately as compared to other Langevin approaches because it imposes less randomness in the slower modes and thus helps to cross energy barriers. Results show that the kinetic temperature is properly bounded using TM with large time steps. We present results of the correct computation of radial distribution functions for the same system of flexible TIP3P water using large time steps.

In summary, a substantial speedup is obtained using TM with larger time steps. As an additional benefit, because the time step is larger, this new integrator is more scalable than less stable integrators for parallel MD simulations. This is due to the difficulty in parallelizing several of the popular fast electrostatic methods such as particle mesh Ewald, cf. [14,38]. This benefit may be smaller with modern multi-grid methods for fast electrostatics [63,73]. Larger time steps come at a price, since significant testing of the dynamics as well as the integrator's parameter space must be performed, cf. [8, p. 185]. This can be alleviated by automating this exploratory process, cf. [39], and is justified when performing multiple MD simulations with different initial conditions, or when running a single long simulation, which are the target

applications of this work.

2. Background. Tremendous effort has been devoted in order to achieve longer MD simulations that exploit the increasingly available computational power, including multiple time stepping algorithms such as Verlet-I [24, 25]/r-RESPA [75]. Linear and nonlinear instabilities in the Verlet-I/r-RESPA method limit its advantage over the single time stepping Verlet/leapfrog method in terms of stability region [33, 52].

A promising approach to achieve larger time steps for long time integration of chaotic Hamiltonian systems is to add a mild Langevin coupling. Examples of successful MTS integrators that allow larger time steps using this approach include LN [5] and LM [32]. LN is a 3-level MTS integrator for Langevin dynamics. It uses constant extrapolation on the outer time step, midpoint extrapolation on the medium time step, and position Verlet with BBK discretization [12] on the innermost time step. This particular formulation is that of [64]. LN's outer integrator has a linear instability that is removed by damping and the Langevin coupling. The damping coefficient, γ, needs to be 5-$20\,\mathrm{ps}^{-1}$ to stabilize LN; this damping destroys many important dynamical properties of the system. Whereas it is not possible to use LN for short-time kinetics, this method is one of the best if Langevin dynamics appropriately models the problem at hand, such as the case of long-time kinetics using implicit solvent or thermodynamic computations. LM uses an improved Langevin integrator, Langevin-Impulse [32, 72], and *Equilibrium* MOLLY [34]. Much lower values of γ, such as $\gamma = 0.2\,\mathrm{ps}^{-1}$, can be used to stabilize LM. And more importantly, because the damping coefficient is small, numerical artifacts are reduced and dynamical properties such as the self-diffusion coefficient can be computed accurately [32]. In the case of explicitly solvated proteins, LM allows Δt to be $12\,\mathrm{fs}$ [32], compared to $\Delta t = 3.3\,\mathrm{fs}$ using Verlet-I/r-RESPA/Impulse, and $\delta t = 2.2\,\mathrm{fs}$ using Verlet/leapfrog.

Thanks to MOLLY's enhanced stability and targeted Langevin coupling, the artifacts arising from computation of constant temperature molecular dynamics are reduced and larger Δt is allowed with TM as compared to LM: currently we use Δt of $16\,\mathrm{fs}$. We believe even larger steps are possible and allowed by accuracy considerations, cf. Appendix A.

2.1. MTS Integrators and Their Instabilities. MD solves the system of ODEs given by

$$\dot{\mathbf{q}} = \mathbf{M}^{-1}\mathbf{p}, \quad \dot{\mathbf{p}} = -\nabla U(\mathbf{q}), \tag{2.1}$$

where $\mathbf{q}$ is the position vector, $\mathbf{p}$ is the momentum vector, $U(\mathbf{q})$ is the potential energy, and the conservative force $F = -\nabla U(\mathbf{q})$.

In an attempt to bridge the time scale gap between simulations and the phenomena of interest, MTS integrators have been introduced and have been an area of active research for more than a decade. The prototypical algorithm is the Verlet-I/r-RESPA integrator, which splits the forces into fast and slow components, $\mathbf{F}^{\mathrm{fast}}$ and $\mathbf{F}^{\mathrm{slow}}$, and evaluates the former more frequently than the latter. The discretization of this problem using Verlet-I/r-RESPA with step size Δt for the slow part is given by Algorithm 1.

Verlet-I/r-RESPA is not a truly multiscale integrator because it is severely limited by instabilities when the longest time step Δt is a multiple of the period of the fastest motion, and a numerical instability at half the the period of the fastest motion. These results have been confirmed through numerical experiments [10] and using simple linear-force model problems [4, 64, 71, 74]. The period of the fastest motion of systems containing explicit water molecules as solvent is around $10\,\mathrm{fs}$ and thus the linear instability occurs at about $5\,\mathrm{fs}$. It turns out that some systematic drift can be observed in simulations reported in the literature even when using longest time steps around 3 or $4\,\mathrm{fs}$, cf. Figure 2 in [10], Figure 3 in [78],

$\frac{1}{2}$ **kick:**

$$\mathbf{p}_{n-1}^{+} = \mathbf{p}_{n-1} + \frac{\Delta t}{2}\mathbf{F}^{\mathrm{slow}}(\mathbf{q}_{n-1}), \tag{2.2}$$

oscillate: Propagate $\mathbf{q}_{n-1}$ and $\mathbf{p}_{n-1}^{+}$ by integrating

$$\dot{\mathbf{q}} = \mathbf{M}^{-1}\mathbf{p}, \quad \dot{\mathbf{p}} = \mathbf{F}^{\mathrm{fast}}(\mathbf{q}) \tag{2.3}$$

for an interval Δt to get $\mathbf{q}_n$ and $\mathbf{p}_n^{-}$.

$\frac{1}{2}$ **kick:**

$$\mathbf{p}_n = \mathbf{p}_n^{-} + \frac{\Delta t}{2}\mathbf{F}^{\mathrm{slow}}(\mathbf{q}_n). \tag{2.4}$$

Algorithm 1: Discretization using Verlet-I/r-RESPA.

and [29]. We have shown that the outer time step, Δt, is limited to less than one third of the period of the fastest motion due to the existence of 3:1 nonlinear instability, and possibly less than one fourth of the period of the fastest motion due to a 4:1 nonlinear instability in Verlet-I/r-RESPA [52]. Specifically, a time step barrier of 3.3 fs is imposed on Verlet-I/r-RESPA due to 3:1 instability for explicitly solvated proteins, and a time step barrier of 4.5 fs is imposed on Verlet-I/r-RESPA for SHAKE-constrained proteins solvated in rigid waters due to the combined 4:1 and 3:1 instabilities [52]. These mild nonlinear instabilities will become more relevant as the available computer power enables longer MD simulations.

However, if there were no such instabilities, *i.e.*, based on accuracy considerations alone, much larger time steps would be allowed for MTS integrators for the potentials that are split into fast and slow components, for example, using the C^1 continuous switching function [69]. In this case $\Delta t_{max} \approx 32$ fs for $cutoff = 6.5$ Å and $\Delta t_{max} \approx 80$ fs for $cutoff = 12$ Å (cf. Appendix A for more details). This motivates us to seek long time step integrators that are not troubled by instabilities as much as previous methods.

2.2. Mollified Impulse Method (MOLLY). The mollified impulse method (MOLLY) is designed to overcome the linear instability of Verlet-I/r-RESPA [21, 22, 31–34]. The longest time step can be lengthened by 50% using MOLLY [33, 34]. MOLLY defines the slow part of the potential energy at time-averaged positions, and the force is made a gradient of the potential energy. The time average is obtained by doing dynamics over vibrations using forces that produce those vibrations. Thus, $U^{\mathrm{slow}}(\mathbf{q})$ becomes $U^{\mathrm{slow}}(\mathcal{A}(\mathbf{q}))$, with the force defined as a gradient of this averaged potential, $-\nabla U^{\mathrm{slow}}(\mathbf{q})$ is replaced by $-\mathcal{A}_{\mathbf{q}}(\mathbf{q})^{\mathrm{T}}\nabla U^{\mathrm{slow}}(\mathbf{q})$, where $\mathcal{A}_{\mathbf{q}}(\mathbf{q})$ is a sparse Jacobian matrix. This perturbation compensates for finite Δt artifacts. Intuitively, averaged positions are better than instantaneous values for a rapidly changing trajectory $\mathbf{q}(t)$. Perturbing the potential rather than the force ensures that the numerical integrator remains symplectic [65]. The pre-factor $\mathcal{A}_{\mathbf{q}}(\mathbf{q})^{\mathrm{T}}$ improves the stability of Verlet-I/r-RESPA by working as a filter that eliminates components of the slow force impulse in the directions of the fast forces. Note that it has been shown that statistical properties are not sensitive to perturbations in the Hamiltonian [13], which partially explains why MOLLY works. The discretization of Equation (2.1) using MOLLY with time step Δt for the slow part is given in [30, 34].

Different averaging functions lead to MOLLY integrators with different stability and accuracy properties. Two different averaging methods, B-spline method based on explicit time

averaging [71] and *Equilibrium* averaging [34], have been devised and evaluated. MOLLY is more multiscale than Verlet-I/r-RESPA: for linear problems, it is truly multiscale – no accuracy or stability reduction (realized in *Equilibrium*); for nonlinear problems, it is less so.

B-spline MOLLY has been implemented in our experimental framework PROTOMOL, which is a component-based framework for MD simulations [55–57]. The framework has a modular design that allows for easy prototyping of complex methods, and it is freely available at `http://protomol.sourceforge.net`. In this work, we use B-spline MOLLY.

2.3. B-spline MOLLY. The MOLLY averaging can be done by numerically integrating an auxiliary, reduced problem:

$$\mathcal{A}(\mathbf{q}) = \frac{1}{\Delta t}\int_0^\infty \phi\left(\frac{t}{\Delta t}\right)\tilde{\mathbf{q}}(t)dt \tag{2.5}$$

where $\phi\left(\frac{t}{\Delta t}\right)$ is a weight function, and $\tilde{\mathbf{q}}(t)$ solves an *auxiliary* problem

$$\mathbf{M}\frac{\mathrm{d}^2}{\mathrm{d}t^2}\tilde{\mathbf{q}} = \mathbf{F}^{\mathrm{reduced}}(\tilde{\mathbf{q}}), \quad \tilde{\mathbf{q}}(0) = \mathbf{q}, \quad \frac{\mathrm{d}}{\mathrm{d}t}\tilde{\mathbf{q}}(0) = 0. \tag{2.6}$$

It was suggested to use B-spline weight functions [21], which are non-zero over a short interval. B-spline weight functions have compact support in time and thus make the method computationally feasible. We use the weight function given by Equation (2.7). This choice makes B-spline MOLLY stable when the outermost time step is half of the shortest period of the system [33].

$$\phi(s) = 0,\ s < 0 \text{ or } s > 1; \quad \phi(s) = 1,\ 0 \le s < 1; \quad \phi(s) = \frac{1}{2},\ s = 1. \tag{2.7}$$

It is straightforward to compute the $\mathcal{A}(\mathbf{q})$. Then $\mathcal{A}_{\mathbf{q}}(\mathbf{q})$ can be computed using the the chain rule with respect to each of the components of x. Assuming that the leapfrog method with time step δt used, the procedure of computing $\mathcal{A}(\mathbf{q})$ and $\mathcal{A}_{\mathbf{q}}(\mathbf{q})$ can be shown as follows: Initialization is given by

$$\begin{array}{llll} \mathbf{x} := \mathbf{q}, & \mathbf{p} := 0, & \mathbf{b} := 0, & t := 0, \\ \mathbf{x}_{\mathbf{q}} := I, & \mathbf{p}_{\mathbf{q}} := 0, & \mathbf{b}_{\mathbf{q}} := 0, & \end{array} \tag{2.8}$$

and step by step integration by

$$\begin{array}{ll} \mathbf{p} := \mathbf{p} + \frac{1}{2}\delta t\mathbf{F}^{\mathrm{reduced}}(\mathbf{x}), & \mathbf{p}_{\mathbf{q}} := \mathbf{p}_{\mathbf{q}} + \frac{1}{2}\delta t\mathbf{F}^{\mathrm{reduced}}_{\mathbf{x}}(\mathbf{x})\mathbf{x}_{\mathbf{q}}, \\ \mathbf{b} := \mathbf{b} + \frac{1}{2}\delta t\mathbf{x}\phi(t/\Delta t), & \mathbf{b}_{\mathbf{q}} := \mathbf{b}_{\mathbf{q}} + \frac{1}{2}\delta t\mathbf{x}_{\mathbf{q}}\phi(t/\Delta t), \\ \mathbf{x} := \mathbf{x} + \delta t\mathbf{M}^{-1}\mathbf{p}, & \mathbf{x}_{\mathbf{q}} := \mathbf{x}_{\mathbf{q}} + \delta t\mathbf{M}^{-1}\mathbf{p}_{\mathbf{q}}, \\ t := t + \delta t, & \\ \mathbf{b} := \mathbf{b} + \frac{1}{2}\delta t\mathbf{x}\phi(t/\Delta t), & \mathbf{b}_{\mathbf{q}} := \mathbf{b}_{\mathbf{q}} + \frac{1}{2}\delta t\mathbf{x}_{\mathbf{q}}\phi(t/\Delta t), \\ \mathbf{p} := \mathbf{p} + \frac{1}{2}\delta t\mathbf{F}^{\mathrm{reduced}}(\mathbf{x}), & \mathbf{p}_{\mathbf{q}} := \mathbf{p}_{\mathbf{q}} + \frac{1}{2}\delta t\mathbf{F}^{\mathrm{reduced}}_{\mathbf{x}}(\mathbf{x})\mathbf{x}_{\mathbf{q}}. \end{array} \tag{2.9}$$

The value $(1/\Delta t)\mathbf{b}$ is used for $\mathcal{A}(\mathbf{q})$ and $(1/\Delta t)\mathbf{b}_{\mathbf{q}}$ for $\mathcal{A}_{\mathbf{q}}(\mathbf{q})$. We continue the above integration until we reach a value of t such that $\phi(t/\Delta t)$ is zero at this value and remains zero for larger values of t. Although $\mathbf{F}^{\mathrm{reduced}}$ normally includes bonds and angles, recent evaluations of B-spline MOLLY indicate that it is possible to use bonds only in the averaging for a comparable stability with outer time step of 5 and 6 fs. The time step for MOLLY averaging, δt, should be chosen such that the leapfrog method is stable. In the above averaging, $\mathbf{F}_{\mathbf{q}} = -U^{\mathrm{reduced}}_{\mathbf{qq}}(\mathbf{q})$ is computed efficiently using analytical Hessians for bonds and angles. We have derived the analytical Hessians [50, pp. 87–92] of different energies for

the CHARMM force field [53, 54]. For water, one can perform the mollification molecule by molecule. In general systems, one can perform it on heavy-atom groups (a heavy atom and hydrogens covalently bonded to it). The Hessian-vector product can also be performed efficiently using methods like those in [49].

Once the Hessians for angles and bonds are computed, they can be assembled to a complete Hessian for the molecule. To illustrate the procedure, consider a water molecule with the oxygen numbered as atom j, and the hydrogens as i and k. For averaging and mollification purposes, B-spline MOLLY only uses bonds and angles in the reduced system. Assuming we already computed the Hessians of bond energies for atoms i and j, and j and k (H_{ij}^{bd} and H_{jk}^{bd}), and of angle energy for atoms i, j and k (H_{ijk}^{a}), then the assembled total Hessian for this molecule, H_{t}, is as follows (upper half only):

$$
\begin{aligned}
H_{\mathrm{t}}(0,0) &= H_{ijk}^{\mathrm{a}}(0,0) + H_{ij}^{\mathrm{bd}}(0,0),\\
H_{\mathrm{t}}(0,1) &= H_{ijk}^{\mathrm{a}}(0,1) + H_{ij}^{\mathrm{bd}}(0,1),\\
H_{\mathrm{t}}(0,2) &= H_{ijk}^{\mathrm{a}}(0,2),\\
H_{\mathrm{t}}(1,1) &= H_{ijk}^{\mathrm{a}}(1,1) + H_{ij}^{\mathrm{bd}}(1,1) + H_{jk}^{\mathrm{bd}}(0,0),\\
H_{\mathrm{t}}(1,2) &= H_{ijk}^{\mathrm{a}}(1,2) + H_{jk}^{\mathrm{bd}}(0,1),\\
H_{\mathrm{t}}(2,2) &= H_{ijk}^{\mathrm{a}}(2,2) + H_{jk}^{\mathrm{bd}}(1,1).
\end{aligned}
\tag{2.10}
$$

This Hessian, H_{t}, is used in place of $-F_{\mathbf{x}}$ in the integration step, Equation (2.9). Because water molecules are decoupled from each other in the mollification, we need to have one such Hessian data structure for each molecule to compute the averaged position and the mollification matrix, $\mathcal{A}(\mathbf{q})$.

3. Targeted MOLLY Integrator. By replacing the regular leapfrog integrator with a self-consistent dissipative leapfrog (SCD-leapfrog) in B-spline MOLLY, one gets Targeted MOLLY (TM). The TM discretization scheme is shown in Algorithm 2 in which $\mathbf{M} \equiv \mathrm{diag}(\cdots, \mathbf{m_i}, \cdots)$ denotes the diagonal mass matrix. TM uses the targeted Langevin coupling to stabilize MOLLY. Because it conserves linear momentum, TM is an improvement over traditional Langevin integrators such as BBK [12] and Langevin-Impulse [72]. In the same sense, TM is an improvement over LM [32].

3.1. Self-Consistent Dissipative Leapfrog Integrator. Dissipative particle dynamics (DPD), a method similar to MD, is another successful example in lengthening time steps [9, 17,23,40,58,59]. It is a coarse-grained simulation technique. Particle positions and velocities are continuous and their values are determined by knowledge of the forces acting on the particles. Besides coarse-graining, DPD introduces the Langevin force in a pair-wise manner which results in conservation of linear momentum in simulations.

The self-consistent dissipative leapfrog (SCD-leapfrog) integrator incorporates pair-wise Langevin forces and random forces in its force evaluation processes and updates velocities and velocity-dependent dissipative forces in a self-consistent manner to moderate the numerical instabilities associated with the fast motions. The SCD-leapfrog has only been used in DPD simulations, and is well studied and presented in [9,59]. It is included here for completeness. The discretization for one step is given in Algorithm 3, which is same as that in [9, 59] except that the dissipative and random forces are computed for the pairs of atoms that have fast interactions instead of for all particles. In Algorithms 2 and 3, $\{q_{ij}\} \equiv \{q_j - q_i\}$ and $\{\dot{q}_{ij}\} \equiv \{\dot{q}_j - \dot{q}_i\}$ are the collection of vectors from atom i's position and velocity to atom j's position and velocity, respectively, if between atoms i and j there is random and dissipative interactions. $\mathbf{F}^{\mathrm{fast}}(\mathbf{q}) = -\nabla \mathbf{U}^{\mathrm{fast}}(\mathbf{q})$, and for flexible waters includes bonds, angles, and short range electrostatic and Lennard-Jones forces. $\mathbf{F}^{D}(\{q_{ij}, \dot{q}_{ij}\})$ is the dissipative force,

$\frac{1}{2}$ **mollified kick:**

$$\mathbf{p}_{n-1}^{+} = \mathbf{p}_{n-1} + \frac{\Delta t}{2}\mathbf{F}^{\mathrm{slow}}(\bar{\mathbf{q}}_{n-1}), \tag{3.1}$$

oscillate: Propagate $\mathbf{q}_{n-1}$ and $\mathbf{p}_{n-1}^{+}$ by integrating the SDE

$$\mathrm{d}\mathbf{q} = \mathbf{M}^{-1}\mathbf{p}\mathrm{dt}, \tag{3.2}$$

$$\mathrm{d}\mathbf{p} = \left(\mathbf{F}^{\mathrm{fast}}(\mathbf{q}) + \mathbf{F}^{\mathrm{D}}(\{q_{ij}, \dot{q}_{ij}\})\right)\mathrm{dt} + \delta t^{-1/2}\sum_{i\neq j}\sigma\omega^{\mathrm{R}}(q_{ij})\hat{q}_{ij}\mathrm{d}W_{ij}, \tag{3.3}$$

where $\mathrm{d}W_{ij} = \mathrm{d}W_{ji}$ are independent increments of the Wiener process. We use the SCD-leapfrog scheme of Algorithm 3. The integrator uses a time step δt for an interval Δt to get $\mathbf{q}_n$ and $\mathbf{p}_n^{-}$. $\mathbf{F}^{\mathrm{D}}$ are the dissipative forces defined in Equation (3.9).

a time averaging: Using Equations (2.5)–(2.9), calculate a temporary vector of time-averaged positions $\bar{\mathbf{q}}_n$ and a Jacobian matrix, $\mathbf{J}_n = \nabla_{\mathbf{q}}\bar{\mathbf{q}}_n$. The time averaging function uses only the fastest forces, e.g., bonds and angles.

$\frac{1}{2}$ **mollified kick:**

$$\mathbf{p}_n = \mathbf{p}_n^{-} + \frac{\Delta t}{2}\mathbf{F}^{\mathrm{slow}}(\bar{\mathbf{q}}_n). \tag{3.4}$$

Algorithm 2: Targeted MOLLY discretization.

$\frac{1}{2}$ **kick:**

$$\mathbf{p}_{n-1}^{+} = \mathbf{p}_{n-1} + \frac{\delta t}{2}\left(\mathbf{F}^{\mathrm{fast}}(\mathbf{q}_{n-1}) + \mathbf{F}^{D}(\{q_{ij}, \dot{q}_{ij}\}_{n-1}) + \delta t^{-1/2}\mathbf{F}^{R}(\{q_{ij}\}_{n-1})\right), \tag{3.5}$$

oscillate:

$$\mathbf{q}_n = \mathbf{q}_{n-1} + \mathbf{M}^{-1}\mathbf{p}_{n-1}^{+}\delta t, \tag{3.6}$$

evaluate: Let $\mathbf{p}_n = \mathbf{p}_{n-1}^{+}$ and calculate $\mathbf{F}^{\mathrm{fast}}(\mathbf{q}_n)$, $\mathbf{F}^{D}(\{q_{ij}, \dot{q}_{ij}\}_n)$, and $\mathbf{F}^{R}(\{q_{ij}\}_n)$.

$\frac{1}{2}$ **kick, part (a):**

$$\hat{\mathbf{p}}_n^{-} = \mathbf{p}_{n-1}^{+} + \frac{\delta t}{2}\left(\mathbf{F}^{\mathrm{fast}}(\mathbf{q}_n) + /\sqrt{\delta t}\mathbf{F}^{R}(\{q_{ij}\}_n)\right), \tag{3.7}$$

$\frac{1}{2}$ **kick, part (b):**

$$\mathbf{p}_n = \hat{\mathbf{p}}_n^{-} + \frac{\delta t}{2}\mathbf{F}^{D}(\{q_{ij}, \dot{q}_{ij}\}_n), \tag{3.8}$$

Calculate $\mathbf{F}^{D}(\{q_{ij}, \dot{q}_{ij}\}_n)$ and go back to Equation (3.8) until convergence (or stop after a certain number of self-consistency iterations). Note that when carrying out several consecutive steps of SCD-leapfrog, the force vectors $\mathbf{F}^{R}(\{q_{ij}\}_{n-1})$ in Equation (3.5) and $\mathbf{F}^{R}(\{q_{ij}\}_n)$ in Equation (3.7) are the same because they are evaluated using the same sets of positions, $\{q_{ij}\}$, and random numbers per pair of particles, $\{\xi_{ij}\}$.

Algorithm 3: Self-consistent dissipative leapfrog for MD simulations.

which depends on the relative positions and velocities of the atoms being considered. Integration of the second term of Equation (3.3) produces random force kicks $F^{\mathrm{R}}(q_{ij})$ that are radially directed and preserve momentum. For targeted interactions between atoms i and j, these forces are calculated using the following formulae:

$$F_i^D(q_{ij}, \dot{q}_{ij}) = -\gamma \omega^D(q_{ij})(\dot{q}_{ij} \cdot \hat{q}_{ij})\hat{q}_{ij}, \tag{3.9}$$

$$F_i^R(q_{ij}) = \sigma \omega^R(q_{ij})\xi_{ij}\hat{q}_{ij}, \tag{3.10}$$

$$F_j^D(q_{ij}, \dot{q}_{ij}) = -F_i^D(q_{ij}, \dot{q}_{ij}), \tag{3.11}$$

$$F_j^R(q_{ij}) = -F_i^R(q_{ij}), \tag{3.12}$$

where $\hat{q}_{ij}$ denotes a unit vector in the direction of q_{ij}. Note that these and the conservative forces are pair-wise additive. The amplitude of the dissipative force is characterized by γ which is related to the effectiveness of damping. The random force, $\mathbf{F}^R$, is characterized by its amplitude, σ, the direction, $\hat{q}_{ij}$, and the Gaussian distributed random variable for each pair of particles, $\xi_{ij} = \xi_{ji}$, with zero mean and unit variance. The symmetry property $\xi_{ij} = \xi_{ji}$ ensures that the total momentum is conserved.

We require that the fluctuation-dissipation theorem for DPD be satisfied, , which ensures that the probability to observe a particular configuration of the targeted Langevin stabilized molecular dynamics particles is given by the Boltzmann distribution in equilibrium. That is, $\omega^R = \sqrt{\omega^D}$, and $k_{\mathrm{B}}T = \sigma^2/(2\gamma)$, where k_{B} is the Boltzmann constant and T the equilibrium temperature, cf. [17]. Note that γ has a unit of $[M]/[T]$ where $[M]$ and $[T]$ are the mass unit and the time unit used in the simulation, which are amu (atomic mass unit) and fs, respectively. Thus the unit of γ is $\mathrm{amu/fs}$.

If only one self-consistency iteration is used, this method reduces to the dissipative leapfrog method. In practice, we allow 2 to 8 iterations. The regular dissipative leapfrog scheme displays pronounced unphysical artifacts in the radial distribution function, $g(r)$, where r is the distance between two particles, and thus does not produce the correct equilibrium properties [9].

3.2. Pair-wise Targeted Langevin Coupling.. In traditional Langevin integrators the motion of every atom is damped individually, which makes them not momentum preserving, and implies that the macroscopic behavior will be diffusive [17] . TM uses pair-wise targeted Langevin couplings, *i.e.,* Langevin forces are targeted at particular pairs of atoms that have interactions which we know would cause instability. Momentum preserving implies that the macroscopic behavior will be hydrodynamic [17]. Atoms in those pairs, including covalently bonded pairs (bonds and angles) and possibly hydrogen-bonded pairs, have fast interactions. These are an $O(N)$ subset of $O(N^2)$ pairs of atoms. As an example, for water, there should be at least three pairs of Langevin forces: one for each of the two hydrogen-oxygen pairs to damp the bond stretching and one for the hydrogen-hydrogen pair to damp the "imaginary" bond stretching and thus the angle bending. A small number of pairs might arise from hydrogen bonds formed with neighboring molecules.

3.3. Sampling Properties. Since SCD leapfrog satisfies the fluctuation-dissipation relation, it reproduces the correct equilibrium behavior of the DPD model [59]. The DPD model reproduces Boltzmann statistics if the fluctuation-dissipation relation is satisfied [17]: It can be shown that the probability distribution function for the canonical ensemble satisfies the Fokker-Planck equation corresponding to the stochastic differential equations of the DPD model. TM is just an MTS version of self-consistent dissipative leapfrog with a small perturbation of the potential: MOLLY, consistent with Verlet-I/r-RESPA in the limit of infinitesimal time step, uses a potential that is just a small perturbation of that used by Verlet-I/r-RESPA.

The above suggests that TM samples from the canonical ensemble at least at small time steps. For systems with small number of atoms, N, if the Langevin coupling is weak, convergence to the canonical ensemble may take a long time. For large N, the canonical ensemble is approximately the same as the microcanonical, and the strength of the Langevin coupling is not relevant. We have investigated the sampling properties of TM using two simple model problems: a harmonic oscillator and a Fermi-Pasta-Ulam problem [50, pp. 59–63]. Results show that parameters for TM can be chosen so that it samples the canonical ensemble for the model problem.

4. Numerical Experiments. We perform simulations of three flexible TIP3P water [35] systems with fastest period around $10\,\text{fs}$: a "small," $10\,\text{Å}$ radius sphere with 423 atoms, a "medium," $20\,\text{Å}$ radius sphere with 3243 atoms and a "large," $42\,\text{Å}$ radius sphere with 30006 atoms. All the systems were equilibrated using NAMD 2.3 [36] during $100\,\text{ps}$ of simulation time by minimization followed by temperature re-scaling to 300 K. The smallest system is equilibrated by another $250\,\text{ps}$ simulation using PROTOMOLwith periodic boundary conditions before the production runs. By equilibrating we avoid highly improbable values of different contributions to energies. We then ran simulations using PROTOMOL. Results discussed in this section can be reproduced using PROTOMOL. All simulations use periodic boundary conditions and Ewald sum for computing full electrostatic interactions [1, 16, 18, 20, 26, 62, 76].

The potential energy function for an electrostatic interaction is given by

$$U_{ij}^{\text{electrostatic}} = \epsilon_{14}\,\kappa\,\frac{d_i\,d_j}{|q_{ij}|} \tag{4.1}$$

where ϵ_{14} is scaling factor for $1-4$ interactions, $\kappa = 8.98755 \times 10^9\,\text{Nm}^2\text{C}^{-2}$ is the Coulomb constant, d_i, d_j are charges for atom i and j, $|q_{ij}| = \|q_j - q_i\|$ is the distance between atoms i and j. Coulomb energies were split into fast and slow parts by using the real space value of Ewald sum as fast force, and the k-space value of Ewald sum as the slow force. Note that this choice of Ewald splitting is suboptimal for accuracy, cf. Section 5.

The energy for a Lennard-Jones interaction is

$$U_{ij}^{\text{Lennard-Jones}} = 4\epsilon_{ij}\left(\left(\frac{\sigma_{ij}}{|q_{ij}|}\right)^{12} - \left(\frac{\sigma_{ij}}{|q_{ij}|}\right)^{6}\right) C^2(|q_{ij}|), \tag{4.2}$$

where ϵ_{ij} and σ_{ij} are the Lennard-Jones energy minimum and cross over point (where the LJ function is zero).

Lennard-Jones energies were split using the following C^2 switching function: [29]

$$C^2(|q_{ij}|) = \begin{cases} 1 & \text{if } |q_{ij}| \le q_0, \\ \dfrac{(|q_{ij}|^2 - {q_c}^2)^2({q_c}^2 + 2|q_{ij}|^2 - 3{q_0}^2)}{({q_c}^2 - {q_0}^2)^3} & \text{if } q_0 \le |q_{ij}| < q_c, \\ 0 & \text{if } |q_{ij}| > q_c, \end{cases} \tag{4.3}$$

where $q_c \equiv \mathit{cutoff}$ is the distance where the function value becomes zero, and $q_0 \equiv \mathit{switchon}$ that where it becomes active.

The energy for a bond interaction is

$$U_k^{\text{bond}} = \frac{1}{2}K_{\text{B}}(|q_{ij}| - l_k)^2, \tag{4.4}$$

where K_{B} is a bond force constant and l_k is a reference bond length between atoms i and j for constraint k. Finally, the energy for an angle interaction is

$$U_k^{\mathrm{angle}} = \frac{1}{2} K_{\mathrm{A}} (\theta_k - \theta_0)^2, \tag{4.5}$$

where K_{A} is an angle force constant, and θ_k and θ_0 are the current value of the angle and the reference angle for angle constraint k.

For flexible water, $K_{\mathrm{A}} = 55$ kcal mol^{-1} degrees^2, $K_{\mathrm{B}} = 450$ kcal mol^{-1} Å^2, $d_{\mathrm{H}} = 0.417$ e, $d_{\mathrm{O}} = -0.834$ e, $l_{\mathrm{O-H}} = 0.957$ Å, and $\theta_0 = 104.52$ degrees. The Lennard-Jones parameters are $\sigma_{\mathrm{H-H}} = 0.4$ Å, $\sigma_{\mathrm{O-O}} = 3.1506$ Å, $\sigma_{\mathrm{O-H}} = 1.75253$ Å, $\epsilon_{\mathrm{H-H}} = 0.046$ kcal mol^{-1}, $\epsilon_{\mathrm{O-O}} = 0.1521$ kcal mol^{-1}, $\epsilon_{\mathrm{O-H}} = 0.08365$ kcal mol^{-1}.

We use a 2-level Targeted MOLLY integrator with outer time step Δt of 16 fs and inner time step δt of 2 fs. The B-spline MOLLY integrator in the outer level uses bonds and angles in the averaging, and it uses Equation (2.7) as its averaging function. The SCD-leapfrog in the inner level uses pairwise Langevin interactions of bonds and angles. The weight functions, ω^D and ω^R, are both set to 1 in this evaluation since the distance between the bonded pairs of atoms does not vary much. We may also adopt the C^1 continuous switching function as our weight function ω^R. Such a choice of weight function may be more suitable for nonbonded interacting pairs since the distance between any such pair of atoms vary more rapidly. The MOLLY averaging step size is 2 fs. The Langevin coupling coefficient, γ, is $4\,\mathrm{amu/ps}$. The number of iterations for self-consistency for all simulations is 2. The Ewald accuracy, $\epsilon \equiv \exp(-p)$, is 10^{-12} for the small water system, and 10^{-7} for the two larger water systems. The relationship between ϵ, α (the splitting factor of the the real- and reciprocal space terms), and r_c and k_c (real- and k-space $cutoff$) is given by

$$r_c = \frac{\sqrt{p}}{\alpha}, \qquad k_c = 2\alpha\sqrt{p}. \tag{4.6}$$

where we used the optimal value of α as given by [19, 60]:

$$\alpha = \sqrt{\pi} \left(\frac{t_R}{t_F} \frac{N}{V^2} \right)^{\frac{1}{6}}, \tag{4.7}$$

where t_R and t_F are the execution times needed to evaluate a single term in the real- and reciprocal-space sums, respectively, and N and V are the number of atoms and volume in a periodic cell, respectively. A representative value of t_R/t_F specific to PROTOMOL has been established as 5.5. Though this will vary on different processors and for different potentials its value is not critical since it enters the equations as a sixth root. Short forces include bonded forces plus the real part of the Ewald sum with corrections. For the smallest system, the parameters were $switchon = 4$ Å and $cutoff$ of 6.5 Å; for the two larger systems, the parameters were $switchon = 6.5$ Å and $cutoff$ of 8.0 Å.

5. Results and Discussion. We show that TM recovers the correct radial distribution functions and the self-diffusion coefficient even when very large time steps are used for simulating flexible waters while maintaining a properly bounded temperature. We also show that the computational overhead associated with mollification is low. Extensions of TM to handle larger molecules are addressed.

Although TM recovers correct radial distribution functions and self-diffusion coefficient for flexible waters at very large time steps, we should note that TM is not a substitute for Hamiltonian dynamics unless the system size, N, is large enough, given that the random and dissipative forces alter the nature of the dynamics.

5.1. Properly Bounded Temperature. From the simulations with TM (with outer time step of 16 fs and inner time step of 2 fs), we show that the kinetic temperature of the system is properly bounded around the prescribed equilibrium temperature. The length of each simulation is 400 ps. We measure the relative drift of molecular temperature, denoted by ΔT_r, in percent, with respect to mean temperature, $\bar{T}$, in Kelvin. In the simulation of the small water system, $\Delta T_r = -1.58 \pm 3.78\%$ with mean temperature of 297.30 K (the prescribed equilibrium temperature is 300 K). The instantaneous kinetic temperature is given by

$$T_k(t) = \sum_{i=1}^{N} \frac{m_i v_i^2(t)}{k_{\mathrm{B}} N_f}, \tag{5.1}$$

where k_{B} is the Boltzmann constant, N_f is the degrees of freedom ($N_f = 3N - 3$ for a system of N particles with fixed total momentum), m_i is the atom weight for atom i, v_i is the velocity of atom i. We block-average over many instantaneous values to get an accurate estimate of the temperature. We also used only the non-hydrogen atoms (*i.e.*, oxygens for waters) to approximate the *molecular* temperature.

5.2. Correct Radial Distributional Functions. TM recovers the structural quantities such as the radial distribution functions. We show agreement between TM (with outer time step of 16 fs and inner time step of 2 fs) and leapfrog (with step size of 1 fs) on the radial distribution function, $g(r)$, computed from the simulations, see Figs. 5.1 and 5.2. The length of each simulation is 400 ps.

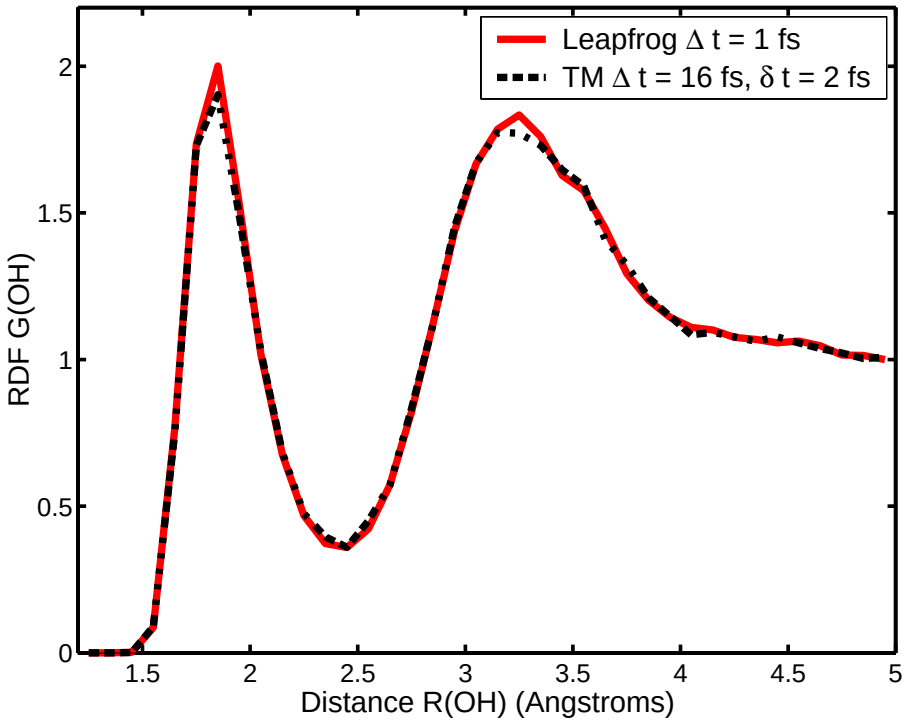

FIG. 5.1. *Radial distribution functions for the O-H interactions for leapfrog ($\delta t = 1$ fs) and Targeted MOLLY (TM, $\Delta t = 16$ fs and $\delta t = 2$ fs). It is seen that the structural property of the system is very well recovered using TM even with large time steps.*

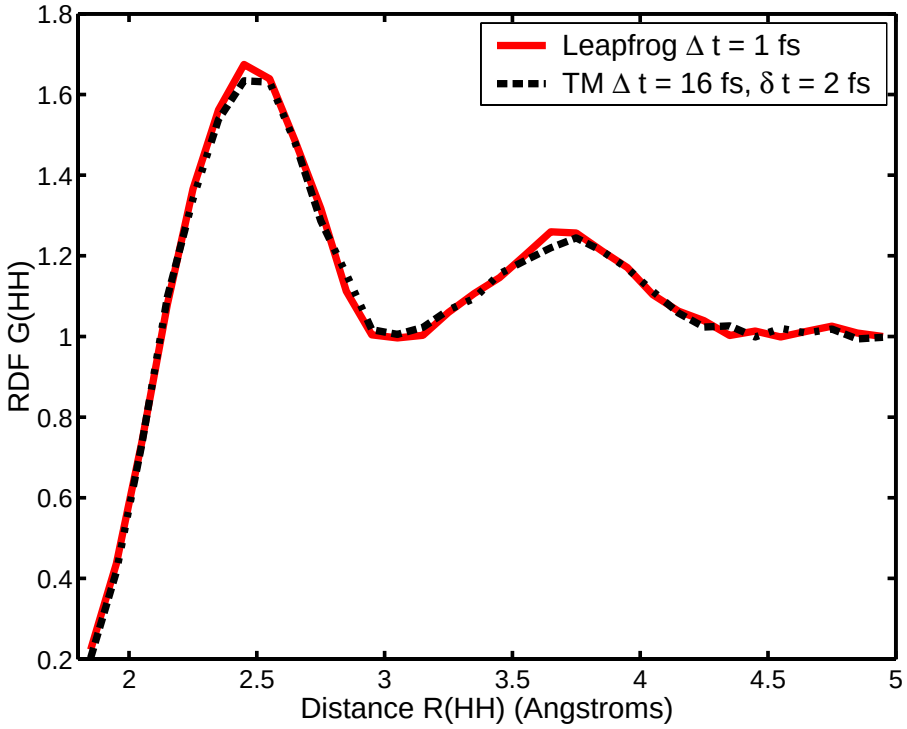

FIG. 5.2. *Same as Fig. 5.1 except for H-H interactions.*

TABLE 5.1

Self-diffusion coefficient (denoted by D) for MD simulation of 141 flexible TIP3P water molecules using different integrators with periodic boundary conditions. The Ewald accuracy is $\epsilon = 10^{-12}$. The error bar is given by twice of the standard deviation.

Integrator Type	Δt (fs)	δt (fs)	γ (amu/ps)	D ($10^{-5}\mathrm{cm}^2/\mathrm{s}$)
leapfrog	-	1	-	3.69 ± 0.01
TM	16	2	4	3.68 ± 0.01

5.3. Correct Self-Diffusion Coefficient. The self-diffusion coefficients computed from different simulations are summarized in Table 5.1. The diffusion coefficient is computed using Einstein's relation for non-overlapping 4.8 ps blocks of 400 ps trajectories, averaging over all time origins of oxygen atoms only. From Table 5.1 we can see that the self-diffusion coefficient is computed correctly using TM as compared with the result using leapfrog. The CPU times of different simulations (simulation length for the smallest system is 400 ps and that for the two larger systems is 10 ps.) are summarized in Tables 5.2 and 5.3 from which we can see a substantial speedup using TM. Compared to leapfrog with time step of 1 fs, TM with $\Delta t = 16$ fs and $\delta t = 2$ fs achieves a six-fold computational speedup, whereas Verlet-I/r-RESPA with outer time step of 4 fs and inner time step of 1 fs only achieves a three-fold speedup. In contrast, LM only computes self-diffusion coefficient correctly with $\Delta t \leq 12$ fs [32] .

Note that one needs to tune the parameters for TM in order to recover correct dynamics. As an example, after choosing an inner time step of $\delta t = 2$ fs, the proper values for γ are 0.7

TABLE 5.2

CPU time (denoted by t) and speed up (denoted by η) of 400 ps of MD simulations of 141 flexible TIP3P water molecules with periodic boundary conditions. The Ewald accuracy is $\epsilon = 10^{-12}$. All simulations were performed on a single node of the Hydra cluster (SUN OS 2.7, 32 nodes each of which is 440 MHz dual-processor SunSparc 10 with a 256 MB shared memory).

Integrator Type	Δt (fs)	δt (fs)	γ (amu/ps)	t (hr)	η -
leapfrog	-	1	-	68.28	1.00
Verlet-I/r-RESPA	4	1	-	23.50	2.91
TM	16	2	4	11.40	5.99

TABLE 5.3

CPU time (denoted by T) and speed up (denoted by η) of 10 ps molecular dynamics simulations of 1081 and 10002 flexible TIP3P water molecules with periodic boundary conditions. The Ewald accuracy is $\epsilon = 10^{-7}$. Each simulation was performed on a single node of the Hydra cluster (SUN OS 2.7, 32 nodes each of which is 440 MHz dual-processor SunSparc 10 with a 256 MB shared memory)

Integrator Type	Δt (fs)	δt (fs)	γ (amu/ps)	3243 atoms T (hr)	η	30006 atoms T (hr)	η
leapfrog	-	1	-	20.02	1.00	1021.31	1.00
Verlet-I/r-RESPA	4	1	-	13.01	1.54	538.92	1.89
TM	16	2	4	6.28	3.20	237.91	4.29

and 0.9 amu/ps for $\Delta t = 8$ fs and $\Delta t = 12$ fs, respectively. These results suggest that the value of γ is a nonlinear function of Δt. Nonetheless, γ goes to zero as Δt goes to zero.

5.4. Low Computational Overhead of Mollification. The computational overhead of mollification is estimated in the range of $1.5 - 5\%$ of the total execution time, depending on the cost of non-bonded force evaluations, see Table 5.4. The larger the cost of the non-bonded force evaluation, the smaller percentage of the overhead incurred by the mollification because this overhead is linear in terms of number of atoms with a very small constant. For TM with $\Delta t = 16$ fs and $\delta t = 2$ fs, the overhead associated with mollification is estimated as about $1 - 2\%$ of the total execution time in which both bonds and angles are included in the MOLLY forces.

5.5. Extension of TM to Handle Larger Molecules. The method was tested on flexible water, however, it is straightforward to extend the method to work with larger molecules such as proteins and DNAs. The averaging and mollification procedures operate on heavy-atom groups instead of molecules, see Appendix B. Since the heavy-atom group construction is only performed once in the beginning of a simulation, the run time overhead of TM for general systems will not be much different from that for flexible water systems.

5.6. Larger Time Steps with Better Force Splitting. While the fast and slow forces splitting of Ewald sum used in these simulations is easy to implement [61], it is suboptimal for accuracy and therefore more appropriate splittings in Ewald sum have been suggested [6, 7, 79]. When using a better, distance-based splitting rather than the regular real- /reciprocal-space splitting, researchers have been able to use a much larger Δt in their simulations of solvated proteins, cf. [79, Section IV]. These results suggest that using a more appropriate splitting, TM should allow even larger time steps.

6. Future Work. Given that a general theory is lacking, more analysis of simple model problems has to be done and more numerical evidence has to be collected to show that TM can

TABLE 5.4

Overhead of mollification for a system of 141 *flexible TIP3P water molecules. Each simulation is* 4 *ps long. All simulations are performed on a single node of a Linux cluster composed of processors of Intel (R) Xeon (TM) 1.7 GHz, 1GB memory, and 2.4.17 kernel. For the integrators, Imp stands for Verlet-I/r-RESPA and BsM stands for B-spline MOLLY. In name-of-integrator(*x*,*y*,*z*),* x *is the outer time step,* Δt *(fs);* y *is the inner time step,* δt *(fs); and* z *is the time step (fs) for MOLLY averaging.*

Force method	MOLLY forces	Integrator	Run time (s)	Overhead(%)
Ewald (10^{-6})	-	Imp(4,1,-)	311.12	-
		Imp(4,2,-)	168.31	-
	Bond	BsM(4,1,1)	321.30	3.17
		BsM(4,1,2)	316.91	1.83
		BsM(4,2,2)	172.69	2.54
	Bond+Angle	BsM(4,1,1)	329.72	5.67
		BsM(4,1,2)	320.47	2.92
		BsM(4,2,2)	176.66	4.73
Ewald (10^{-12})	-	Imp(4,1,-)	584.40	-
		Imp(4,2,-)	320.61	-
	Bond	BsM(4,1,1)	597.80	2.24
		BsM(4,1,2)	594.71	1.73
		BsM(4,2,2)	326.70	1.87
	Bond+Angle	BsM(4,1,1)	606.30	3.61
		BsM(4,1,2)	603.80	3.21
		BsM(4,2,2)	328.72	2.47

be used for long time kinetics of deterministic equations of motion for large N. We will also investigate whether TM can be used for rigorous study of other structural and thermodynamic properties.

The choice of TM parameters, such as γ, δt, Δt, and number of self-consistency iterations remains to be determined. Our previous work in [32] and Appendix A gives an indication on how to proceed.

A more stable version of MOLLY for TM should allow larger time steps. We are exploring the effect of inclusion of a subset of short-range non-bonded interactions in the MOLLY averaging. We may also need other alternatives for achieving the self-consistency in Algorithm 3, which may help conserve equilibrium temperature better.

Finally, when combined with coarse-graining, TM might be a competitive method for dissipative particle dynamics. Inviscid fluids and celestial mechanics are areas where this research might be beneficial. These directions need to be investigated.

Acknowledgments. Some of the calculations were performed at the Hydra cluster (SUN OS 2.7, 32 nodes each of which is 440 MHz dual-processor SunSparc 10 with a 256 MB shared memory) owned by the College of Engineering of the University of Notre Dame, the BOB Beowulf cluster (110 nodes, Intel (R) Xeon (TM) 1.7 GHz dual-processor, 1GB of memory per node, and Linux 2.4.17 kernel) owned by the College of Science and College of Engineering at the University of Notre Dame, and the Norwegian Super-computing Facilities (IBM p690 Regatta H with AIX 5.1 and SGI Origin 3800 with IRIX 6.5) through a Norwegian Research Council grant. Thierry Matthey helped in running many of the simulations. The authors are grateful to R. D. Skeel for valuable discussions and suggesting the name *targeted Langevin interactions.* The authors are also thankful to Dr. P. Procacci, Dr. R. Zhou and Dr. D. Gezelter for valuable comments and discussions.

Appendix A. Estimate of the Upper Limit of Outer Step Sizes. The largest allowable time steps based on accuracy concerns can be estimated according to the characteristic time of the system [48]:

$$t_{\mathrm{char}=2\pi(\rho(\mathbf{M}^{-1/2}\mathbf{U}_{\mathbf{xx}}^{\mathrm{slow}}\mathbf{M}^{-1/2}))^{-1/2}} \tag{A.1}$$

where $\rho(A)$ is the spectral radius of the matrix A. The characteristic time corresponds to the fastest normal mode of the slow interactions, which happens between two atoms separated by cutoff distance, q_c. To retain enough accuracy in incorporating the slow force contributions to the solution to the governing equations of motion, it is necessary to incorporate the slow force contributions every one tenth of the characteristic time of the slow forces. The value of one tenth of t_{char} for the time step ΔT gives an error of 1.5% in the fastest normal mode of a quadratic potential.

We estimate the characteristic time of a solvated molecular system through the use of a 1-d model. The model consists of two atoms who interact with each other through the Coulombic potential, Equation (4.1). And these two atoms each belong to a different water molecule and are separated by a distance greater than the cutoff q_c. In our model, $m_H = 1.008$ amu, $m_O = 15.9994$ amu, $d_H = 0.417$ e and $d_O = 0.814$ e (1amu $= 1.6605402 \times 10^{-27}$kg, 1e $= 1.60217733 \times 10^{-19}$C). For our simple model, the energy function and Hessian reduce to very simple forms. Assume the equation of the Coulomb energy is given by

$$U(x_1, x_2) = C_1 / \left| x_1 - x_2 \right|, \tag{A.2}$$

where $C_1 = \epsilon_{14}\kappa d_i d_j$. For different interactions, this constant is $C_1^{H-H} = 4.0118 \times 10^{-29}$, $C_1^{H-O} = 8.0235 \times 10^{-29}$ and $C_1^{O-O} = 1.6047 \times 10^{-28}$. Without loss of generality, assume $x_1 > x_2$. Then the Hessian, U_{xx}, becomes the following:

$$\mathrm{U}_{xx} = 2C_1 \begin{bmatrix} \frac{1}{(x_1-x_2)^3} & \frac{-1}{(x_1-x_2)^3} \\ \frac{-1}{(x_1-x_2)^3} & \frac{1}{(x_1-x_2)^3} \end{bmatrix}. \tag{A.3}$$

Assume the mass matrix is $\mathbf{M} = \mathrm{diag}(\mathbf{m_1}, \mathbf{m_2})$. Thus

$$B = 2C_1 \begin{bmatrix} \frac{1}{(x_1-x_2)^3 m_1} & \frac{-1}{(x_1-x_2)^3\sqrt{m_1 m_2}} \\ \frac{-1}{(x_1-x_2)^3\sqrt{m_1 m_2}} & \frac{1}{(x_1-x_2)^3 m_2} \end{bmatrix}, \tag{A.4}$$

whose spectral radius is $\rho(B) = \max\{\lambda_1, \lambda_2\} = \frac{2C_1(m_1+m_2)}{m_1 m_2 (x_1-x_2)^3}$. By making $x_1 - x_2 = q_c$, and plugging in the known variables, we can compute the spectral radius and thus the characteristic time for different q_c values very easily. O-H interactions are the fastest among O-H, H-H and O-O interactions. Thus, we have Equation (A.5) to compute the maximum allowable step sizes for different cutoff values.

$$\Delta t_{\max} = 32.61(q_c/6.5)^{1.5}. \tag{A.5}$$

Examples of maximum allowable time steps are $\Delta t_{max} \approx 32$ fs for $q_c = 6.5$ Å, and $\Delta t_{max} \approx 96$ fs for $q_c = 13.5$ Å. Note that this procedure can be used to estimate characteristic of other systems as well.

Appendix B. Extension of TM to General Molecular Systems. The averaging and mollification procedure in B-spline MOLLY and TM could operate on heavy-atom groups so that the methods can work with larger molecules such as proteins and DNA. A heavy-atom

group consists of one non-hydrogen atom such as oxygen, carbon, nitrogen, sulphur, *etc*, and one or more hydrogen atoms bonded to the heavy atom. Given a larger molecule, one can construct a heavy-atom group list by going through the bond list and angle list created in the topology data structure for the molecule. In the heavy-atom group list, the lists for bonds and angles in each group should be present. It now becomes trivial to assemble the Hessian for each heavy-atom group. After the Hessian is assembled, the mollification is done on heavy-atom groups. Suppose we now have a heavy-atom group that consists of 3 hydrogens (indexed as 1, 2 and 3) and one carbon (indexed as 0), and there are 3 polar C-H bonds, *i.e.*, B01 for 0-1, B02 for 0-2 and B03 for 0-3, and 3 angles, *i.e.*, A102 for 1-0-2, A103 for 1-0-3 and A203 for 2-0-3), then the total Hessian, $H_{\rm t}$, is as follows (only the upper half):

$$H_{\rm t}(0,0) = B01(0,0) + B02(0,0) + B03(0,0), \tag{B.1}$$

$$H_{\rm t}(0,0) += A102(1,1) + A103(1,1) + A203(1,1), \tag{B.2}$$

$$H_{\rm t}(0,1) = B01(0,1) + A102(1,0) + A103(1,0), \tag{B.3}$$

$$H_{\rm t}(0,2) = B02(0,1) + A102(1,2) + A203(1,0), \tag{B.4}$$

$$H_{\rm t}(0,3) = B03(0,1) + A102(1,2) + A203(1,2), \tag{B.5}$$

$$H_{\rm t}(1,1) = B01(1,1) + A102(0,0) + A103(0,0), \tag{B.6}$$

$$H_{\rm t}(1,2) = A102(0,2), \tag{B.7}$$

$$H_{\rm t}(1,3) = A103(0,2), \tag{B.8}$$

$$H_{\rm t}(2,2) = B02(1,1) + A102(2,2) + A203(0,0), \tag{B.9}$$

$$H_{\rm t}(2,3) = A203(0,2), \tag{B.10}$$

$$H_{\rm t}(3,3) = B03(1,1) + A103(2,2) + A203(2,2). \tag{B.11}$$

REFERENCES

[1] M. P. Allen and D. J. Tildesley. *Computer Simulation of Liquids*. Clarendon Press, Oxford, New York, 1987. Reprinted in paperback in 1989 with corrections.

[2] D. Bai and A. Brandt. Multiscale algorithms in molecular dynamics. In A. Brandt, J. Bernholc, and K. Binder, editors, *Multiscale Computational Methods in Chemistry and Physics*, volume 177, pages 250–266, Amsterdam, Netherlands, 2001. IOS Press.

[3] A. Balaeff, L. Mahadevan, and K. Schulten. Elastic rod model of a DNA loop in the *lac* operon. *Phys. Rev. Lett.*, 83(23):4900–4903, 1999.

[4] E. Barth and T. Schlick. Extrapolation versus impulse in multiple-timestepping schemes. II. Linear analysis and applications to Newtonian and Langevin dynamics. *J. Chem. Phys.*, 109(5):1633–1642, 1998.

[5] E. Barth and T. Schlick. Overcoming stability limitations in biomolecular dynamics. I. Combining force splitting via extrapolation with Langevin dynamics in LN. *J. Chem. Phys.*, 109(5):1617–1632, 1998.

[6] P. F. Batcho, D. A. Case, and T. Schlick. Optimized particle-mesh Ewald/multiple-time step integration for molecular dynamics simulations. *J. Chem. Phys.*, 115(9):4003–4018, 2001.

[7] P. F. Batcho and T. Schlick. New splitting formulations for lattice summations. *J. Chem. Phys.*, 115(18):8312–8326, 2001.

[8] B. J. Berne and J. E. Straub. Novel methods of sampling phase space in the simulation of biological systems. *Curr. Topics in Struct. Biol.*, 7:181–189, 1997.

[9] G. Besold, I. Vattulainen, M. Kartunnen, and J. M. Polson. Towards better integrators for dissipative particle dynamics simulations. *Phys. Rev. E*, 62(6):R7611–R7614, 2000.

[10] T. Bishop, R. D. Skeel, and K. Schulten. Difficulties with multiple timestepping and the fast multipole algorithm in molecular dynamics. *J. Comp. Chem.*, 18(14):1785–1791, 1997.

[11] C. L. Brooks III, M. Karplus, and B. M. Pettitt. *Proteins: A Theoretical Perspective of Dynamics, Structure and Thermodynamics*, volume LXXI of *Advances in Chemical Physics*. John Wiley & Sons, New York, 1988.

[12] A. Brünger, C. B. Brooks, and M. Karplus. Stochastic boundary conditions for molecular dynamics simulations of ST2 water. *Chem. Phys. Lett.*, 105:495–500, 1982.

[13] A. Crisanti, M. Falcioni, and A. Vulpiani. On the effects of an uncertainty on the evolution law in dynamical systems. *Physica A*, 160:482–502, 1989.

[14] T. A. Darden, D. M. York, and L. G. Pedersen. Particle mesh Ewald. An $N \cdot \log(N)$ method for Ewald sums in large systems. *J. Chem. Phys.*, 98:10089–10092, 1993.
[15] Y. Duan and P. Kollman. Pathways to a protein folding intermediate observed in a 1 microsecond simulation in aqueous solution. *Science*, 282:740–744, 1998.
[16] H. Dufner, S. M. Kast, J. Brickmann, and M. Schlenkrich. Ewald summation versus direct summation of shifted-force potentials for the calculation of electrostatic interactions in solids: A quantitative study. *J. Comp. Chem.*, 18:660–676, 1997.
[17] P. Español and P. B. Warren. Statistical mechanics of dissipative particle dynamics. *Europhys. Lett.*, 30(4):191–196, 1995.
[18] P. Ewald. Die Berechnung optischer und elektrostatischer Gitterpotentiale. *Ann. Phys.*, 64:253–287, 1921.
[19] D. Fincham. Optimisation of the Ewald sum for large systems. *Mol. Sim.*, 13:1–9, 1994.
[20] T. Forester and W. Smith. On multiple time-step algorithms and the Ewald sum. *Mol. Sim.*, 13(3):195–204, 1994.
[21] B. García-Archilla, J. M. Sanz-Serna, and R. D. Skeel. Long-time-step methods for oscillatory differential equations. *SIAM J. Sci. Comput.*, 20(3):930–963, 1998.
[22] B. García-Archilla, J. M. Sanz-Serna, and R. D. Skeel. The mollified impulse method for oscillatory differential equations. In D. F. Griffiths and G. A. Watson, editors, *Numerical Analysis 1997*, pages 111–123, London, 1998. Pitman.
[23] R. D. Groot and P. B. Warren. Dissipative particle dynamics: Bridging the gap between atomistic and mesoscopic simulation. *J. Chem. Phys.*, 107(11):4423–4435, 1997.
[24] H. Grubmüller. Dynamiksimulation sehr großer Makromoleküle auf einem Parallelrechner. Master's thesis, Physik-Dept. der Tech. Univ. München, Munich, 1989.
[25] H. Grubmüller, H. Heller, A. Windemuth, and K. Schulten. Generalized Verlet algorithm for efficient molecular dynamics simulations with long-range interactions. *Mol. Sim.*, 6:121–142, 1991.
[26] A. Grzybowski, E. Gwozdz, and A. Brodka. Ewald summation of electrostatic interactions in molecular dynamics of a three-dimensional system with periodicity in two directions. *Phys. Rev.*, 61:6706–6712, 2000.
[27] E. Hairer and C. Lubich. Asymptotic expansions and backward analysis for numerical integrators. In *Dynamics of Algorithms*, pages 91–106, New York, 2000. IMA Vol. Math. Appl 118, Springer-Verlag.
[28] E. Hairer and C. Lubich. Long-time energy conservation of numerical methods for oscillatory differential equations. *SIAM J. Numer. Anal.*, 38(2):414–441, 2000.
[29] D. D. Humphreys, R. A. Friesner, and B. J. Berne. A multiple-time-step molecular dynamics algorithm for macromolecules. *J. Phys. Chem.*, 98(27):6885–6892, 1994.
[30] J. A. Izaguirre. *Longer Time Steps for Molecular Dynamics*. PhD thesis, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA, 1999.
[31] J. A. Izaguirre. Generalized mollified multiple time stepping methods for molecular dynamics. In A. Brandt, J. Bernholc, and K. Binder, editors, *Multiscale Computational Methods in Chemistry and Physics*, volume 177 of *NATO Science Series: Series III Computer and Systems Sciences*, pages 34–47. IOS Press, Amsterdam, Netherlands, 2001.
[32] J. A. Izaguirre, D. P. Catarello, J. M. Wozniak, and R. D. Skeel. Langevin stabilization of molecular dynamics. *J. Chem. Phys.*, 114(5):2090–2098, 2001.
[33] J. A. Izaguirre, Q. Ma, T. Matthey, J. Willcock, T. Slabach, B. Moore, and G. Viamontes. Overcoming instabilities in Verlet-I/r-RESPA with the mollified impulse method. In T. Schlick and H. H. Gan, editors, *Proceedings of 3rd International Workshop on Methods for Macromolecular Modeling*, volume 24 of *Lecture Notes in Computational Science and Engineering*, pages 146–174. Springer-Verlag, Berlin, New York, 2002.
[34] J. A. Izaguirre, S. Reich, and R. D. Skeel. Longer time steps for molecular dynamics. *J. Chem. Phys.*, 110(19):9853–9864, 1999.
[35] W. L. Jorgensen, J. Chandrasekhar, J. D. Madura, R. W. Impey, and M. L. Klein. Comparison of simple potential functions for simulating liquid water. *J. Chem. Phys.*, 79:926–935, 1983.
[36] L. Kalé, R. Skeel, M. Bhandarkar, R. Brunner, A. Gursoy, N. Krawetz, J. Phillips, A. Shinozaki, K. Varadarajan, and K. Schulten. NAMD2: Greater scalability for parallel molecular dynamics. *J. Comp. Phys.*, 151:283–312, 1999.
[37] C. Kane, J. E. Marsden, and M. Ortiz. Symplectic energy-momentum integrators. *J. Math. Phys.*, 40:3353–3371, 1999.
[38] K. Kholmurodov, W. Smith, K. Yasuoka, T. Darden, and T. Ebisuzaki. A smooth-particle mesh ewald method for DL_POLY molecular dynamics simulation package on the Fujitsu VPP700. *J. Comp. Chem.*, 21(13):1187–1191, 2000.
[39] A. Ko. MDSimAid: An automatic recommender for optimization of fast electrostatic algorithms for molecular simulations. Master's thesis, University of Notre Dame, Notre Dame, Indiana, USA, 2002.
[40] J. M. V. A. Koelman and P. J. Hoogerbrugge. Dynamic simulations of hard-sphere suspensions under steady shear. *Europhys. Lett.*, 21:363, 1993.

[41] A. R. Leach. *Molecular Modelling: Principles and Applications.* Addison-Wesley Longman, Reading, Massachusetts, 1996.
[42] B. Leimkuhler and S. Reich. A reversible averaging integrator for multiple time-scale dynamics. *J. Comput. Phys.*, 171:95–114, 2001.
[43] M. Levitt. Molecular dynamics of native protein. *J. Mol. Biol.*, 168:595–620, 1983.
[44] M. Levitt. Protein folding by restrained energy minimization and molecular dynamics. *J. Mol. Biol.*, 170:723–764, 1983.
[45] M. Levitt. Molecular dynamics of macromolecules in water. *Chem. Scr.*, 29A:197–203, 1989.
[46] M. Levitt and A. Warshel. Computer simulation of protein folding. *Nature*, 253:694–698, 1975.
[47] A. Lew, J. E. Marsden, M. Ortiz, and M. West. Asynchronous variational integrators. to be published.
[48] T. R. Littell, R. D. Skeel, and M. Zhang. Error analysis of symplectic multiple time stepping. *SIAM J. Numer. Anal.*, 34(5):1792–1807, 1997.
[49] M. López-Marcos, J. M. Sanz-Serna, and R. D. Skeel. Explicit symplectic integrators using Hessian–vector products. *SIAM J. Sci. Comput.*, 18:223–238, 1997.
[50] Q. Ma. *Novel Multiscale Algorithms for Molecular Dynamics*. PhD thesis, University of Notre Dame, Notre Dame, Indiana, USA, 2003.
[51] Q. Ma and J. A. Izaguirre. Long time step molecular dynamics using targeted Langevin stabilization. In *Proceedings of the ACM Symposium on Applied Computing*, pages 178–182, Melbourne, Florida, March 2003. ACM Press.
[52] Q. Ma, J. A. Izaguirre, and R. D. Skeel. Verlet-I/r-RESPA/Impulse is limited by nonlinear instability. *SIAM J. on Sci. Comput.*, 24(6):1951–1973, 2003.
[53] 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, I. W. E. Reiher, B. Roux, M. Schlenkrich, J. Smith, R. Stote, J. Straub, M. Watanabe, J. Wiorkiewicz-Kuczera, D. Yin, and M. Karplus. All-hydrogen empirical potential for molecular modeling and dynamics studies of proteins using the CHARMM22 force field. *J. Phys. Chem. B*, 102:3586–3616, 1998.
[54] 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, B. Roux, M. Schlenkrich, J. Smith, R. Stote, J. Straub, M. Watanabe, J. Wiorkiewicz-Kuczera, D. Yin, and M. Karplus. Self-consistent parameterization of biomolecules for molecular modeling and condensed phase simulations. *FASEB J.*, page 6:A143, 1992.
[55] T. Matthey. *Framework Design, Parallelization and Force Computation in Molecular Dynamics.* PhD thesis, University of Bergen, Bergen, Norway, 2002.
[56] T. Matthey, T. Cickovski, S. Hampton, A. Ko, Q. Ma, T. Slabach, and J. A. Izaguirre. PROTOMOL: an object-oriented framework for prototyping novel algorithms for molecular dynamics. Submitted to ACM Trans. Math. Softw., 2003.
[57] T. Matthey, A. Ko, and J. A. Izaguirre. ProtoMol, an object-oriented framework for algorithmic development. In D. Abramson, A. Bogdanov, J. J. Dongarra, and P. M. Sloot, editors, *Computational Science- ICCS 2003, International conference Melbourne, Australia and St. Petersburg, Russia*, volume 2659 of *Lecture Notes in Computer Science*, pages 50–59. Springer-Verlag, 2003.
[58] I. Pagonabarraga and D. Frenkel. Dissipative particle dynamics for interacting systems. *J. Chem. Phys.*, 115(11):5015–5026, 2001.
[59] I. Pagonabarraga, M. H. J. Hagen, and D. Frenkel. Self-consistent dissipative particle dynamics algorithm. *Europhys. Lett.*, 42(4):377–382, 1998.
[60] J. W. Perram, H. G. Petersen, and S. W. de Leeuw. An algorithm for the simulation of condensed matter which grows as the $\frac{3}{2}$ power of the number of particles. *Mol. Phys.*, 65:875–893, 1988.
[61] P. Procacci and M. Marchi. Taming the Ewald sum in molecular dynamics simulations of solvated proteins via a multiple time step algorithm. *J. Chem. Phys.*, 104(8):3003–3012, 1996.
[62] P. Procacci, M. Marchi, and G. J. Martyna. Electrostatic calculations and multiple time scales in molecular dynamics simulation of flexible molecular systems. *J. Chem. Phys.*, 108(21):8799–8803, 1998.
[63] C. Sagui and T. Darden. Multigrid methods for classical molecular dynamics simulations of biomolecules. *J. Chem. Phys.*, 114(15):6578–6591, 2001.
[64] A. Sandu and T. Schlick. Masking resonance artifacts in force-splitting methods for biomolecular simulations by extrapolative Langevin dynamics. *J. Comput. Phys*, 151(1):74–113, 1999.
[65] J. M. Sanz-Serna and M. P. Calvo. *Numerical Hamiltonian Problems*. Chapman and Hall, London, 1994.
[66] M. R. Shirts and V. S. Pande. Mathematical analysis of coupled parallel simulations. *Phys. Rev. Lett.*, 86(22):4983–4987, 2001.
[67] R. D. Skeel. Integration schemes for molecular dynamics and related applications. In M. Ainsworth, J. Levesley, and M. Marletta, editors, *The Graduate Student's Guide to Numerical Analysis*, SSCM, pages 119–176. Springer-Verlag, Berlin, 1999.
[68] R. D. Skeel. Why does molecular dynamics work. Presented in CIMMS-IPAM Workshop – Molecular

Modeling and Computation: Perspectives and Challenges, November 2002.

[69] R. D. Skeel and J. J. Biesiadecki. Symplectic integration with variable stepsize. *Annals of Numer. Math.*, 1:191–198, 1994.

[70] R. D. Skeel and D. J. Hardy. Practical construction of modified Hamiltonians. *SIAM J. on Sci. Comput.*, 23(4):1172–1188, 2001.

[71] R. D. Skeel and J. A. Izaguirre. The five femtosecond time step barrier. In P. Deuflhard, J. Hermans, B. Leimkuhler, A. Mark, S. Reich, and R. D. Skeel, editors, *Computational Molecular Dynamics: Challenges, Methods, Ideas*, volume 4 of *Lecture Notes in Computational Science and Engineering*, pages 303–318. Springer-Verlag, Berlin Heidelberg New York, 1998.

[72] R. D. Skeel and J. A. Izaguirre. An impulse integrator for Langevin dynamics. *Mol. Phys.*, 100(24):3885–3891, 2002.

[73] R. D. Skeel, I. Tezcan, and D. J. Hardy. Multiple grid methods for classical molecular dynamics. *J. Comp. Chem.*, 23(6):673–684, 2002.

[74] R. D. Skeel, G. Zhang, and T. Schlick. A family of symplectic integrators: stability, accuracy, and molecular dynamics applications. *SIAM J. Sci. Comput.*, 18(1):203–222, 1997.

[75] M. Tuckerman, B. J. Berne, and G. J. Martyna. Reversible multiple time scale molecular dynamics. *J. Chem. Phys.*, 97(3):1990–2001, 1992.

[76] W. Weber, P. H. Hünenberger, and J. A. McCammon. Molecular dynamics simulations of a polyalanine octapeptide under Ewald boundary conditions: Influence of artificial periodicity on peptide conformation. *J. Phys. Chem. B*, 104:3668–3675, 2000.

[77] S. Wolfram. *A new kind of science*. Wolfram Media, Inc., Champaign, IL, 2002.

[78] R. Zhou and B. J. Berne. A new molecular dynamics method combining the reference system propagator algorithm with a fast multipole method for simulating proteins and other complex systems. *J. Chem. Phys.*, 103(21):9444–9459, 1995.

[79] R. Zhou, E. Harder, H. Xu, and B. J. Berne. Efficient multiple time step method for use with Ewald and Particle Mesh Ewald for large biomolecular systems. *J. Chem. Phys.*, 115(5):2348–2358, August 1 2001.