

Long time step molecular dynamics using targeted Langevin stabilization*

Qun Ma[†] Jesús A. Izaguirre[‡]

September 13, 2002

*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.

[†]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

Long time step molecular dynamics using targeted Langevin stabilization

Abstract

We introduce the B-spline Mollified Impulse (MOLLY) and the Targeted MOLLY (TM) for molecular dynamics (MD). TM uses targeted Langevin coupling to stabilize B-spline MOLLY. Results show that with a proper choice of parameters, the self-diffusion coefficient can be correctly estimated using TM with outer time step of 16 fs. The energy and temperature are bounded during those simulations, and a six-fold speedup is obtained as compared to leapfrog with step size of 1 fs. The radial distribution functions computed from the simulations using TM are very similar to those obtained from the simulations using leapfrog.

Key words. Molecular dynamics, long molecular dynamics simulations, multiple-time-stepping (MTS), Verlet-I/r-RESPA/Impulse, mollified impulse method, Targeted Langevin stabilization.

1 Introduction

Molecular dynamics (MD) is the most widely used technique for biomolecular simulations. Long trajectories are computed by solving Newton's equations of motion, *i.e.*, the system of ODEs given by

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

where \mathbf{q} is the position vector, \mathbf{p} is the momentum vector, $U(\mathbf{q})$ is the potential energy, $-U'(\mathbf{q})$ is the force, and \mathbf{M} is the mass matrix.

In an attempt to bridge the time scale gap between simulations and the phenomena of interest, multiple time stepping (MTS) integrators have been introduced and have been an area of active research for more than a decade [1–10]. The prototypical algorithm is the Verlet-I [9, 10]/r-RESPA [8] integrator, which splits the forces into fast and slow components, F^{fast} and F^{slow} , and evaluates the former more frequently than the latter.

It has been found that the step size of Verlet-I/r-RESPA/Impulse is limited by both linear [11–15] and nonlinear instabilities [1] at half, one third and possibly one fourth of the periods of the fastest normal modes (referred to linear instability, 3:1 and 4:1 nonlinear instabilities, respectively). For the SHAKE-constrained simulations of explicitly solvated proteins, the 3:1 and 4:1 nonlinear resonances associated with the fastest remaining modes limit the step size of Verlet-I/r-RESPA/Impulse to 4.0 fs [1].

A promising approach to achieve longer time steps while preserving dynamics is to add a mild Langevin coupling, examples including LN [16] and LM [4]. LN is successful in stabilizing the simulation, but it destroys the dynamics; LM is both successful in stabilizing the simulation and preserving the dynamics. It allows outer time step Δt to be 12 fs when estimating the self-diffusion coefficient of waters (a 300% asymptotic speedup) [4]. The damping coefficient has to be very small though, which limits the step size. Dissipative particle dynamics (DPD), a method similar to MD, is another successful example in lengthening time steps [17, 18]. It uses Langevin coupling that is momentum-preserving.

We present a new MTS integrator, the Targeted Mollified Impulse method (Targeted MOLLY,

or TM), for MD simulations. TM uses MOLLY [2–6] as the outer-most integrator, and the self-consistent dissipative leapfrog [17], a method commonly used in Dissipative Particle Dynamics, as the inner-most integrator in which a random force and a dissipative force for interacting pairs of atoms are selectively introduced. We present an evaluation of a two-level TM method on a model system of TIP3P [19] waters with flexible bonds and angles. Results show that with a proper choice of parameters, the self-diffusion coefficient can be correctly estimated using TM with outer time step of 16 fs which results in a six-fold speedup as compared to leapfrog with step size of 1 fs. As an additional benefit, because the time step is larger, this new integrator is more scalable than less stable integrators for parallel MD simulations.

2 Targeted MOLLY Integrator

2.1 Self-Consistent Dissipative Leapfrog Integrator

A commonly used integrator for DPD is the the self-consistent dissipative leapfrog, which 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. We use the procedure outlined in [17], but using atoms in place of DPD particles. The dissipative and random forces are given below:

$$\mathbf{F}^D(\mathbf{r}_{ij}, \mathbf{v}_{ij}) = -\gamma\omega^D(\mathbf{r}_{ij})(\mathbf{v}_{ij} \cdot \hat{\mathbf{r}}_{ij})\hat{\mathbf{r}}_{ij}, \quad (2)$$

$$\mathbf{F}^R(\mathbf{r}_{ij}) = \sigma\omega^R(\mathbf{r}_{ij})\xi_{ij}\delta t^{-1/2}\hat{\mathbf{r}}_{ij}, \quad (3)$$

where we have used the notation $\mathbf{r}_{ij} \equiv \mathbf{r}_j - \mathbf{r}_i$ and $\mathbf{v}_{ij} \equiv \mathbf{v}_j - \mathbf{v}_i$, and $\hat{\mathbf{r}}_{ij}$ denotes a unit vector in the direction of \mathbf{r}_{ij} ; r_i and v_i are the position and velocity of atom i , respectively. 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, $\hat{\mathbf{r}}_{ij}$, the Gaussian distributed random variable with zero mean and unit variance, and the time step, δt . The reason for the appearance of the term $\delta t^{-1/2}$ in Eq. (3) was shown in [20, Section II]. As a result, $\langle f_i^2 \rangle \equiv \langle (\mathbf{F}_i^R / \sqrt{\delta t})^2 \rangle = \sigma^2 / \delta t$, *i.e.*, the spread of the random force becomes wider as the time step becomes smaller, which holds true if we interpret the random force as a Wiener process [21, p. 28].

The weight functions, ω^D and ω^R in this protocol are both set to 1, which automatically satisfies $\omega^R = \sqrt{\omega^D}$ and thus ensures that the probability to observe a particular configuration of the targeted molecular dynamics particles is given by the Boltzmann distribution in equilibrium. Also γ and σ satisfy $k_B T = \sigma^2 / (2\gamma)$ where k_B is the Boltzmann constant and T the equilibrium temperature. 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 [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, $g(r)$, and thus does not produce the correct equilibrium properties [17]. The self-consistent scheme conserves the equilibrium temperature better. More importantly, the self-consistent leapfrog is time reversible and momentum-preserving.

2.2 B-spline Mollified Impulse (MOLLY)

Researchers have developed the mollified impulse method (MOLLY) [2–6] to overcome the linear instability of Verlet-I/r-RESPA/Impulse. The longest step size can be lengthened by 50% using MOLLY [5, 6]. 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, $U^{\text{slow}}(q)$ becomes $U^{\text{slow}}(\mathcal{A}(q))$. The force is defined as a gradient of this averaged potential,

$$-\nabla U^{\text{slow}}(q) \text{ is replaced by } -\mathcal{A}_q(q)^T \nabla U^{\text{slow}}(\mathcal{A}(q)), \quad (4)$$

where $\mathcal{A}_q(q)$ is a sparse Jacobian matrix. This perturbation compensates for finite Δt artifacts. Perturbing the potential rather than the force ensures that the numerical integrator remains symplectic [22].

The MOLLY averaging can be done by numerically integrating an auxiliary, reduced problem:

$$\mathcal{A}(q) = \frac{1}{\Delta t} \int_0^\infty \phi\left(\frac{t}{\Delta t}\right) \tilde{q}(t) dt \quad (5)$$

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

$$M \frac{d^2}{dt^2} \tilde{q} = F^{\text{reduced}}(\tilde{q}), \quad \tilde{q}(0) = q, \quad \frac{d}{dt} \tilde{q}(0) = 0. \quad (6)$$

Different averaging functions lead to MOLLY integrators with different stability and accuracy properties. The paper, Ref. [2], suggests using B-spline weight functions, which are non-zero over a short interval. B-spline weight functions have compact support in time and

thus make the method computationally feasible. One such B-spline weight function, stable at half of the fastest period, is called LONGAVERAGE:

$$\phi(s) = \begin{cases} 0, & s < 0 \quad \text{or} \quad s > 1, \\ 1, & 0 \leq s < 1, \\ \frac{1}{2}, & s = 1. \end{cases} \quad (7)$$

B-spline MOLLY has been implemented in PROTOMOL, an experimental component-based framework for MD simulations [23,24]. PROTOMOL’s modular design allows for easy prototyping of complex methods. It is freely available at <http://www.nd.edu/~lcls/protomol>.

It is straightforward to compute the $\mathcal{A}(q)$. Then $\mathcal{A}_q(q)$ can be computed using the chain rule with respect to each of the components of x . The time step for MOLLY averaging, δt , should be chosen such that the leapfrog method is stable. In the averaging, $F_q = -U_{qq}^{\text{reduced}}(q)$ can be computed efficiently using analytical Hessians for bonds and angles [25]. 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 the bonded hydrogens). The Hessian-vector product can also be performed efficiently using methods like those in Ref. [26].

2.3 Targeted MOLLY Integrator

Targeted MOLLY (TM) uses the self-consistent dissipative leapfrog as the innermost integrator and B-spline MOLLY as the outermost integrator. MOLLY’s enhanced stability and the momentum-preserving Langevin coupling allows computing the dynamics accurately using TM with Δt larger than previous methods, as confirmed by the numerical results in Section 3.

The TM discretization scheme is shown in Algorithm 1.

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

$$p_{n-1}^+ = p_{n-1} + \frac{\Delta t}{2} F^{\text{slow}}(\bar{q}_{n-1}), \quad (8)$$

oscillate: Propagate q_{n-1} and p_{n-1}^+ by integrating

$$\dot{q} = M^{-1}p, \quad \dot{p} = F^{\text{fast}}(q) + F^{\text{R}}(q) + F^{\text{D}}(q) \quad (9)$$

for an interval Δt to get q_n and p_n^- using the the self-consistent dissipative leapfrog scheme. F^{R} and F^{D} , the random and dissipative forces, respectively, are defined in Eqs. (3) and (2).

a time averaging: Calculate a temporary vector of time-averaged positions \bar{q}_n and a Jacobian matrix, $J_n = \nabla_q \bar{q}_n$. The time averaging function uses only the fastest forces $F^{\text{reduced}}(q)$.

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

$$p_n = p_n^- + \frac{\Delta t}{2} F^{\text{slow}}(\bar{q}_n). \quad (10)$$

Algorithm 1: Targeted MOLLY discretization.

In traditional Langevin integrators the motion of every atom is damped individually, which makes them not momentum preserving. In TM, we use *targeted Langevin interactions*, i.e., Langevin forces are applied to particular pairs of atoms that have interactions which we know would cause instability. Those pairs include covalently bonded pairs (bonds and angles) and possibly hydrogen-bonded pairs. 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 Results and Discussions

3.1 Model, protocol and simulation programs

We perform simulations of a 10 Å radius sphere flexible TIP3P water system which has 423 atoms. The system was equilibrated using NAMD 2.3 during 100 ps of simulation time by minimization followed by temperature re-scaling to 300 K with spherical boundary conditions. The system was then equilibrated by 250 ps simulation using PROTOMOL with periodic boundary conditions. By equilibrating we avoid highly improbable values of different contributions to energies. Then we ran simulations using PROTOMOL. Results discussed in this section can be reproduced using PROTOMOL. These simulations used periodic boundary conditions and Ewald sum for computing full electrostatic interactions [27, 28].

We use a 2-level TM integrator for the simulations with outer time step of 16 fs and inner time step of 2 fs. In the outermost integrator, we use bonds and angles in the B-spline averaging, and LONGAVERAGE as averaging function. In the innermost integrator, we use pairwise Langevin interactions including both bonds and angles. The MOLLY averaging step size is 2 fs. The Langevin coupling coefficient, γ , is 4.0 amu/ps. The number of iterations

for self-consistency for all simulations is 2. Slow forces include only the k-space term of the Ewald sum of Coulomb forces. Fast forces include bonded forces plus the real space term of the Ewald sum with corrections. Lennard Jones forces use a C^2 switching function. The Ewald accuracy is 10^{-12} . The *switchon* value is 4 Å and *cutoff* value is 6.5 Å.

3.2 Correct Dynamics using Large Time Steps

We summarize the self-diffusion coefficients, CPU time and speedup from different simulations in Table 1. All simulations were performed on a single node of the Hydra cluster (SUN OS 2.7, 32 nodes each of which is 440 MHz SunSparc 10 with a 256 MB memory). 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 (1) we can see that the self-diffusion coefficient is computed correctly using the TM method with the outer time step of 16 fs as compared with the result using leapfrog with time step of 1 fs. We also see a substantial speedup using the TM method: for TM at $\Delta t = 16$ fs we achieved a six-fold computational speedup over leapfrog with time step of 1 fs and a two-fold speedup over Verlet-I/r-RESPA/Impulse with $\Delta t = 4$ fs, $\delta t = 1$ fs. Note that LM (Langevin-Impulse plus Equilibrium MOLLY) only computes accurate dynamics (self-diffusion coefficient) with $\Delta t = 12$ fs [4].

In those simulations from which we compute the self-diffusion coefficient using TM method, we observe that the molecular temperature and the total energy are bounded. The relative drift of molecular temperature, measured as $\Delta T_{mol}/\bar{T}_{mol}$ where ΔT_{mol} is the total drift of molec-

Table 1: Self-diffusion coefficient (denoted by D), CPU time (denoted by t) and speed up (denoted by η) for 400 ps MD simulation of 141 flexible TIP3P water molecules with periodic boundary conditions. The error bar is given by twice of the standard deviation.

Integrator	Δt	δt	γ	D	t	η
Type	(fs)	(fs)	(amu/ps)	($10^{-5}\text{cm}^2/\text{s}$)	(hr)	-
Leapfrog	-	1	-	3.69 ± 0.01	68.28	1.00
Verlet-I/r-RESPA/Impulse	4	1	-	-	23.50	2.91
TM	16	2	4.0	3.68 ± 0.01	11.40	5.99

ular temperature and \bar{T}_{mol} is the average molecular temperature, is $-1.58 \pm 3.78\%$ with an average of 297.30 K (note the prescribed temperature is 300 K). The relative drift of total energy, measured as $\Delta E_{tot}/\bar{E}_{kin}$ where ΔE_{tot} is the drift of total energy and \bar{E}_{kin} is the average of kinetic energy [1], is $-7.01 \pm 8.79\%$. The total energy exhibits large fluctuations because of the very large step sizes.

3.3 Sampling Properties

Sampling from a well-defined ensemble is desirable feature of any simulation method, especially for measuring thermodynamic and transport properties. We show evidence that the TM method generates equilibrium properties correctly by showing the agreement between TM and leapfrog on the radial distribution function, $g(r)$, computed from the simulations, see Fig. 1.

We should note that the TM method is not a substitute for Hamiltonian integrators such

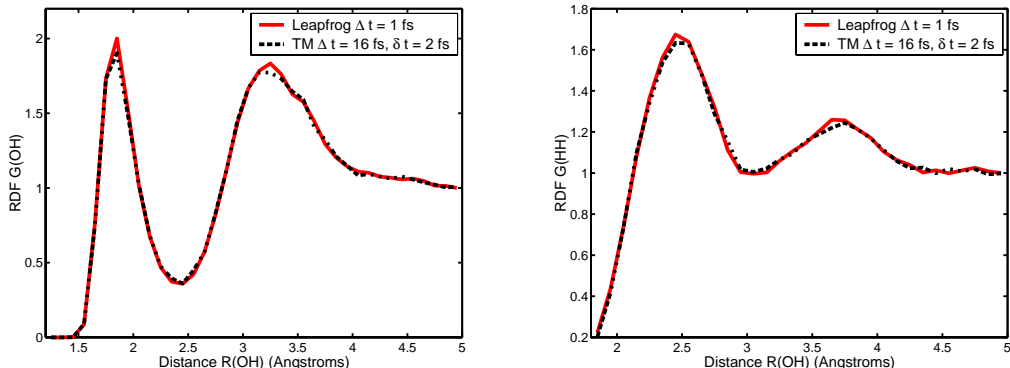


Figure 1: Radial distribution function for the O-H and H-H interactions for leapfrog and Targeted MOLLY (TM). It is seen that the structural property of the system is very well conserved.

as leapfrog, giving the fact that the random and dissipative forces alters the “nature” of the dynamics. Whether or not TM can be used for rigorous study of other thermodynamic properties remains an open question because of the lack of a rigorous proof that it samples from the canonical ensemble. However, the evidence suggests that we are sampling from a canonical ensemble: self-consistent dissipative leapfrog has been proven to sample from the canonical ensemble in the limit of infinitesimal time step [29] as has Verlet-I/r-RESPA/Impulse. MOLLY uses a potential that is just a slight perturbation of that used by Verlet-I/r-RESPA/Impulse. It is consistent with it in the limit of infinitesimal time step.

3.4 Computational Overhead of Mollification

The computational overhead of mollification is estimated in the range of 1.5 – 5% of the total execution time. 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.

3.5 Increasing Stability

While the fast and slow forces splitting of Ewald sum used in these simulations is easy to implement [30], it is not the most appropriate choice because the slow part also consists of some fast force contributions. A more appropriate splitting in Ewald sum has been suggested [31–33]. When using a better, distance-based splitting rather than a real- /reciprocal-space splitting, researchers have been able to use Δt that is 100% larger [33, Section IV]. These results suggest that using a more appropriate splitting, the TM method should be stable for even larger time steps.

Acknowledgments

Some of the calculations were performed at the Norwegian Super-computing Facilities through a Norwegian Research Council grant. Others are performed at the Hydra cluster owned by the College of Engineering of the University of Notre Dame and the Bob Linux cluster owned by the College of Science and College of Engineering of the University of Notre Dame. Thierry Matthey helped run 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 and Dr. R. Zhou for valuable comments.

References

- [1] Q. Ma, J. A. Izaguirre, and R. D. Skeel, submitted to *SIAM Journal on Scientific Computing*. Preprint at <http://www.nd.edu/~qma1/pub/MaIS0x.pdf> (unpublished).
- [2] B. García-Archilla, J. M. Sanz-Serna, and R. D. Skeel, *SIAM J. Sci. Comput.* **20**, 930 (1998).
- [3] B. García-Archilla, J. M. Sanz-Serna, and R. D. Skeel, in *Numerical Analysis 1997*, edited by D. F. Griffiths and G. A. Watson (Pitman, London, 1998), pp. 111–123.
- [4] J. A. Izaguirre, D. P. Catarello, J. M. Wozniak, and R. D. Skeel, *J. Chem. Phys.* **114**, 2090 (2001).
- [5] J. A. Izaguirre *et al.*, in *Proceedings of 3rd International Workshop on Methods for Macromolecular Modeling*, Vol. 24 of *Lecture Notes in Computational Science and Engineering*, edited by T. Schlick and H. H. Gan (Springer-Verlag, Berlin, New York, 2002), pp. 146–174, also available at <http://www.nd.edu/~izaguirr/papers/newM3paper.pdf>.
- [6] J. A. Izaguirre, S. Reich, and R. D. Skeel, *J. Chem. Phys.* **110**, 9853 (1999).
- [7] J. J. Biesiadecki and R. D. Skeel, *J. Comput. Phys.* **109**, 318 (1993).
- [8] M. Tuckerman, B. J. Berne, and G. J. Martyna, *J. Chem. Phys.* **97**, 1990 (1992).
- [9] H. Grubmüller, Master's thesis, Physik-Dept. der Tech. Univ. München, Munich, 1989.

- [10] H. Grubmüller, H. Heller, A. Windemuth, and K. Schulten, *Molecular Simulation* **6**, 121 (1991).
- [11] T. Bishop, R. D. Skeel, and K. Schulten, *J. Comp. Chem.* **18**, 1785 (1997).
- [12] A. Sandu and T. Schlick, *J. Comput. Phys* **151**, 74 (1999).
- [13] E. Barth and T. Schlick, *J. Chem. Phys.* **109**, 1633 (1998).
- [14] R. D. Skeel and J. A. Izaguirre, in *Computational Molecular Dynamics: Challenges, Methods, Ideas*, Vol. 4 of *Lecture Notes in Computational Science and Engineering*, edited by P. Deuffhard *et al.* (Springer-Verlag, Berlin Heidelberg New York, 1998), pp. 303–318.
- [15] R. D. Skeel, G. Zhang, and T. Schlick, *SIAM J. Sci. Comput.* **18**, 203 (1997).
- [16] E. Barth and T. Schlick, *J. Chem. Phys.* **109**, 1617 (1998).
- [17] G. Besold, I. Vattulainen, M. Kartunnen, and J. M. Polson, *Physical Review E* **62**, R7611 (2000).
- [18] I. Pagonabarraga, M. Hagen, and D. Frenkel, *Europhysics Letters* **42**, 377 (1998).
- [19] W. L. Jorgensen *et al.*, *J. Chem. Phys.* **79**, 926 (1983).
- [20] R. D. Groot and P. B. Warren, *J. Chem. Phys.* **107**, 4423 (1997).

- [21] P. E. Kloeden and E. Platen, *Numerical Solution of Stochastic Differential Equations*, Vol. 23 of *Applications of Mathematics: Stochastic Modelling and Applied Probability* (Springer-Verlag, New York, 1992), second corrected printing 1995.
- [22] J. M. Sanz-Serna and M. P. Calvo, *Numerical Hamiltonian Problems* (Chapman and Hall, London, 1994).
- [23] T. Matthey and J. A. Izaguirre, in *Proc. of the Tenth SIAM Conf. on Parallel Processing for Scientific Computing (PP01), Proceedings in Applied Mathematics* (Society for Industrial and Applied Mathematics, Philadelphia, 2001).
- [24] T. Matthey, A. Ko, and J. A. Izaguirre, submitted to the 2003 International Conference on Software Engineering (unpublished).
- [25] Q. Ma and J. A. Izaguirre, submitted to *J. Chem. Phys.* (unpublished).
- [26] M. López-Marcos, J. M. Sanz-Serna, and R. D. Skeel, *SIAM J. Sci. Comput.* **18**, 223 (1997).
- [27] P. Procacci, M. Marchi, and G. J. Martyna, *J. Chem. Phys.* **108**, 8799 (1998).
- [28] P. Ewald, *Ann. Phys.* **64**, 253 (1921).
- [29] P. Español and P. B. Warren, *Europhysics Letters* **30**, 191 (1995).
- [30] P. Procacci and M. Marchi, *J. Chem. Phys.* **104**, 3003 (1996).
- [31] P. F. Batcho, D. A. Case, and T. Schlick, *J. Chem. Phys.* **115**, 4003 (2001).

[32] P. F. Batcho and T. Schlick, *J. Chem. Phys.* **115**, 8312 (2001).

[33] R. Zhou, E. Harder, H. Xu, and B. J. Berne, *J. Chem. Phys.* **115**, 2348 (2001).