JCTC Journal of Chemical Theory and Computation

Equipartition and the Calculation of Temperature in Biomolecular Simulations

Michael P. Eastwood, Kate A. Stafford,[§] Ross A. Lippert, Morten Ø. Jensen,
Paul Maragakis, Cristian Predescu, Ron O. Dror, and David E. Shaw*,[†]

D. E. Shaw Research, New York, New York 10036

Received June 5, 2009

Abstract: Since the behavior of biomolecules can be sensitive to temperature, the ability to accurately calculate and control the temperature in molecular dynamics (MD) simulations is important. Standard analysis of equilibrium MD simulations—even constant-energy simulations with negligible long-term energy drift—often yields different calculated temperatures for different motions, however, in apparent violation of the statistical mechanical principle of equipartition of energy. Although such analysis provides a valuable warning that other simulation artifacts may exist, it leaves the actual value of the temperature uncertain. We observe that Tolman's generalized equipartition theorem should hold for long stable simulations performed using velocity-Verlet or other symplectic integrators, because the simulated trajectory is thought to sample almost exactly from a continuous trajectory generated by a shadow Hamiltonian. From this we conclude that all motions should share a single simulation temperature, and we provide a new temperature estimator that we test numerically in simulations of a diatomic fluid and of a solvated protein. Apparent temperature variations between different motions observed using standard estimators do indeed disappear when using the new estimator. We use our estimator to better understand how thermostats and barostats can exacerbate integration errors. In particular, we find that with large (albeit widely used) time steps, the common practice of using two thermostats to remedy so-called hot solvent−cold solute problems can have the counter-intuitive effect of causing temperature imbalances. Our results, moreover, highlight the utility of multiple-time step integrators for accurate and efficient simulation.

1. Introduction

Fueled by algorithmic improvements and by the growth of computer power, molecular dynamics (MD) simulations are making increasingly important scientific contributions to biology. There is considerable interest in further accelerating simulations and improving their accuracy. Since most biomolecular simulations are of classical systems at equilibrium, one useful measure of accuracy is the extent to which the distribution of energy among different degrees of freedom is consistent with the equipartition theorem of statistical mechanics.[1] The most familiar consequence of equipartition is that each particle in an equilibrium system has an average kinetic energy of $k_BT/2$ (where T is the temperature of the system and k_B is the Boltzmann constant) arising from its motion in each spatial dimension. This result does not depend on details of the potential energy function (the "force field"). In biomolecular simulations, thermalization of kinetic energy typically occurs on a subnanosecond time scale,[2,3] so substantial deviations from equipartition in long simulations are likely symptoms of a problem with the simulation methodology. One symptom, whose presence is often tested for in practice, is a difference between the temperature of the solvent and solute, often referred to as a *hot solvent−cold solute* problem.[4] A hot solvent−cold solute problem could

* Corresponding author. E-mail: David.Shaw@DEShawResearch.com.

[†] David E. Shaw is also affiliated with the Center for Computational Biology and Bioinformatics, Columbia University, New York, New York 10032.

[§] Current address: Department of Biochemistry and Molecular Biophysics, Columbia University, New York, NY 10032.

10.1021/ct9002916 © 2010 American Chemical Society
Published on Web 05/24/2010

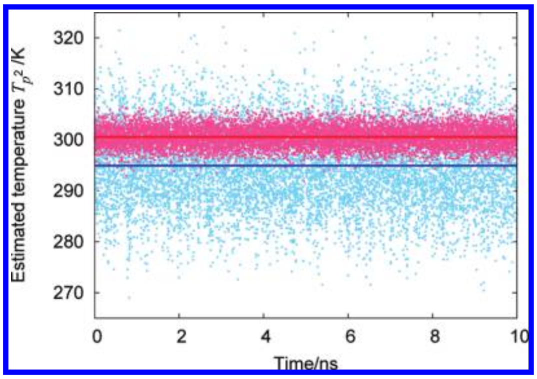

Figure 1. A hot solvent−cold protein problem. Temperatures of the protein ubiquitin (blue points) and water solvent (red points) are shown as a function of simulation time. Data were taken from an all-atom constant energy simulation that used velocity-Verlet integration with a 2 fs time step and bonds to hydrogen constrained; more details are given in Section 3.2. Temperatures were evaluated using the squares of the on-step velocity-Verlet momenta. The lines denote the average temperature values over the entire simulation.

have many potential causes, and testing for the presence of this symptom (and similar deviations from equipartition) has helped to diagnose underlying problems in barostats,[5] thermostats[6,7] and in approximate treatments of long-range electrostatic[8−10] and dispersive[11] interactions and has led to various methodological improvements.

Here, we investigate how truncation errors arising from the finite simulation time step δ_t affect equipartition and the calculation of temperature. We mainly focus on the widely used velocity-Verlet integrator,[12,13] but the basic theoretical finding applies to symplectic integrators in general. To help rule out the well-documented causes of a breakdown of equipartition noted above,[5−11] we initially focus on constant-energy simulations that are stable (that is, those that show little long-term energy drift). Fortunately, this is achievable with currently typical simulation parameters, so the results are directly relevant in practice. Although not as dramatic as hot solvent−cold solute problems arising from other origins, the effects of truncation errors can still be substantial. In Figure 1, for example, the temperature of a protein and the surrounding water molecules are shown as a function of simulation time. The temperature of the protein is seen to be lower than that of the water by about 6 K.

Such results are widely understood to expose real simulation artifacts originating in the finite integration time step, but it is unclear whether they reflect any actual temperature differences or even whether temperature has a precise definition for $\delta_t > 0$. In this paper, we show how the definition of temperature generalizes to $\delta_t > 0$ and show that, in examples like the one above, different motions do share a single temperature. Our reasoning is straightforward. The velocity-Verlet integrator is symplectic[14] and is, thus, thought to sample positions and momenta almost exactly from a trajectory generated by a modified Hamiltonian,[15] often called a *shadow Hamiltonian*. Since particle momenta do not enter this Hamiltonian quadratically, the equipartition relation is not applicable. We expect, however, that generalized equipartition,[1,16] which holds for a broad class of Hamiltonians, will be applicable to the shadow Hamiltonian. This implies the existence of a single well-defined simulation temperature for all motions that is given by the product of momentum and velocity and that is easily evaluated in practice.

To avoid potential confusion at the outset and to clarify why estimating temperature from the product of momentum p and velocity v (a "pv formula") is distinct from previous approaches, we emphasize the finding[17] that when $\delta_t > 0$, $v \neq p/m$, where m is mass; we also explain more carefully what we mean here by velocity and momentum. By momentum, we mean the canonical momentum that enters the shadow Hamiltonian. This is directly provided by the integrator. For velocity Verlet, the momentum is simply the usual on-step velocity-Verlet momentum. By velocity, we mean the *instantaneous* rate of change of position on the underlying trajectory generated by the shadow Hamiltonian. The integrator directly yields positions (and momenta) but not their time derivatives, and the velocity cannot be exactly expressed in terms of a finite number of positions and momenta. In particular, as noted above, $v \neq p/m$, even though p/m might commonly be called a velocity; for velocity Verlet, p/m (often called the on-step "velocity-Verlet velocity") differs from the velocity by $\mathcal{O}(\delta_t^2)$. Thus if only the on-step velocity-Verlet momenta are used to evaluate temperature (a "p^2 formula," as used in Figure 1), then temperatures will also be in error by order δ_t^2. Nevertheless, it is straightforward to construct more accurate velocity estimators.[18] One simple approach is a polynomial interpolation over positions sampled at different times; the velocity estimator appearing in Beeman's version of Verlet[13] is a well-known special case, and unsurprisingly it is possible to increase the accuracy further by interpolating over more positions. Perhaps counterintuitively, however, our pv formula shows that having obtained an accurate estimate of v, the temperature follows via the product pv (even though p/m may itself be a poor estimator of v) and *not* the square of the accurately estimated velocity (a "v^2 formula"), as appears to be typically assumed.[18,19] Indeed, as will become clearer below, if temperature is estimated using highly accurate velocities alone, a hot solvent−cold protein problem, like that shown in Figure 1, will simply be replaced by a cold solvent−hot protein problem of similar magnitude.

We test our theoretical conclusions numerically for two systems. First, we calculate the temperatures of vibrational and translational motion in a diatomic fluid as a function of δ_t. We find that p^2 and v^2 temperature estimates each yield substantially different values for the two motions, but the pv estimator shows that the temperatures of these motions are in fact identical within a very small statistical error. Thus although conventional equipartition (by which we mean the usual, as opposed to generalized, equipartition relation) breaks down, generalized equipartition holds, and a well-defined temperature exists. Using analytical estimates, we confirm that although deviations from conventional equipartition do not reflect temperature differences, they do reflect the real difference between the Hamiltonian and its shadow. Second, we perform all-atom MD simulations of ubiquitin in explicit solvent, examining the temperature of the different quasiharmonic protein motions and comparing the overall temperature of the protein to the solvent. Again, we find different motions to share a single temperature, even when conventional equipartition is not satisfied.

In addition to providing an accurate estimator for simulation temperature and using it to confirm that generalized equipartition is satisfied in stable simulations, we use it to investigate how integration errors can be exacerbated by use of a thermostat or barostat. This danger has been recently highlighted;[18] we discuss it in light of the new estimator and demonstrate some potential pitfalls. Notably, we use our estimator to show that unless the time step is chosen to be sufficiently small, use of multiple thermostats can lead to a breakdown of generalized equipartition with genuine temperature imbalances and with heat flow in the system. The commonly used remedy of hot solvent−cold solute problems in which one thermostat is applied to protein and another to the solvent, in an effort to maintain them at the same temperature, can thus potentially have a counterintuitive, and counterproductive, effect.

The root cause of all the simulation artifacts investigated in this paper is truncation error. Satisfying generalized equipartition by no means implies that the simulation is free from this source of artifacts; indeed, the breakdown of conventional equipartition signals their existence. Reducing the time step naturally reduces truncation error and brings the different temperature estimators into agreement, but due to the computational expense of MD, this solution is often unpalatable. One promising approach to reduce errors is to modify the integration scheme. Using the deviations from conventional equipartition as a criterion, we show, for example, that for our test systems, the reversible reference system propagation algorithm (r-RESPA) multiple-time step scheme[20] can achieve the benefit of a reduced velocity-Verlet time step at a fraction of the computational expense.

2. Theory

To generalize the definition of simulation temperature to $\delta_t > 0$, we make use of two established concepts: generalized equipartition and the shadow Hamiltonian, which we briefly review in Sections 2.1 and 2.2, respectively. In Section 2.3, we give our definition of simulation temperature for $\delta_t > 0$. Using the harmonic oscillator as an analytically tractable example, we quantify errors in some conventional estimates of simulation temperature in Section 2.4. Appendix A describes how to estimate temperatures of motions that involve multiple atoms, such as quasiharmonic motions in proteins. The effect of integration errors on simulation pressure is discussed in Appendix B.

2.1. Generalized Equipartition. Generalized equipartition (eqs 2 and 3) was derived by Tolman,[16] who considered the following canonical ensemble average for the Hamiltonian system $H(\mathbf{p},\mathbf{q})$:

$$\begin{aligned}\left\langle x_i \frac{\partial H}{\partial x_i}\right\rangle &= \frac{1}{Q}\int dx\left(\frac{-x_i}{\beta}\right)\frac{\partial e^{-\beta H}}{\partial x_i}\\ &= k_B T - \frac{1}{\beta Q}\int dx_1, ...,dx_{i-1}dx_{i+1}, ...,dx_{2N_f}[x_i e^{-\beta H}]_{x_i^{\min}}^{x_i^{\max}}\end{aligned} \tag{1}$$

We use x_i to label an element of either position or momentum. N_f is the number of positional degrees of freedom, $\beta = 1/k_B T$, and $Q = \int dx \exp(-\beta H)$. Under the relatively mild requirement that the surface (second) term on the right-hand side vanishes, use of Hamilton's equations leads to the exact result:

$$\langle p_i \dot{q}_i\rangle = k_B T \tag{2}$$

$$-\langle q_i \dot{p}_i\rangle = k_B T \tag{3}$$

where q_i and p_i label individual positions and conjugate momenta respectively, and the dot denotes a time derivative. The velocities are

$$v_i \equiv \dot{q}_i = \frac{\partial H}{\partial p_i} \tag{4}$$

We have presented Tolman's original proof for the canonical (*NVT*) ensemble, because of its brevity. For the microcanonical (*NVE*) ensemble, which is relevant to our development below, the proof is described elsewhere;[1] the result is identical, apart from an i-independent correction of order N_f^{-1}.[21] The use of periodic boundary conditions means linear momentum is often conserved in simulations. This constraint leads to a modification to eq 2 of order N^{-1}, where N is the number of particles in the simulation.[21,22] We ignore effects of this magnitude except where explicitly noted. More importantly, if q_i is a periodic coordinate, then the second term on the right-hand side of eq 1 may be nonzero, in which case eq 3 will not hold. A commonly encountered example is for simulations using periodic boundary conditions in which the position q_i of an atom is restricted to values that lie within the simulation box; assuming that the Hamiltonian is translationally invariant, then knowledge of a single positional coordinate q_i provides no information about $\dot{p}_i$, so evidently these quantities are uncorrelated, and $\langle q_i\dot{p}_i\rangle = \langle q_i\rangle\,\langle \dot{p}_i\rangle = 0$, as may also be demonstrated by explicitly evaluating the second term on the right-hand side of eq 1.

In MD simulations we usually use a Hamiltonian of the form

$$H_0(\mathbf{p}, \mathbf{q}) = U(\mathbf{q}) + \frac{1}{2}\mathbf{p}^T\mathbf{m}^{-1}\mathbf{p} \tag{5}$$

where $\mathbf{q}$ denotes atom positions, $\mathbf{m}$ is the diagonal mass matrix ($\mathbf{m}_{ij} = \delta_{ij}m_i$), and U is the force field. Since H_0 contains only quadratic terms in p_i, eq 4 shows that—for exact trajectories—velocities and momenta are related through $m_i v_i = p_i$. Eq 2 thus reduces to the familiar form of kinetic energy equipartition:

$$\langle m_i v_i^2\rangle_0 = k_B T \tag{6}$$

$$\langle p_i^2/m_i\rangle_0 = k_B T \tag{7}$$

where the 0 is used to emphasize that a Hamiltonian of the form H_0 is assumed. The shadow Hamiltonian corresponding to H_0 need not take this form, however, as we review below.

MD simulations are often performed subject to holonomic constraints, for example, to keep certain bond lengths fixed. Since eq 2 assumes an unconstrained ensemble average, it does not directly apply. In principle, one can construct a new Hamiltonian describing the dynamics of a system subject to

N_c constraints by finding $3N - N_c$ unconstrained generalized positions and their conjugate momenta; eq 2 will then apply. For our purposes, however, it is sufficient to establish two results. First, for any subset A of position coordinates that are not involved in a constraint with position coordinates outside that subset, we find—after some algebra—that a result similar to eq 2 holds

$$\sum_{i \in I_A} \langle p_i v_i \rangle = (N_{f;A} - N_{c;A}) k_B T \tag{8}$$

Here I_A contains the indices of the coordinates in A, and $N_{f;A} - N_{c;A}$ is the number of positional degrees of freedom in A minus the number of constraints to which they are subject. Second, in the common case where the constraints are functions only of interatomic distance, it is straightforward to identify some unconstrained generalized coordinates. Formally, a subset B of the positions in A may be identified and transformed to any linear average coordinate $Q_B = \sum_{i \in I_B} w_i q_i$, with $\sum_{i \in I_B} w_i = 1$ and $N_{f;B} - 1$ relative coordinates, such that Q_B is unconstrained. Thus

$$\langle P_B V_B \rangle = k_B T \tag{9}$$

where $V_B = \dot{Q}_B$ and $P_B = \sum_{i \in I_B} p_i$ is the momentum conjugate to Q_B. Typically, Q_B will be a center-of-mass coordinate. Although eqs 8 and 9 are intuitively obvious, we mention them here to make explicit that, like eq 2, they hold without need to assume a Hamiltonian of the form H_0. Only when a Hamiltonian of form H_0 is assumed can they be written in terms of squared velocities or momenta, as in eqs 6 and 7.

2.2. Shadow Hamiltonian. We focus on the velocity-Verlet integration of H_0, for which

$$\mathbf{q}(t + \delta_t) = \mathbf{q}(t) + \delta_t \mathbf{m}^{-1}\mathbf{p}(t) + \frac{\delta_t^2}{2}\mathbf{m}^{-1}\mathbf{F}(t) \tag{10}$$

$$\mathbf{p}(t + \delta_t) = \mathbf{p}(t) + \frac{\delta_t}{2}(\mathbf{F}(t) + \mathbf{F}(t + \delta_t)) \tag{11}$$

where the elements of the force vector are $F_i = -\partial U/\partial q_i$, as usual. This integration scheme is symplectic, a property that can be maintained in the presence of holonomic constraints.[23] A symplectic integrator is one for which the mapping $(\mathbf{p}(t), \mathbf{q}(t)) \rightarrow (\mathbf{p}(t + \delta_t), \mathbf{q}(t + \delta_t))$ is a canonical transformation, just as it is for continuous Hamiltonian dynamics. This suggests there might be a Hamiltonian whose *exact* dynamics generates the flow $(\mathbf{p}(t), \mathbf{q}(t)) \rightarrow (\mathbf{p}(t + \delta_t), \mathbf{q}(t + \delta_t))$. This shadow Hamiltonian H_{δ_t} is expected to be similar but not equal to H_0, whose *approximate* dynamics generates the same flow. Finding H_{δ_t} is a problem of backward error analysis.[15,24] One may construct an asymptotic expansion for H_{δ_t} by adding terms to H_0 to create a Hamiltonian whose exact dynamics matches that of eqs 10 and 11 order by order in δ_t. Since velocity Verlet is symmetric (reversing the sign of the time step gives the inverse method), only even powers of δ_t appear

$$H_{\delta_t}(\mathbf{p}, \mathbf{q}) = H_0(\mathbf{p}, \mathbf{q}) + \frac{\delta_t^2}{2!}\delta H^{(2)}(\mathbf{p}, \mathbf{q}) + \frac{\delta_t^4}{4!}\delta H^{(4)}(\mathbf{p}, \mathbf{q}) + ... \tag{12}$$

It is possible to construct accurate numerical approximations for the correction terms;[25] we note that an estimator (eq 68 of ref 26) for $\delta H^{(2)}$ with errors of order δ_t^2 (and thus an estimator of H_{δ_t} with errors of order δ_t^4) has existed for many years in the CHARMM code,[27] where it is called a “high-frequency correction.” Except for particularly simple forms of H_0, there is no guarantee eq 12 converges, but numerical tests to high orders have found the conservation of the shadow Hamiltonian to improve when successively higher order terms are included.[25]

The second-order term in eq 12 is[17]

$$\delta H^{(2)} = \frac{1}{6}(\mathbf{m}^{-1}\mathbf{p})^{\mathrm{T}}\mathbf{K}(\mathbf{m}^{-1}\mathbf{p}) - \frac{1}{12}\mathbf{F}^{\mathrm{T}}\mathbf{m}^{-1}\mathbf{F} \tag{13}$$

where the Hessian has elements $K_{ij} = \partial^2 U/\partial q_i \partial q_j$. Since $\mathbf{K}$ depends on $\mathbf{q}$, even at second-order, H_{δ_t} has a different form than H_0, and the velocities and momenta are thus no longer related through a simple mass factor.[17] Specifically, eq 4 gives

$$\mathbf{v} = \mathbf{m}^{-1}\mathbf{p} + \frac{\delta_t^2}{6}\mathbf{m}^{-1}\mathbf{K}\mathbf{m}^{-1}\mathbf{p} + \mathcal{O}(\delta_t^4) \tag{14}$$

2.3. Equipartition for the Shadow Hamiltonian. The nontrivial relationship between velocities and momenta (eq 14) for the shadow Hamiltonian has immediate consequences for equipartition. In particular, eq 2 clearly no longer precisely reduces to eqs 6 and 7. This suggests that the simulation temperature should be defined using $T \equiv T_{pv}$, where

$$k_B T_{pv} = \langle p_i v_i \rangle_{\delta_t} \tag{15}$$

and the δ_t subscript emphasizes that the ensemble average depends on the time step through the shadow Hamiltonian. With this definition, the temperature for all motions $\{i\}$ will be the same, if generalized equipartition is satisfied. The alternative quantities

$$k_B T_{p^2} = \langle p_i^2 / m_i \rangle_{\delta_t} \tag{16}$$

$$k_B T_{v^2} = \langle m_i v_i^2 \rangle_{\delta_t} \tag{17}$$

differ from T by an amount $\mathcal{O}(\delta_t^2)$. We use the terms pv, p^2, or v^2 formula to refer to any method of calculating the temperature that is in the spirit of eqs 15, 16, or 17, respectively. (The p^2 formula corresponds to the usual method of obtaining temperature when using velocity-Verlet integration, that is using only on-step momenta provided by the integrator.) As demonstrated below, it is also possible to calculate the temperature, making use of eq 3, provided appropriate (nonperiodic) coordinates are used. Note that the temperature, as defined in eq 15, is a property of the distribution sampled during the simulation, rather than a property of the desired distribution that would have been sampled in the small-time-step limit. Statistical reweighting,[24,28] while a powerful tool to infer the desired ensemble from the one sampled, thus cannot be directly applied to sampled p^2 values to obtain $T \equiv T_{pv}$. The temperature one would obtain by such a reweighting is instead the (known) temperature of the desired ensemble. This has been explicitly demonstrated (for *NVT* simulations using a Nosé–Poincaré thermostat) in numerical experiments where reweighting T_{p^2} accurately recovered the thermostat's target temperature.[24]

To calculate the temperature from eq 15, one needs to calculate both velocities and momenta. The momenta **p** are, by construction, directly available at every step of the simulation. The velocities $\mathbf{v} \equiv \dot{\mathbf{q}} \neq \mathbf{m}^{-1}\mathbf{p}$ are not but can be accurately estimated from several consecutive positions via interpolation, as described in Section 3.3. Since the standard terminology we have adopted can lead to confusion, we emphasize that velocity-Verlet samples *momenta*—specifically, the canonical momenta of the shadow Hamiltonian—and not velocities. Naturally, this state of affairs is unaltered if eqs 10 and 11 are explicitly written in terms of what are called velocity-Verlet velocities, $\mathbf{v}_{\mathrm{vv}} \equiv \mathbf{m}^{-1}\mathbf{p}$. As its definition shows, $\mathbf{v}_{\mathrm{vv}}$ is simply and precisely related to the *momenta* but differs from the velocities $\mathbf{v} \equiv \dot{\mathbf{q}}$, when $\delta_t > 0$.

In Section 2.1, we made a few comments on the applicability of the generalized equipartition formula. Here we make some related comments on the applicability of eq 15 to MD simulations. First, eq 15 was derived assuming that a shadow Hamiltonian exists, i.e., that the dynamics generated by the integrator is the exact dynamics of some underlying Hamiltonian. This is strictly true for certain simple forms of H_0 but not for biomolecular force fields, where the asymptotic expansion eq 12 does not converge. For simulations with small energy drift, however, we find much encouragement in earlier work[25] that Hamiltonians defined by truncating eq 12 can describe the dynamics generated by the integrator extremely accurately. Nevertheless, an important part of this paper is to test numerically whether generalized equipartition holds. Second, in addition to the usual statistical error, estimators for the temperature, based on eq 15, contain errors from the velocity interpolation. As discussed further below, n^{th}-order polynomial interpolation essentially leads to $\mathcal{O}(\delta_t{}^n)$ errors, thus sufficiently high-order interpolation can make the errors in the estimated temperature negligible for practical purposes. Third, thermostats and barostats used in MD simulations may entail modification to the equations of motion such that they are no longer of Hamiltonian form (this is the case for both Nosé–Hoover[29] and Berendsen[30] thermostats). Although a rigorous analysis of such effects on eq 15 appears possible for some thermostats, in this paper we make the simplifying assumption that any modifications can be ignored. This is intuitively reasonable for thermostats coupled to a large number of degrees of freedom and is borne out by our numerical results on ubiquitin, which are very similar for *NVE* and *NVT* simulations. Similarly, although we focus on straightforward MD simulations here, we expect eq 15 to be relevant to Monte Carlo sampling methods that use molecular dynamics, such as parallel tempering.[31] Finally, we note that, while eq 15 was derived assuming a symplectic integrator, milder conditions are sufficient. In particular, if the integrator conserves phase-space volume and has a conserved quantity $\tilde{H}_{\delta_t}(\mathbf{p}, \mathbf{q})$, which need not be a Hamiltonian, then eq 15 will hold if $\partial\tilde{H}_{\delta_t}/\partial p_i = v_i$.

2.4. Harmonic Oscillator. We briefly illustrate the above results for the simple case of a one-dimensional harmonic oscillator with mass m and spring constant k:

$$H_0^{\mathrm{osc}}(p, q) = \frac{1}{2}kq^2 + \frac{p^2}{2m} \qquad (18)$$

As is well-known (for example, see ref 17), the shadow Hamiltonian is a harmonic oscillator with modified mass and spring constant,

$$H_{\delta_t}^{\mathrm{osc}}(p, q) = \frac{1}{2}k_{\delta_t}q^2 + \frac{p^2}{2m_{\delta_t}} \qquad (19)$$

The modified parameters m_{δ_t} and k_{δ_t} are given by

$$\frac{m_{\delta_t}}{m} = \frac{\omega}{\omega_{\delta_t}}\sqrt{1 - \left(\frac{1}{2}\omega\delta_t\right)^2} \approx 1 - \frac{1}{6}(\omega\delta_t)^2 \qquad (20)$$

$$\frac{k_{\delta_t}}{k} = \frac{\omega_{\delta_t}}{\omega}\sqrt{1 - \left(\frac{1}{2}\omega\delta_t\right)^2} \approx 1 - \frac{1}{12}(\omega\delta_t)^2 \qquad (21)$$

where the modified frequency $\omega_{\delta_t} = (k_{\delta_t}/m_{\delta_t})^{1/2}$ is

$$\frac{\omega_{\delta_t}}{\omega} = \frac{\arcsin\left(\frac{1}{2}\omega\delta_t\right)}{\frac{1}{2}\omega\delta_t} \approx 1 + \frac{1}{24}(\omega\delta_t)^2 \qquad (22)$$

Both the shadow mass and spring constant decrease with increasing time step, and vanish as $\omega\delta_t \rightarrow 2$, which coincides with the stability limit of the integrator.

Suppose that the harmonic oscillator is weakly coupled to a heat bath at temperature T. The quantities T_{pv}, T_{p^2}, and T_{v^2} defined in the previous subsection are related to T as follows

$$\frac{T_{pv}}{T} = 1; \quad \frac{T_{p^2}}{T} = \frac{m_{\delta_t}}{m}; \quad \frac{T_{v^2}}{T} = \frac{m}{m_{\delta_t}} \qquad (23)$$

Whereas T_{pv} correctly shows the oscillator to have the same temperature as the bath, T_{p^2} underestimates and T_{v^2} overestimates the temperature by the same factor. The disagreement between the three estimators provides a valuable indication of the magnitude of truncation error. In addition, we may also use eq 3; if we define $k_{\mathrm{B}}T_{q\dot{p}} = -\langle q\dot{p}\rangle_{\delta_t}$ and $k_{\mathrm{B}}T_{qF} = \langle q(kq)\rangle_{\delta_t}$, then

$$\frac{T_{q\dot{p}}}{T} = 1; \frac{T_{qF}}{T} = \frac{k}{k_{\delta_t}} \qquad (24)$$

Finally, since some codes have estimators of the shadow energy available, it is of interest to estimate the temperature obtained using the difference of the shadow energy and the potential energy. With the definition $k_{\mathrm{B}}T_{H-U}/2 = \langle H_{\delta_t}\rangle - \langle kq^2/2\rangle$, we find

$$\frac{T_{H-U}}{T} = 2 - \frac{k}{k_{\delta_t}} \qquad (25)$$

showing that the difference of the shadow energy and potential energy yields a temperature estimator with errors of order $\delta_t{}^2$.

3. Simulation and Analysis Details

All simulations were performed using Desmond.[32] Periodic boundary conditions were used, and center-of-mass motion was removed every time step. Every picosecond, coordinates, and velocities were saved for nine consecutive time steps to allow interpolation. Since we wish to isolate truncation errors from round-off errors, Desmond and analysis programs used double-precision arithmetic. Energy drift was small: even assuming it is converted entirely into kinetic energy, the drift over the entire length of each *NVE* simulation corresponds to a temperature change of less than 0.2 K. All errorbars denote statistical errors, which were estimated using a blocking method.[33]

3.1. Diatomic Fluid. The simulated system contained 1000 oxygen-like diatomic molecules of 32 atomic mass units in a cubic box with a side length of 44.42 Å. The force field contained only two kinds of terms, each a simple function of interatomic distance r: intramolecular bond stretch terms of the form $k(r - r_0)^2/2$, and van der Waals interactions of Lennard-Jones form $4\varepsilon[(\sigma/r)^{12} - (\sigma/r)^6]$ between atoms on different molecules; there were no electrostatic interactions. Most parameters ($r_0 = 1.15$ Å, $\sigma = 2.9$ Å, and $\varepsilon = 0.586152$ kJ/mol) were taken to be the OPLS-AA/L force field[34] values for oxygen. The bond force constant was reduced by approximately a factor of six from the OPLS-AA/L value for oxygen to $k = 1368$ kJ/mol/Å^2, which corresponds to a lengthened bond-vibration period of 48.05 fs. These parameters ensure that energy transfer between the high-frequency vibrations and the other motions is rapid (equipartition is reached in hundreds of ps). The van der Waals interactions were truncated at 10 Å and calculated using a neighbor list of pairs within 11.25 Å that was updated every ∼15 fs. We performed 10 *NVE* simulations of 400 ns using the velocity-Verlet integration scheme with time steps ranging from 0.5 to 5 fs. We also performed 9 NVE simulations of 400 ns using the r-RESPA integrator[20] with bond stretches calculated every 0.5 fs and with intermolecular interactions between 2 and 10 times less frequently. All simulations started from the same configuration (which had been pre-equilibrated using a thermostat at 300 K) and were each assigned different initial velocities that were randomly chosen from a Maxwell−Boltzmann distribution.

3.2. Ubiquitin. The set up and parameters for the ubiquitin simulations were similar to those used previously.[35] PDB entry 1D3Z[36] was solvated with 5302 explicit water molecules, giving a total of 17 137 atoms in a cubic box of side 55.71 Å. We used the OPLS-AA/L all-atom force field,[34] as implemented in GROMACS version 3.1.4,[37] for the protein, together with the SPC water model.[38] Electrostatic forces were computed using the particle mesh Ewald method[39] with a screening Gaussian width of $10/(3\sqrt{2}) \approx 2.36$ Å and with fifth-order interpolation to a cubic mesh of 64 × 64 × 64 points; real-space contributions to the electrostatics and van der Waals interactions were truncated at 10 Å and calculated from a list of pairs separated by less than 11 Å that was assembled every ∼12 fs. Water molecules and lengths of bonds to hydrogens were rigidly constrained using M-SHAKE,[40] as implemented[41] in Desmond. Energy minimization and equilibration under conditions of constant temperature and pressure yielded a conformation, from which 8 simulations of 11 ns were started. These simulations were as follows: three *NVE* velocity-Verlet simulations with different time steps (1.25, 2, and 2.5 fs); one *NVE* simulation using an r-RESPA multiple time step scheme,[20] in which nonbonded interactions were evaluated every 2.5 fs and the remaining interactions every 1.25 fs; four simulations using a velocity-Verlet time step of 2 fs and coupled in different ways to Berendsen thermostats[30] with relaxation times of 0.5 ps (either a single thermostat coupled to the entire system, or the protein or water alone, or two independent thermostats with the first coupled to the water and the second to the protein).

To calculate the temperature of the protein, T_{pv}^{protein}, we made use of eq 8, where i ran over all protein atoms and Cartesian dimensions. The temperatures of the water and the entire system, T_{pv}^{water} and T_{pv}^{system}, were calculated analogously. The quantities $T_{p^2}^{\text{protein}}$, $T_{p^2}^{\text{water}}$, $T_{p^2}^{\text{system}}$ were calculated in the same way, except that the interpolated velocity was replaced by the corresponding momenta divided by the mass; this is how temperatures of components are normally calculated in simulation. We also calculated the analogous v^2 quantities. The position coordinates used for the quasiharmonic analysis were center-of-mass coordinates of protein heavy atoms and their covalently bonded hydrogen atoms. This gives a total of 1800 position coordinates when overall rotation and translation are excluded.

3.3. Interpolation. While positions and momenta are directly available from the integrator at every time step, their time derivatives are not, but can be estimated by interpolation. Here we use straightforward polynomial interpolation. For velocities, fitting an n^{th}-order polynomial through the positions at the time of interest and the following $n/2$ and preceding $n/2$ times and taking the time derivative gives a time-symmetric n^{th}-order approximation to the velocities, $\mathbf{v}^{(n)}$. The second-order result

$$\mathbf{v}^{(2)}(t) = (\mathbf{x}(t + \delta_t) - \mathbf{x}(t - \delta_t))/(2\delta_t) \tag{26}$$

simply recovers the velocity-Verlet velocities. Higher-order results may be expressed in terms of the velocity-Verlet results at different time steps. For example

$$\mathbf{v}^{(4)}(t) = \frac{1}{6}(8k_1 - k_2) \tag{27}$$

$$\mathbf{v}^{(6)}(t) = \frac{1}{30}(45k_1 - 9k_2 + k_3) \tag{28}$$

$$\mathbf{v}^{(8)}(t) = \frac{1}{420}(672k_1 - 168k_2 + 32k_3 - 3k_4) \tag{29}$$

where

$$k_n = \sum_{m=1}^{n} \mathbf{v}^{(2)}(t + (2m - n - 1)\delta_t) \tag{30}$$

Rates of change of momenta were calculated by fitting polynomials to successive momenta in an analogous manner. The same procedure may be used in the presence of constraints.

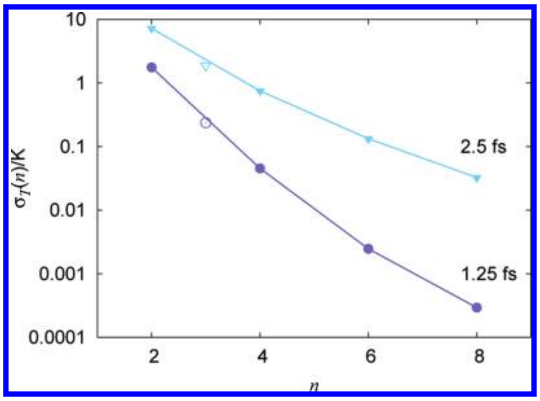

Figure 2. Estimated velocity errors as a function of interpolation order. The weighted rms velocity error defined in eq 31 is given in units of Kelvin. All points are from one of two *NVE* simulations of ubiquitin. The simulation time step for the dark circles was 1.25 fs and for the lighter triangles was 2.5 fs. The lines are just guides to the eye. The solid points are time-symmetric interpolations, and the open points with $n = 3$ are the time-asymmetric (Beeman) interpolations described in the text.

Although the interpolation error in $\mathbf{v}^{(n)}$ is $\mathcal{O}(\delta_t^{\,n})$, there is no guarantee of convergence as n increases. In practice, however, interpolation error appears to rapidly diminish with increasing n, for small n. This is illustrated in Figure 2, where we show the root-mean-square (rms) temperature difference between n^{th}- and tenth-order estimates for protein degrees of freedom in *NVE* simulations of ubiquitin. Specifically, we calculate $\sigma_T(n)$ which we define via

$$\sigma_T^2(n) = \frac{1}{k_B^2 N_f}\sum_{i,\alpha}\langle (p_{i\alpha}(v_{i\alpha}^{(n)} - v_{i\alpha}^{(10)}))^2\rangle_{\delta_t} \qquad (31)$$

where i runs over all protein atoms and α over dimensions x, y, and z. Unless otherwise stated, below we use eighth-order interpolation, which is sufficient to make interpolation error substantially smaller than statistical error in the results we present. Figure 2 suggests that even for the large 2.5 fs time step, eighth-order interpolation corresponds to typical errors in temperature of an individual degree of freedom of less than 0.1 K, and a fourth-order approximation is already a substantial improvement over the second-order result. We have also found interpolation to be useful for trajectories generated by the commonly used r-RESPA method, albeit with slower convergence when tested with outer time steps in the 4−6 fs range (data not shown).

Finally, polynomial fits through positions at times $t + \delta_t$ and earlier are also possible (Figure 2). Although not time symmetric, such interpolations may be useful for thermostats because they have the advantage of giving improved accuracy velocity estimates "on the fly." In particular, the third-order result recovers the velocities in Beeman's implementation of Verlet, $\tilde{\mathbf{v}}^{(3)}(t) = (-\mathbf{v}^{(2)}(t - \delta_t) + 2\mathbf{v}^{(1/2)}(t - \delta_t/2) + 2\mathbf{v}^{(2)}(t))/3$, where the half-step velocity is $\mathbf{v}^{(1/2)}(t - \delta_t/2) = (\mathbf{x}(t) - \mathbf{x}(t - \delta_t))/\delta_t$. Beeman's algorithm is often used instead of velocity Verlet or leapfrog when accurate velocities are important, and Figure 2 shows the improvement over the velocity-Verlet velocities.

4. Results

4.1. Diatomic Fluid. The diatomic fluid is a useful test system, because it contains anharmonicities and motions of different frequencies, yet is simple enough that approximation errors can be estimated based on analytical harmonic oscillator results. We performed velocity-Verlet simulations with different time steps. These ranged up to 5 fs, or about a tenth of the vibrational period of 48 fs. This range was chosen because, in biomolecular simulations, time steps of up to about a tenth of the fastest vibrational period are in common use. (A time step of 2 fs is often chosen, for example, in protein simulations in which bonds to hydrogen are constrained; the fastest motions—angle bending motions involving hydrogens and certain bond stretches—have periods of approximately 20 fs.) For each time step, we calculated the ratio of vibrational and translational temperatures. The translational temperature depends on the center-of-mass velocities and momenta $\{\mathbf{V}, \mathbf{P}\}$, while the vibrational temperature depends on $\{v, p\}$ (the rates of change of the bond lengths $\{r\}$ and the projections of relative momenta $\{\mathbf{p}_\text{r}\}$ along the bonds, respectively; see Appendix A):

$$k_\text{B}T_{pv}^{\text{trans}} = \frac{N_\text{mol}}{N_\text{mol} - 1}\overline{\langle \mathbf{P}\cdot\mathbf{V}\rangle_{\delta_t}} \qquad k_\text{B}T_{pv}^{\text{vib}} = \overline{\langle pv\rangle_{\delta_t}} \qquad (32)$$

The bar denotes an average over all N_mol molecules, and the $N_\text{mol}/(N_\text{mol} - 1)$ prefactor reflects the constraints on the center-of-mass momenta arising from the conservation of the total linear momentum. The atomic velocities were evaluated by polynomial fitting. The center-of-mass and vibrational velocities were calculated in terms of these interpolated velocities. As shown in Figure 3, the ratio $T_{pv}^{\text{vib}}/T_{pv}^{\text{trans}}$ is indeed unity within statistical error, showing that generalized equipartition is satisfied in the simulations.

The same ratio is shown on an enlarged y-scale in Figure 3a and compared to the results obtained using fourth, rather than eighth, order interpolation for the velocities. The observation that generalized equipartition holds, but appears to be violated for larger time steps when the fourth-order estimator is used, is consistent with the theoretical expectation that temperature can be estimated with errors that are smaller (higher order) than $\mathcal{O}(\delta_t^{\,4})$.

We also calculated alternative temperature estimators, starting with the typical p^2 formulas:

$$k_\text{B}T_{p^2}^{\text{trans}} = \frac{N_\text{mol}/M}{N_\text{mol} - 1}\overline{\langle \mathbf{P}\cdot\mathbf{P}\rangle_{\delta_t}}, \quad k_\text{B}T_{p^2}^{\text{vib}} = \overline{\langle p^2/\mu\rangle_{\delta_t}} \qquad (33)$$

where M and μ are molecular and reduced masses, respectively. As seen in Figure 3, when calculated this way, the vibrational (higher frequency) motion appears to be cooler. In Figure 3, we also show the results of estimating the temperature ratio using the v^2 formulas (with accurate velocity estimates):

$$k_\text{B}T_{v^2}^{\text{trans}} = \frac{N_\text{mol}M}{N_\text{mol} - 1}\overline{\langle \mathbf{V}\cdot\mathbf{V}\rangle_{\delta_t}}, \quad k_\text{B}T_{v^2}^{\text{vib}} = \overline{\langle \mu v^2\rangle_{\delta_t}} \qquad (34)$$

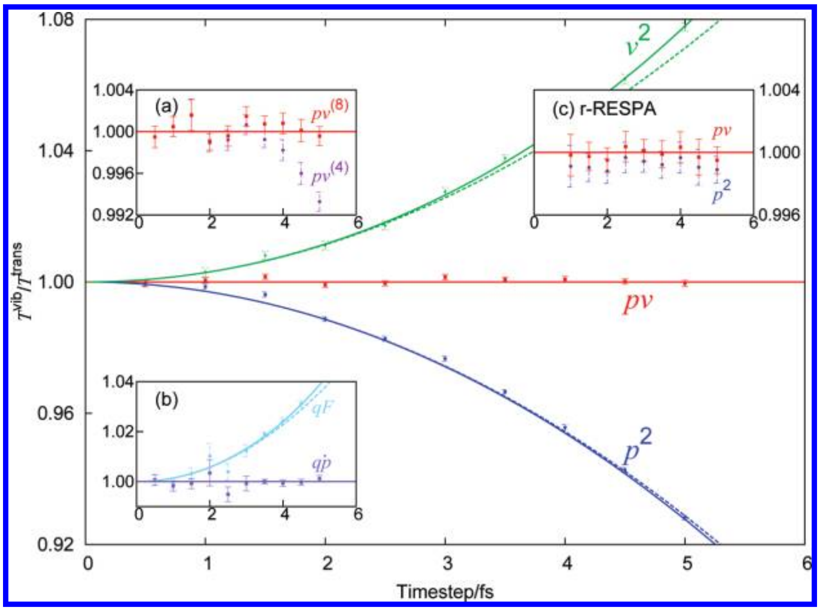

Figure 3. Equipartition for the diatomic fluid over a range of integration time steps. The main figure and insets (a) and (b) show the time-step-dependence of the ratio of different estimates (based on eqs 32−35) of the vibrational and translational temperatures for the same velocity-Verlet simulations. Lines show the harmonic oscillator results (eqs 20−24), with the dashed lines denoting asymptotes that only include terms up to $(\omega_0\delta_t)^2$. The main figure shows that, with the *pv* estimator, the ratio $T^{\mathrm{vib}}/T^{\mathrm{trans}}$ is one within error, indicating that generalized equipartition is achieved in the simulations. The p^2 estimator (blue) gives a ratio less than one, making the vibrational motions appear cooler, whereas the v^2 estimator gives a ratio greater than one. Inset (a) shows only the results of the *pv* estimator in a region close to $T^{\mathrm{vib}}/T^{\mathrm{trans}} = 1$. The red points are identical to those in the main figure, and the purple circles show the result of using velocities obtained from a lower-order polynomial interpolation (fourth rather than eighth). Inset (b) shows that the $q\dot{p}$ estimator gives a temperature ratio close to one, confirming generalized equipartition is satisfied, but that the *qF* estimator (see main text) makes the vibrations appear hotter. Inset (c) shows the result of the *pv* and p^2 estimators for r-RESPA simulations (with a fixed inner time step of 0.5 fs) as a function of the outer time step.

In this case, the higher frequency motion appears hotter. The magnitude of the deviations from conventional equipartition that are revealed using eqs 33 and 34 are also seen to be well predicted using the harmonic oscillator results of Section 2.4.

As shown in Appendix A, $T_{q\dot{p}}^{\mathrm{vib}}$, defined by

$$-k_{\mathrm{B}}T_{q\dot{p}}^{\mathrm{vib}} = \overline{\langle r\dot{p}\rangle_{\delta_t}} \tag{35}$$

is also essentially equal to the vibrational temperature. We calculated $T_{q\dot{p}}^{\mathrm{vib}}$ in terms of the bond vectors $\{\mathbf{r}\}$ and the molecular angular momenta and velocities $\{\mathbf{L}, \boldsymbol{\omega}\}$ using the identity $r\dot{p} \equiv \mathbf{r}\cdot\dot{\mathbf{p}}_{\mathrm{r}} + \boldsymbol{L}\cdot\boldsymbol{\omega}$, with $\mathbf{p}_{\mathrm{r}}$ and the atomic velocities contained in ω obtained from interpolation. Additionally, we calculated the analogous, but approximate, temperature estimator T_{qF}^{vib} (Appendix B). Figure 3b shows that $T_{q\dot{p}}^{\mathrm{vib}}$ is the same as the translational temperature within statistical error, consistent with generalized equipartition being satisfied. In contrast, the T_{qF}^{vib} estimator leads to hotter temperatures for the vibrational (i.e., higher frequency) motion. This deviation can again be understood in terms of the harmonic oscillator results. T_{qF} is relevant to pressure computations, as described in Appendix B. We find that the average pressure calculated using a p^2 expression for the ideal part and a virial part calculated in the usual manner[13] depends on whether a molecular or atomic expression is used. For the smallest (0.5 fs) time step, the molecular and atomic results are the same within statistical error of 0.5 bar, but for the largest (5 fs) time step they differ by 17.8 ± 0.4 bar, which is close to the 17.0 bar predicted by eq 44.

Although eqs 33 and 34 are less accurate than eq 32 for estimating temperature, the deviations from conventional equipartition that they reveal do reflect real differences between the shadow Hamiltonian and H_0. They thus provide a warning that truncation errors may affect other quantities. We find vibrational frequencies and the magnitude of bond length fluctuations to change, for example, by approximately 2% over the range of time steps studied, as would be expected on the basis of the harmonic oscillator results. One straightforward way to reduce truncation errors is to use a multiple-time-step scheme, where the stiff-bonded forces are evaluated more frequently than the softer intermolecular interactions. Since the intermolecular interactions usually dominate the computational expense, this approach often only has modest cost. Figure 3c shows the vibrational to translational temperature ratios for the pv and p^2 estimators from r-RESPA simulations of the diatomic fluid as a function of the outer time step. The intermolecular interactions were calculated on the outer time step, while the bonded interactions were evaluated every 0.5 fs. The agreement between the estimators is excellent even for large outer time steps; the v^2 estimator (not shown) is also in agreement with the other estimators.

4.2. Ubiquitin. As described in Appendix A, it is straightforward to calculate the temperature of different quasiharmonic motions of a protein. We have done this for different simulations of ubiquitin solvated in water. Initially, we performed *NVE* simulations with differing values of δ_t. We calculated the temperature $T_{pv}^{(i)}$ using eq 40 for all quasihar-

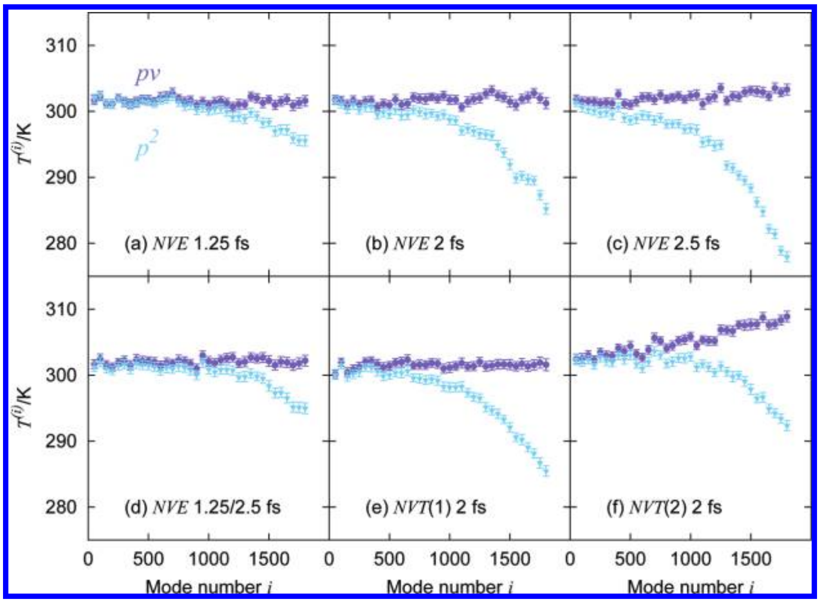

Figure 4. Temperature as a function of mode number *i* for ubiquitin. The modes are ordered by mean-square fluctuation, with the largest-amplitude (lowest-frequency) motions to the left. The temperatures are averages over 50 consecutive modes. The dark circles show the results of the *pv* formula (eq 40), and the light triangles are the results of the corresponding p^2 formula. Panels (a−c) show the results of *NVE* simulations with time steps of 1.25, 2, and 2.5 fs. Panel (d) shows results of the *NVE* simulation using a r-RESPA multiple time step method. Panel (e) shows the results when the entire system is coupled to a single thermostat, whereas panel (f) shows the results of simultaneously applying independent thermostats to water and protein.

monic modes *i*. As shown in Figure 4a−c, which corresponds to time steps of 1.25, 2, and 2.5 fs, $T_{pv}^{(i)}$ is essentially independent of *i*. This strongly suggests that generalized equipartition is indeed achieved in these simulations. We also estimated temperatures using $k_B T_{p^2}^{(i)} = \langle p_i^2 \rangle_{\delta_t}$. This estimate varies substantially by mode and, as expected, is significantly lower for higher-frequency motions. The deviation from conventional equipartition increases with the size of the time step (Figure 4). We also find $T_{v^2}^{(i)} = \langle v_i^2 \rangle_{\delta_t} / k_B$ to have a strong but opposite dependence on mode (not shown). Note that all three estimates agree well for the low-frequency motions. Since the v^2 formulas are found to overestimate temperature by approximately as much as the p^2 formula underestimates it, we just discuss the *pv* and p^2 results below.

It is natural to try to reduce the truncation errors signaled by the breakdown of conventional equipartition. As noted above, one approach is to use a multiple-time-step method. In Figure 4d, we show the results of using the r-RESPA method with bonded forces calculated every 1.25 fs and with nonbonded interactions every 2.5 fs. The magnitude of the discrepancy between $T_{pv}^{(i)}$ and $T_{pp}^{(i)}$ is reduced to an amount similar to the 1.25 fs time step velocity-Verlet simulation. This is because the highest frequency motions are bond vibrations and angle-bending motions that involve hydrogen atoms. The r-RESPA solution is inexpensive, if calculating the bonded interactions takes a relatively small part of the overall computation time, which is typically the case.

If a thermostat is applied to the system and the instantaneous temperature—which determines thermostat, and hence particle, dynamics—is estimated using the p^2 formula, then integration errors may be amplified. A common, if fairly innocuous, case is illustrated in Figure 4e, where a single Berendsen thermostat is applied to the entire system. The temperature as a function of mode is very similar to the *NVE* simulation with the same (2 fs) time step. There is a small discrepancy between the system temperature (301.76 ± 0.02 K) and the target temperature of 300 K. The discrepancy reflects the fact that the p^2 formula slightly underestimates the water temperature ($T_{p^2}^{\text{water}} < T_{pv}^{\text{water}}$).

Table 1. Water-Protein Temperature Differences from p^2 and v^2 Estimators for *NVE* Simulations[a]

time step/fs	ΔT_{p^2}	ΔT_{pv}
1.25	2.11	−0.11
2.0	5.62	−0.06
2.5	8.91	−0.08

[a] Temperature differences are expressed as $\Delta T = T^{\text{water}} - T^{\text{protein}}$ in Kelvin. Statistical errors are approximately 0.1 to 0.2 K.

Water models used in biomolecular simulation are often rigid, so the highest frequency motions are in the protein. Thus, although with the p^2 formula the water may only appear a degree or two cooler than the true value, the protein may appear substantially cooler, leading to an apparent temperature difference, $\Delta T_{p^2} \equiv T_{p^2}^{\text{water}} - T_{p^2}^{\text{protein}}$. In the NVE simulations with a 2 fs time step, for example, $\Delta T_{p^2} = 5.6 \pm 0.1$ K, and this rises to 8.9 ± 0.2 K for a 2.5 fs time step, whereas the accurately calculated temperature difference $\Delta T_{pv} \equiv T_{pv}^{\text{water}} - T_{pv}^{\text{protein}}$ is 0 within error in both cases; see Table 1). If the symptom of truncation error revealed by the p^2 estimator is combatted by applying two thermostats simultaneously—one to protein and one to solvent—then larger errors result than in the case of a single system-wide thermostat. This is not due to an intrinsic problem with the use of multiple thermostats, which can be used safely if an appropriate time step is chosen. Rather, the relatively large time steps commonly chosen in MD simulations for ef-

Table 2. Estimated Component Temperatures for *NVT* Simulations Performed with Different Thermostats[a]

thermostat	$T_{p^2}^{\text{system}}$	$T_{p^2}^{\text{water}}$	$T_{p^2}^{\text{protein}}$	T_{pv}^{system}	T_{pv}^{water}	T_{pv}^{protein}
one (system)	299.97	300.47	294.80	301.76	301.76	301.78
one (water)	299.49	299.97	294.55	301.28	301.26	301.52
one (protein)	305.02	305.50	300.03	306.84	306.81	307.14
two (protein, water)	299.98	300.05	299.18	301.78	301.35	306.30

[a] Temperatures are in Kelvin. The target temperature was 300.0 K in all simulations. The simulations differ only in the number of thermostats (one or two) and the atoms to which the thermostats are coupled. Statistical errors are approximately 0.1 to 0.2 K for protein temperatures and 0.02 to 0.04 K for water/system temperatures.

ficiency lead to errors that can be amplified by certain choices of thermostat. For our simulation with a 2 fs time step and two thermostats, we find that $T_{pv}^{\text{protein}} = 306.3 \pm 0.1$ K, whereas $T_{pv}^{\text{water}} = 301.35 \pm 0.02$ K; in addition $T_{pv}^{(i)}$ is no longer approximately constant (Figure 4f). This signals a breakdown of generalized equipartition. Energy flows from one thermostat to the protein, then transfers to the water, and is finally removed by the other thermostat. Applying a single thermostat to the system, but just coupling it to a subset of the particles, should not lead to a breakdown of generalized equipartition, although if the component contains high-frequency motions, then this can lead to a substantial error in the system temperature. For example, we find that coupling a single thermostat to the protein leads to a simulation temperature about 7 K above the target temperature; see Table 2. We stress that the underlying cause of these problems is truncation error, not a problem with the thermostat itself.

5. Discussion and Conclusions

In this paper, using the established concepts of generalized equipartition and the shadow Hamiltonian, we have introduced a clear definition of simulation temperature that explicitly takes into account the finite simulation time step. We have shown that this temperature can be evaluated accurately and straightforwardly in practice. We tested generalized equipartition in numerical examples relevant to biomolecular simulation in which truncation errors lead to deviations from conventional equipartition and thus to different temperature estimates for different motions when conventional estimators are used. We confirmed that generalized equipartition is in fact satisfied in these examples, with different motions sharing a single well-defined simulation temperature.

The observation that generalized equipartition can be satisfied even for rather large time steps naturally does not imply that the simulations are free from artifacts due to truncation error, but it does help highlight the actual nature of the errors. As signaled by the breakdown of conventional equipartition, the shadow Hamiltonian differs from the Hamiltonian that we wish to simulate by an amount $\mathcal{O}(\delta_t^2)$, and thus their dynamics and thermodynamics will differ too.

One practical benefit of obtaining accurate temperature estimates, even when the Hamiltonian itself is subject to $\mathcal{O}(\delta_t^2)$ errors, is that testing generalized equipartition can be a valuable simulation diagnostic. A violation of equipartition demonstrated using the methods of this paper points to underlying problems with the integration scheme, as in the two-thermostat example described above. A second practical benefit is that accurately estimating the temperature can remove what may be the largest source of error in the description of low-frequency motions, as we now briefly explain. Low-frequency motions are often of greatest interest, and by their nature, most error in their description comes via their coupling to higher-frequency motions, which present more of a challenge to the integrator. If high- and low-frequency motions are weakly coupled, as expected for bond vibrations and larger-scale protein conformational change, for example, the low-frequency dynamics should be accurately described by the integrator. Error in the estimated temperature can then become the dominant error in the overall description of the low-frequency motion, because the temperature—when computed as a sum over all atomic motions using the p^2 formula—is polluted by errors due to the fast motions.

Our results also make clear that for Verlet integration, estimating the temperature using the v^2 formula leads to $\mathcal{O}(\delta_t^2)$ errors even if the velocities could be computed exactly. A corollary is that Beeman's version of Verlet, which gives velocities with only $\mathcal{O}(\delta_t^3)$ errors, will still yield temperatures with $\mathcal{O}(\delta_t^2)$ errors if those temperatures are estimated using a v^2 formula, as is conventional when using this integrator. Most simulations use some form of temperature control, and an inaccurate estimated temperature can affect the dynamics through the thermostat (or barostat). Fortunately, for the common case of a small globular protein solvated by constrained water molecules and coupled to a single system-wide thermostat, the resultant errors will be small because the fastest motions are in the protein, which comprises only a small part of the system. Care might be needed if the system contains a larger fraction of high-frequency motions, as would be the case in a simulation of a protein crystal or a lipid bilayer or in a simulation using an unconstrained water model. Clearly, systems that are particularly sensitive to temperature and pressure are more likely to exhibit substantial artifacts. Systems near a phase transition, for example, need more care; under ambient conditions, such systems include certain lipid bilayers and marginally stable small peptides and proteins.

For the simulation thermostat to accurately control temperature, it would be desirable to calculate the instantaneous temperature using the *pv* formula. We have shown that improved estimates of temperature at a given time are possible using information that can in principle be made available by the integrator (see the non-time-symmetric interpolation in Figure 2). Although constructing a thermostat along these lines is possible, this may not be the most promising approach. In addition to breaking time reversibility, such a thermostat would require the MD code to retain information about particle positions from earlier time steps, thereby adding complexity and likely reducing performance of a parallel code. One simple way to side-step these issues may be to continue to use the p^2 formula but to couple the thermostat to lower-frequency motions. This approach does not remove $\mathcal{O}(\delta_t^2)$ errors from the p^2 temperature estimate, but it can substantially reduce the prefactor. In our *NVE* ubiquitin simulations, for example, we find the error in $T_{p^2}^{\mathrm{water}}$ to be reduced by a factor of over five when it is calculated from translational motion only, rather than translational and rotational. This suggests that coupling a single thermostat to the translational motion of water molecules may be a useful approach.

Our results warn against combining multiple thermostats with large time steps. With a properly chosen time step, multiple thermostats can be a valuable tool to ensure equilibration even when equipartitioning is slow.[42] Application of multiple thermostats was also once useful to control very large hot solvent−cold solute artifacts, such as can occur when cutoff electrostatics are used. The improvements in methodology and computer codes over the last 15−20 years, however, have led to a situation where the dominant deviations from conventional equipartition are truncation errors and the generalized equipartition is satisfied. In such a situation, using multiple thermostats to rectify the deviation from conventional equipartition will have the counterproductive effect of causing true temperature imbalances in the system. Although in this paper we reported results obtained with the Berendsen thermostat, we have also found very similar results in tests with a Nosé−Hoover thermostat. Caution may also be required when combining stochastic thermostats with large time steps (particularly if the thermostat relaxation time is short), because such thermostats are typically coupled to many individual degrees of freedom (as in Langevin dynamics, for example).

The simplest way to reduce truncation error is obvious: reduce the time step. In practice, there is often reluctance to do this, in part because of the large computational expense of simulations and in part because of the fact that, while artifacts undeniably exist, their direct impact is largest on fast motions and their effect on properties likely to be of interest in long-time-scale simulations is much less clear. Indeed, partly motivated by the observation that even the large commonly used time steps (of about a tenth of the period of the fastest motions) are approximately a factor of three below the stability limit of velocity Verlet, some authors have suggested increasing the time step further.[43,44] By showing that generalized equipartition can hold even for time steps somewhat beyond the commonly used range, our results lend some support to this idea. On the other hand, regardless of whether generalized equipartition is satisfied, truncation errors will affect simulation results, and it is difficult to assess the impact on properties of interest in complicated biological systems. Thus, a more promising approach to balancing accuracy and efficiency may be to change the integrator. Results for both our test systems highlighted the effectiveness of the r-RESPA integrator for reducing errors at little cost; such an approach is likely to be useful for biomolecular simulation, since the fast motions that are the major source of truncation error are usually inexpensive to calculate and can thus be calculated with a reduced time step at little cost. In some cases, in particular on specialized hardware that greatly accelerates nonbonded interactions,[45] a substantial fraction of time may be spent on bonded interactions. In future work, we will describe new integrators that increase accuracy efficiently in such cases.

Appendix A

Alternative Coordinate Systems. It can be helpful to calculate temperatures for different modes of motion, such as the collective motions of a large subset of atoms. In principle, a straightforward recipe to do this is to identify a canonical transformation between the Cartesian atomic coordinates and conjugate momenta and a set of coordinates of interest. If this can be done, generalized equipartition should then hold for the new variables, which can be evaluated in terms of the atomic coordinates and momenta provided by the integrator. Time derivatives of the generalized coordinates may be obtained using interpolation. We illustrate with two examples relevant to the systems studied in this paper.

Translation, Vibration, and Rotation for a Diatomic Molecule. Consider a diatomic molecule, which may be part of a larger system, that consists of atoms A and B with mass m_A and m_B, respectively. The transformation from the atomic positions and momenta ($\mathbf{q}_A$, $\mathbf{q}_B$, $\mathbf{p}_A$, $\mathbf{p}_B$) to center-of-mass and relative positions and momenta ($\mathbf{R} = (m_A\mathbf{q}_A + m_B\mathbf{q}_B)/M$, $\mathbf{r} \equiv (x, y, z) = \mathbf{q}_B - \mathbf{q}_A$, $\mathbf{P} = \mathbf{p}_A + \mathbf{p}_B$, $\mathbf{p}_r \equiv (p_{r;x}, p_{r;y}, p_{r;z}) = \mu(\mathbf{p}_B/m_B - \mathbf{p}_A/m_A)$), where $M = m_A + m_B$ and $1/\mu = 1/m_A + 1/m_B$, is canonical. So is the further transformation of the relative motion into vibrational and rotational motion ($r = |\mathbf{r}|$, $\theta = \arccos(z/r)$, $\phi = \arctan(y/x)$, $p = \mathbf{p}_r\cdot\mathbf{r}/r$, $l_\theta = -(x^2 + y^2)^{1/2}p_{r;z} + (xp_{r;x} + yp_{r;y})z/(x^2 + y^2)^{1/2}$, $l_\phi = xp_{r;y} - yp_{r;x}$). Generalized equipartition relations may thus be written for translational and internal motion and for rotational and vibrational contributions to the internal motion:

$$\frac{1}{3}\langle \mathbf{P}\cdot\mathbf{V}\rangle_{\delta_t} = \frac{1}{3}\langle \mathbf{p}_r\cdot\mathbf{v}_r\rangle_{\delta_t} = k_BT \tag{36}$$

$$\frac{1}{2}\langle \mathbf{L}\cdot\boldsymbol{\omega}\rangle_{\delta_t} = \langle pv\rangle_{\delta_t} = k_BT \tag{37}$$

Here $\mathbf{V} = \dot{\mathbf{R}}$, $\mathbf{v}_r = \dot{\mathbf{r}}$, $v = \dot{r}$, and we have identified $\mathbf{L}\cdot\boldsymbol{\omega} \equiv l_\theta\dot{\theta} + l_\phi\ddot{\phi}$, where by definition $\mathbf{L} = \mathbf{r} \times \mathbf{p}_r$ and $\boldsymbol{\omega} = \mathbf{r} \times \mathbf{v}_r/r^2$ as usual. Likewise, using eq 3, we have the additional expressions for the total internal and the vibrational motion:

$$\frac{1}{3}\langle \mathbf{r}\cdot\dot{\mathbf{p}}_r\rangle_{\delta_t} = \langle r\dot{p}\rangle_{\delta_t} = -k_BT \tag{38}$$

which are valid even with periodic boundary conditions (in the unlikely situation that the bond length can exceed half the simulation box length, care must be taken not to incorrectly wrap the relative position coordinate). Replacement of $\dot{\mathbf{p}}_r$ in the above formula with the analogously defined relative force

$$\mathbf{F}_r = \mu(\mathbf{F}_B/m_B - \mathbf{F}_A/m_A) \tag{39}$$

where $\mathbf{F}_A = -\partial U/\partial\mathbf{q}_A$ denotes an atomic force, would be an approximation for finite integration time steps.

The generalized equipartition formulas obtained in this section are very familiar in the case $\delta_t \to 0$; the formal reasoning here makes clear that they should also apply to the shadow Hamiltonian and thus hold in simulation, provided that the time derivatives that they contain can be accurately estimated.

Quasiharmonic Motions of a Protein. Quasiharmonic analysis and principal component analysis are popular closely related methods for analyzing protein motions[46] and have occasionally been used in the context of equipartition.[47] The basic approach can often be decomposed into three transformations: First, and optionally, the protein atoms or some subset are mass-weighted ($\mathbf{q}_0$, $\mathbf{p}_0$) → ($\mathbf{q}_1 = \mathbf{m}^{1/2}\mathbf{q}$, $\mathbf{p}_1 = \mathbf{m}^{-1/2}\mathbf{p}$); then these coordinates are transformed by means of a $\mathbf{q}_1$-dependent overall translation and rotation to minimize the root-mean-square deviation (rmsd) of $\mathbf{q}_1$ to a reference structure, ($\mathbf{q}_1$, $\mathbf{p}_1$) → ($\mathbf{q}_2 = \mathbf{A}_{\text{rot}}\hat{A}_{\text{trans}}\mathbf{q}_1$, $\mathbf{p}_2 = \mathbf{A}_{\text{rot}}\mathbf{p}_1$); finally, an orthogonal transformation ($\mathbf{q}_2$, $\mathbf{p}_2$) → ($\mathbf{q}' = \mathbf{R}^{\text{T}}\mathbf{q}_2$, $\mathbf{p}' = \mathbf{R}^{\text{T}}\mathbf{p}_2$) makes the covariance matrix $\langle(\mathbf{q}' - \langle\mathbf{q}'\rangle)(\mathbf{q}' - \langle\mathbf{q}'\rangle)^{\text{T}}\rangle_{\delta_t} = \mathbf{R}^{\text{T}}\langle(\mathbf{q}_2 - \langle\mathbf{q}_2\rangle)(\mathbf{q}_2 - \langle\mathbf{q}_2\rangle)^{\text{T}}\rangle_{\delta_t}\mathbf{R}$ diagonal. Motion along a subset of the $\mathbf{q}'$ coordinates with the largest eigenvalues often correspond to interesting fluctuations of the protein around its native state. It is natural to define a temperature for mode i via

$$\langle p'_i v'_i\rangle_{\delta_t} = k_{\text{B}}T_{pv}^{(i)} \tag{40}$$

(or $\langle q'_i \dot{p}'_i\rangle_{\delta_t} = k_{\text{B}}T_{q\dot{p}}^{(i)}$), where $v'_i = \dot{q}'_i$. Unfortunately, although the first and last transformations above are canonical, the momenta generated in the rmsd-fitting step are only approximations to the true conjugate momenta. We thus expect $T_{pv}^{(i)}$ to differ slightly from T. In practice, we expect this discrepancy to be small for an ordered protein with a large number of degrees of freedom, and we neglect it.

Appendix B

Pressure. We show here, by means of a simple example, that the equivalence of atomic and molecular definitions of the simulation pressure for $\delta_t > 0$ may be viewed as a consequence of generalized equipartition and that this equivalence is broken, if the virial is computed in the normal way and the temperature is evaluated using a p^2 (or indeed pv or v^2) estimate. Although it is straightforward to obtain exact expressions for the simulation pressure, we have not found a practical method to estimate it from simulation data in a way that is as simple as evaluating the simulation temperature. Our results suggest that using a pv estimate for temperature will reduce pressure errors relative to a p^2 estimate but not eliminate them. This is consistent with the work of Pastor et al., who demonstrated that a different estimator of temperature (derived from Verlet velocities from the previous half-step) leads to exact estimates of pressure for a harmonic oscillator (unlike the use of p^2, pv, or v^2 estimates) due to a favorable cancellation of errors.[48]

In the canonical ensemble, starting from the thermodynamic definition of pressure as a volume derivative of the free energy, $P = -\partial F/\partial V$, it is straightforward to express the simulation pressure as a sum of kinetic and virial contributions. Assuming periodic boundary conditions and no constraints and viewing the Hamiltonian as a function of atomic positions and momenta, one obtains the *atomic* expression:

$$P = \frac{N}{V}k_{\text{B}}T - \left\langle\left(\frac{\partial H_{\delta_t}(\mathbf{p}, \mathbf{q}, V)}{\partial V}\right)_{\mathbf{p},\mathbf{q}}\right\rangle_{\delta_t} \tag{41}$$

In terms of molecular center-of-mass positions and momenta $\{\mathbf{R}, \mathbf{P}\}$, and relative coordinates $\{\mathbf{r}, \mathbf{p}_{\text{r}}\}$, the following *molecular* expression is more natural:

$$P = \frac{N_{\text{mol}}}{V}k_{\text{B}}T - \left\langle\left(\frac{\partial H_{\delta_t}(\{\mathbf{P}, \mathbf{R}, \mathbf{p}_{\text{r}}, \mathbf{r}\}, V)}{\partial V}\right)_{\{\mathbf{P},\mathbf{R},\mathbf{p}_{\text{r}},\mathbf{r}\}}\right\rangle_{\delta_t} \tag{42}$$

N_{mol} denotes the number of molecules, and N denotes the number of atoms. The two expressions are equivalent, but this equivalence can be broken by the (approximate) method used to compute the temperature and virial, as we explain by means of a simple example.

Consider an ideal diatomic gas with some intramolecular bonded interaction but negligible intermolecular interactions. Since the molecular virial vanishes, the molecular pressure formula immediately yields the correct result, $PV = N_{\text{mol}}k_{\text{B}}T$. (Assuming that the T appearing in the molecular pressure is obtained from the kinetic energy of molecular center-of-mass motion, T will be estimated correctly for this idealized system regardless of whether a pv or p^2 formula is used, provided the integrator—like velocity Verlet—preserves translational invariance.) We find that the atomic virial of our ideal system reduces to $(N_{\text{mol}}/3)\overline{\langle\mathbf{r}\cdot\dot{\mathbf{p}}_{\text{r}}\rangle_{\delta_t}}$, where the overbar simply denotes an average over all molecules. If generalized equipartition (eq 38) holds, then the atomic virial further reduces to $-N_{\text{mol}}k_{\text{B}}T$ and thus precisely cancels half of the kinetic term, yielding the correct pressure. The natural approach to computing the pressure when using velocity-Verlet integration, however, is to estimate T from the atomic momenta using a p^2 formula and from the atomic virial using $(N_{\text{mol}}/3)\overline{\langle\mathbf{r}\cdot(-\partial U/\partial\mathbf{r})\rangle_{\delta_t}}$. (This is essentially the approach implemented in Desmond,[32] for example.) Then we find that the atomic pressure differs from the molecular pressure according to

$$(P_{\text{mol}}^{\text{ideal}} - P_{\text{atom}}^{\text{ideal}})V = \frac{1}{3}N_{\text{mol}}k_{\text{B}}(T_{qF}^{\text{vib}} - T_{p^2}^{\text{vib}}) \tag{43}$$

where the two different approximations to the vibrational temperatures are $T_{p^2}^{\text{vib}} = \overline{\langle p^2/\mu\rangle_{\delta_t}}$, and $T_{qF}^{\text{vib}} = \overline{\langle rF_{\text{eff}}\rangle_{\delta_t}}$. The effective force includes a centrifugal term and is defined via $rF_{\text{eff}} = \mathbf{r}\cdot\mathbf{F}_{\text{r}} + \mathbf{L}\cdot\mathbf{L}/(\mu r^2)$, with $\mathbf{F}_{\text{r}}$ defined in eq 39.

Equation 43 depends on the details of the intramolecular interaction, but if we assume that the effective intramolecular potential, i.e., with a $L^2/(2\mu r^2)$ centrifugal term included, is approximately harmonic, then we may use the results of Section 2.4 to yield

$$\begin{aligned}(P_{\text{mol}}^{\text{ideal}} - P_{\text{atom}}^{\text{ideal}})V &= \frac{1}{3}N_{\text{mol}}k_{\text{B}}T\left(\frac{k}{k_{\delta_t}} - \frac{m_{\delta_t}}{m}\right)\\ &\approx \frac{1}{3}N_{\text{mol}}k_{\text{B}}T\left(\frac{1}{12}(\omega\delta_t)^2 + \frac{1}{6}(\omega\delta_t)^2\right)\\ &= N_{\text{mol}}k_{\text{B}}T\frac{(\omega\delta_t)^2}{12}\end{aligned} \tag{44}$$

Using the atomic formula, in the way described above, will thus underestimate the simulation pressure. For a time step of $\omega\delta_t = 0.6$, the overall error in the atomic pressure is 3% of the ideal gas value. For condensed-phase systems under ambient conditions, similar errors could easily dominate the pressure, which is itself the difference of two large almost canceling terms. For compressible systems, such as membranes, this is cause for caution. Using a molecular (or group-based) pressure is one obvious way to reduce errors (in our simple example, this approach eliminates errors). Alternatively, or in addition, the temperature estimate could be improved. Since two-thirds of the error in the atomic pressure estimate originates in the temperature estimate and one-third from the virial estimate, use of a perfect estimate of temperature will improve the pressure estimate but will not eliminate errors. As noted above, however, Pastor et al. have shown how the temperature estimate may be changed to cancel errors arising from the virial part, thus providing an even more accurate pressure estimator; this method is available in CHARMM.[27]

Acknowledgment. We thank Rebecca Kastleman for editorial assistance.

References

(1) Huang, K. *Statistical Mechanics*; John Wiley & Sons: New York, 1987.

(2) Straub, J. E.; Thirumalai, D. *Proteins: Struct., Funct., Genet.* **1993**, *15*, 360–373.

(3) Tobias, D. J.; Martyna, G. J.; Klein, M. L. *J. Phys. Chem.* **1993**, *97*, 12959–12966.

(4) Lingenheil, M.; Denschlag, R.; Reichold, R.; Tavan, P. *J. Chem. Theory Comput.* **2008**, *4*, 1293–1306.

(5) Feller, S. E.; Zhang, Y.; Pastor, R. W.; Brooks, B. R. *J. Chem. Phys.* **1995**, *103*, 4613–4621.

(6) Harvey, S. C.; Tan, R. K.-Z.; Cheatham, T. E., III. *J. Comput. Chem.* **1998**, *19*, 726–740.

(7) Mor, A.; Ziv, G.; Levy, Y. *J. Comput. Chem.* **2008**, *29*, 1992–1998.

(8) Levitt, M.; Sharon, R. *Proc. Natl. Acad. Sci. U.S.A.* **1988**, *85*, 7557–7561.

(9) Guenot, J.; Kollman, P. A. *Protein Sci.* **1992**, *1*, 1185–1205.

(10) Arnold, G. E.; Ornstein, R. L. *Proteins: Struct., Funct., Genet.* **1994**, *18*, 19–33.

(11) Sagui, C.; Darden, T. A. *Annu. Rev. Biophys. Biomol. Struct.* **1999**, *28*, 155–179.

(12) Swope, W. C.; Andersen, H. C.; Berens, P. H.; Wilson, K. R. *J. Chem. Phys.* **1982**, *76*, 637–649.

(13) Allen, M. P.; Tildesley, D. J. *Computer simulation of liquids*; Oxford University Press: New York, 1989.

(14) Ruth, R. D. *IEEE Trans. Nucl. Sci.* **1983**, *30*, 2669–2671.

(15) Hairer, E.; Lubich, C.; Wanner, G. Geometric Numerical Integration. In *Structure-Preserving Algorithms for Ordinary Differential Equations; volume 31 of Springer Series in Computational Mathematics*: Springer: Berlin, Germany, 2006.

(16) Tolman, R. C. *Phys. Rev.* **1918**, *11*, 261–275.

(17) Gans, J.; Shalloway, D. *Phys. Rev. E: Stat. Phys., Plasmas, Fluids, Relat. Interdiscip. Top.* **2000**, *61*, 4587–4592.

(18) Cuendet, M. A.; van Gunsteren, W. F. *J. Chem. Phys.* **2007**, *127*, 184102.

(19) MacGowan, D.; Heyes, D. M. *Mol. Simul.* **1988**, *1*, 277–297.

(20) Tuckerman, M.; Berne, B. J.; Martyna, G. J. *J. Chem. Phys.* **1992**, *97*, 1990–2001.

(21) Uline, M. J.; Siderus, D. W.; Corti, D. S. *J. Chem. Phys.* **2008**, *128*, 124301.

(22) Shirts, R. B.; Burt, S. R.; Johnson, A. M. *J. Chem. Phys.* **2006**, *125*, 164102.

(23) Leimkuhler, B. J.; Skeel, R. D. *J. Comput. Phys.* **1994**, *112*, 117–125.

(24) Bond, S. D.; Leimkuhler, B. J. *Acta Numerica* **2007**, *16*, 1–65.

(25) Engle, R. D.; Skeel, R. D.; Drees, M. *J. Comput. Phys.* **2005**, *206*, 432–452.

(26) Zhou, J.; Reich, S.; Brooks, B. R. *J. Chem. Phys.* **2000**, *112*, 7919–7929.

(27) Brooks, B. R. et al. *J. Comput. Chem.* **2009**, *30*, 1545–1614.

(28) Izaguirre, J. A.; Hampton, S. S. *J. Comput. Phys.* **2004**, *200*, 581–604.

(29) Hoover, W. G. *Phys. Rev. A: At., Mol., Opt. Phys.* **1985**, *31*, 1695–1697.

(30) Berendsen, H. J. C.; Postma, J. P. M.; van Gunsteren, W. F.; DiNola, A.; Haak, J. R. *J. Chem. Phys.* **1984**, *81*, 3684–3690.

(31) Sugita, Y.; Okamoto, Y. *Chem. Phys. Lett.* **1999**, *314*, 141–151.

(32) Bowers, K. J.; Chow, E.; Xu, H.; Dror, R. O.; Eastwood, M. P.; Gregersen, B. A.; Klepeis, J. L.; Kolossváry, I.; Moraes, M. A.; Sacerdoti, F. D.; Salmon, J. K.; Shan, Y.; Shaw, D. E. Novel Algorithms for Scalable Molecular Dynamics Simulations on Commodity Clusters. In *Proceedings of the ACM/IEEE Conference on Supercomputing (SC06)*; ACM/IEEE Conference on Supercomputing (SC06), Tampa, FL, November 11−17, 2006; ACM Press: New York, 2006.

(33) Flyvbjerg, H.; Petersen, H. G. *J. Chem. Phys.* **1989**, *91*, 461–466.

(34) Kaminski, G.; Friesner, R. A.; Tirado-Rives, J.; Jorgensen, W. L. *J. Phys. Chem. B* **2001**, *105*, 6474–6487.

(35) Maragakis, P.; Lindorff-Larsen, K.; Eastwood, M. P.; Dror, R. O.; Klepeis, J. L.; Arkin, I. T.; Jensen, M. Ø.; Xu, H.; Trbovic, N.; Friesner, R. A.; Palmer, A. G., III.; Shaw, D. E. *J. Phys. Chem. B* **2008**, *112*, 6155–6158.

(36) Cornilescu, G.; Marquardt, J. L.; Ottiger, M.; Bax, A. *J. Am. Chem. Soc.* **1998**, *120*, 6836–6837.

(37) Lindahl, E.; Hess, B.; van der Spoel, D. *J. Mol. Mod.* **2001**, *7*, 306–317.

(38) Berendsen, H. J. C.; Postma, J. P. M.; van Gunsteren, W. F.; Hermans, J. Interaction models for water in relation to protein hydration. In *Intermolecular Forces*; Pullman, B., Ed.; D. Reidel Publishing Company: Dordrecht, The Netherlands, 1981.

(39) Darden, T.; York, D.; Pedersen, L. *J. Chem. Phys.* **1993**, *98*, 10089–10092.

(40) Kräutler, V.; Van Gunsteren, W. F.; Hünenberger, P. H. *J. Comput. Chem.* **2001**, *22*, 501–508.

(41) Lippert, R. A.; Bowers, K. J.; Dror, R. O.; Eastwood, M. P.; Gregersen, B. A.; Klepeis, J. L.; Kolossváry, I.; Shaw, D. E. *J. Chem. Phys.* **2007**, *126*, 046101.

(42) Tuckerman, M. E.; Berne, B. J.; Martyna, G. J.; Klein, M. L. *J. Chem. Phys.* **1993**, *99*, 2796–2808.

(43) Mazur, A. K. *J. Comput. Phys.* **1997**, *136*, 354–365.

(44) Feenstra, K. A.; Hess, B.; Berendsen, H. J. C. *J. Comput. Chem.* **1999**, *20*, 786–798.

(45) Shaw, D. E. et al. Anton, a special-purpose machine for molecular dynamics simulation. In *Proceedings of the 34th Annual International Symposium on Computer Architecture (ISCA '07)*; 34th Annual International Symposium on Computer Architecture (ISCA '07), San Diego, CA, June 9−13, 2007ACM Press: New York, 2007.

(46) Case, D. A. *Curr. Opin. Struct. Biol.* **1994**, *4*, 285–290.

(47) Oda, K.; Miyagawa, H.; Kitamura, K. *Mol. Simul.* **1996**, *16*, 167–177.

(48) Pastor, R. W.; Brooks, B. R.; Szabo, A. *Mol. Phys.* **1988**, *65*, 1409–1419.

CT9002916