On the use of mass scaling for stable and efficient simulated tempering with molecular dynamics*

Tetsuro NAGAI†, George A. PANTELOPULOS‡,
Takuya TAKAHASHI§, John E. STRAUB¶

November 23, 2016

Abstract

Simulated tempering (ST) is a generalized-ensemble algorithm that employs trajectories exploring a range of temperatures in order to effectively sample rugged energy landscapes. When implemented using the molecular dynamics (MD) method, ST can require the use of short time steps for ensuring the stability of trajectories at high temperatures. To address this shortcoming, a mass-scaling ST (MSST) method is presented in which the particle mass is scaled in proportion to the temperature. Mass scaling in the MSST method leads to velocity distributions that are independent of temperature and eliminates the need for velocity scaling after the accepted temperature updates that are required in conventional ST simulations. The homogeneity in time scales with changing temperature improves the stability of simulations and allows for the use of longer time steps at high temperatures. As a result, the MSST is found to be more efficient than the standard ST method, particularly for cases in which a large temperature range is employed.

Keywords: generalized-ensemble algorithm, simulated tempering (ST), mass scaling, Nosé-Hoover thermostat, molecular dynamics ■

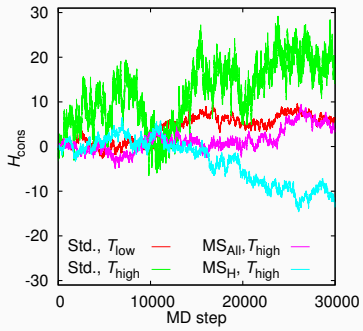
*Final version for *J. Comp. Chem.* **37**, 2017–2028 (2016); DOI: 10.1002/jcc.24430

†Department of Bioinformatics, College of Life Sciences, Ritsumeikan University, Kusatsu, Shiga 525-8577, Japan; present affiliation is Department of Physics, Graduate School of Science, Nagoya University, Nagoya, Aichi 464-8602, Japan; correspondence: tnagai@nagoya-u.jp

‡Department of Chemistry, Boston University, Boston, Massachusetts 02215, USA

§Department of Bioinformatics, College of Life Sciences, Ritsumeikan University, Kusatsu, Shiga 525-8577, Japan

¶Department of Chemistry, Boston University, Boston, Massachusetts 02215, USA



Simulated tempering (ST) is a commonly practiced generalized-ensemble algorithm that combines simulations at multiple temperatures. To perform efficient and stable ST simulation using molecular dynamics, the mass-scaling ST method is presented. The proper mass scaling employed in the new method makes the velocity distributions independent of temperature. This homogeneity enables simple temperature updates and makes the tempering simulation stable at high temperatures. The MSST method is applied to the folding of the Trpcage peptide over a wide range of temperature to demonstrate the gain in numerical stability.

INTRODUCTION

Performing efficient and stable molecular simulations is an indispensable task for understanding biological systems. The generalized-ensemble algorithms have been developed and applied to biological systems (for reviews, see, e.g., Refs. 1–4) to overcome insufficient sampling of conventional molecular dynamics (MD) and Monte Carlo (MC) simulations. Commonly used generalized-ensemble algorithms include simulated tempering (ST)[5,6], the replica-exchange method (REM)[7,8] (see also Refs. 9,10), and the multicanonical algorithm (MUCA)[11,12]. Closely related to the MUCA are the Wang-Landau method[13,14], statistical temperature methods[15–17], and metadynamics[18]. The ST method considers the temperature as a dynamical variable in addition to the (microscopic) state of the system. As the temperature is updated in the ST simulations, a random walk in the temperature space occurs, whereby the random walk in the energy space is realized and the energy barriers are overcome. Further developments of these generalized-ensemble algorithms have been intensively sought, including a multi-dimensional extension[19–24], and the combination of Tsallis statistics[25] with ST (see, e.g., Ref. 26) as well as the generalized REM (see, e.g., Ref. 27–29). The MD version of MUCA was developed in Refs. 30,31. The key to the MD version of REM, which is referred to as replica-exchange molecular dynamics (REMD)[32], was the scaling of the velocity after the replica-exchange attempt was accepted. A similar approach was also useful in the MD version of ST (see e.g. Sec. IIC in Ref. 20).

Although ST and REM are similar in that they fully utilize the combinations of the Boltzmann distributions at several temperatures, there are critical differences. The ST method requires only one replica, whereas the REM requires an increasing number of replicas for a larger system. When an excessive number of replicas prevents use of the REM, the ST method would be a powerful tempering method. Another difference is the weight determination. While the REM does not require any weight determination, free energy estimates at each temperature are necessary in ST. Even though it can be cumbersome to obtain these values, some promising methods have recently been presented by Pande and coworkers[33] and Nguyen et al[34].

This work presents a novel algorithm that significantly enhances the MD version of ST simulations. MD simulations become numerically less stable at increasing temperatures due to the short timescales involved in high energy collisions that demand a short time step. To address this issue and stabilize REMD simulations, two of the authors recently proposed the mass-scaling REMD (MSREMD) method[35,36], based on the multi-dimensional REM[37] and the mass scaling that has many applications[38–45] including QM/MM[46] and *ab initio* MD simulations[47]. In the MSREMD method, the scaling of mass in proportion to the temperature produces an identical velocity distribution among temperatures. The homogeneity makes it feasible to exchange replicas without velocity scaling and to stabilize simulations.

In this paper, adopting the concept of MSREMD to ST, we present a mass-scaling ST (MSST) method in which the mass values are set in proportion to the temperature. We demonstrate that mass scaling is a powerful and convenient tool within a certain class of the generalized-ensemble paradigm. The velocity distribution realized by the mass scaling makes it feasible to update the temperature without velocity scaling, which is necessary after every accepted temperature update in conventional ST simulations. Introducing a more general

framework, we also demonstrate that the mass scaling can be applied to a subset of systems. The usefulness of mass scaling at high temperatures is also confirmed in application to a simple fluid, a model lipid system and the folding of model peptide, Trpcage[48], in the context of ST simulation.

This article is organized as follows. In Methodology, the ST method is briefly reviewed in the MD context and the MSST method is introduced. We show that the MSST method preserves the numerical stability of the simulation at high temperatures in applications to a simple fluid. After comparison of the physical quantities obtained with the ST and MSST simulations, we compare (integrated) autocorrelation time as indicator of efficiency. Following this, we establish the equivalence between mass scaling and time step adjustment for Nosé-Hoover thermostats. The utility of mass scaling is further discussed based on the results of a more complex system. Finally, we discuss the sampling efficiency of MSST via an application of protein folding.

METHODOLOGY

Review of the ST method

Before examining the MSST method, we briefly review the original ST method[5,6] in the context of MD simulation. Let us consider an N-particle system, of which the velocity and coordinate vectors are given by $\dot{q} = \{\dot{\boldsymbol{q}}_1, \ldots, \dot{\boldsymbol{q}}_N\}$ and $q = \{\boldsymbol{q}_1, \ldots, \boldsymbol{q}_N\}$, respectively. For simplicity, the momentum is assumed to be $\boldsymbol{p}_k = m_k \dot{\boldsymbol{q}}_k$, where m_k is the mass of the kth particle. The total energy of the system is given by

$$H = K + E, \tag{1}$$

where K and E are the kinetic and potential energies, respectively. The kinetic energy is given by

$$K(\dot{q}) = \sum_{k=1}^{N} \frac{m_k \dot{\boldsymbol{q}}_k^2}{2}. \tag{2}$$

In ST simulations, the temperature is considered as a dynamical variable that is updated during simulations. The temperature can assume one of M temperature values of T_1, T_2, $\ldots$, T_M (i.e., the temperature space is discretized). In other words, the Boltzmann factor $e^{-H/k_{\mathrm{B}}T}$ is considered as an unnormalized joint probability of microscopic state and temperature. The weight factor of the ST simulation is given by

$$\Pi_{\mathrm{ST}}(q, \dot{q}, T_i) = \exp[-(K(\dot{q}) + E(q))/k_{\mathrm{B}}T_i + a(T_i)], \tag{3}$$

where $a(T_i)$ is the parameter introduced to obtain the flat distribution in the temperature space. In the MD version of the ST simulation, updates of q and $\dot{q}$ are performed at a fixed temperature T_i by canonical MD simulation, and temperature updates are performed by the MC method with q and $\dot{q}$ fixed.

Let us consider the transition probability from T_i to T_j: the original state is denoted by $X = \{q, \dot{q}, T_i\}$ and the candidate by $X' = \{q, \dot{q}', T_j\}$, where the velocity $\dot{q}'$ is scaled[20] as

$$\dot{q}' = \sqrt{\frac{T_j}{T_i}}\dot{q}, \tag{4}$$

to cancel an unhelpful contribution from the kinetic energy to the transition probability. Based on the Metropolis criteria[49], the transition probability is given by

$$\begin{aligned} &w\left(X \to X'\right) \\ &\quad = \min\left[1, \exp\left[-\left(\frac{1}{k_{\mathrm{B}}T_j} - \frac{1}{k_{\mathrm{B}}T_i}\right)E(q) + a(T_j) - a(T_i)\right]\right]. \end{aligned} \tag{5}$$

Taking account of the velocity scaling (see Eq. 4), the configuration dependent contribution to the dimensionless free energy

$$f_i = -\ln \int \exp\left[-\frac{E(q)}{k_{\mathrm{B}}T_i}\right] dq \tag{6}$$

can be shown to be a proper choice for $a(T_i)$ to ensure that a flat distribution over the temperature is obtained.

The thermodynamic average at T_i of coordinate-related quantities, which are usually of interest, can be obtained in the standard way

$$\begin{aligned} \langle A \rangle_{T_i} &= \frac{\int A(q) \exp\left[-E(q)/k_{\mathrm{B}}T_i\right] dq}{\int \exp\left[-E(q)/k_{\mathrm{B}}T_i\right] dq} \\ &= \langle A(q)|T_i \rangle_{\mathrm{ST}}\,, \end{aligned} \tag{7}$$

which yields

$$\langle A \rangle_{T_i} = \frac{1}{N_i}\sum_{j}^{N_i} A(q_{ij}), \tag{8}$$

where q_{ij} is the coordinate of the jth sample at T_i and N_i the number of samples at T_i.

MSST method

The key concept of the MSST method is that the mass is scaled in proportion to the reference temperature T_i. In the simplest form, the mass value of the kth particle at T_i is given by

$$m_k(T_i) = m_{0,k}\frac{T_i}{T_1}, \tag{9}$$

where $m_{0,k}$ is the mass value of the kth particle at the lowest temperature and the kinetic energy at T_i is

$$K_i(\dot{q}) = \sum_{k=1}^{N} \frac{m_{0,k}\frac{T_i}{T_1}\dot{\boldsymbol{q}}_k^2}{2}. \tag{10}$$

The weight factor of the MSST simulation is given by

$$\begin{aligned} \mathit{\Pi}_{\mathrm{MSST}}(q,\dot{q},T_i) &= \exp\left[-\left(K_i(\dot{q})+E(q)\right)/k_{\mathrm{B}}T_i + a(T_i)\right] \\ &= \exp\left[-\left(\sum_{k=1}^{N}\frac{m_{0,k}\dot{\boldsymbol{q}}_k^2}{2k_{\mathrm{B}}T_1}\right) - E(q)/k_{\mathrm{B}}T_i + a(T_i)\right], \end{aligned} \tag{11}$$

which demonstrates that the velocity distributions are independent of temperature. In contrast, as an increase in mass in the canonical ensemble generates the larger momenta, the momenta are described by a different probability distribution function. The homogeneity in MSST enables the temperature to be updated without velocity scaling and stabilizes the simulation at higher temperatures.

The transition probability of MSST can be obtained by the Metropolis criterion[49] without velocity scaling:

$$\begin{aligned} w(\{q,\dot{q},T_i\}\rightarrow\{q,\dot{q},T_j\}) &= \min\left[1,\frac{\mathit{\Pi}_{\mathrm{MSST}}(q,\dot{q},T_j)}{\mathit{\Pi}_{\mathrm{MSST}}(q,\dot{q},T_i)}\right] \\ &= \min\left[1,\frac{\exp[-\left(\sum_{k=1}^{N}\frac{m_{0,k}\dot{\boldsymbol{q}}_k^2}{2T_1}\right) - E(q)/k_{\mathrm{B}}T_j + a(T_j)]}{\exp[-\left(\sum_{k=1}^{N}\frac{m_{0,k}\dot{\boldsymbol{q}}_k^2}{2T_1}\right) - E(q)/k_{\mathrm{B}}T_i + a(T_i)]}\right] \\ &= \min\left[1,\exp\left[-\left(\frac{1}{k_{\mathrm{B}}T_j}-\frac{1}{k_{\mathrm{B}}T_i}\right)E(q) + a(T_j) - a(T_i)\right]\right]. \end{aligned} \tag{12}$$

In this way, the standard transition probability of ST, Eq. 5, is recovered without velocity scaling.

Similar to the original ST method, to make the distribution of samples with respect to the temperature ($P_{\mathrm{S}}(T_i)$) flat, i.e., $P_{\mathrm{S}}(T_i) \propto \int d\dot{q}\, dq\, \mathit{\Pi}_{\mathrm{MSST}}(q,\dot{q},T_i) = \text{const.}$, the configuration part of the free energy f_i is shown to be a suitable choice for $a(T_i)$. As the physical quantities of interest are usually not a function of velocities, the average of any coordinate-related quantities are identical in the ST and MSST. The average kinetic energies are also the same between the two methods, because the average kinetic energy is only a function of the temperature.

The above formalism is sufficient for the Langevin[50] and Andersen[51] thermostats. We note that, as the Nosé-Hoover thermostat[52,53] involves additional variables, the terms originated from them must be added in the interest of precision. Assuming ergodicity, the Nosé-Hoover thermostat with the reference temperature T realizes the probability density function of $\{q,\dot{q},\eta\}$:

$$\begin{aligned} &f_{\mathrm{NH}}(q,\dot{q},\eta) \\ &\propto \exp\left[-H(q,\dot{q})/k_{\mathrm{B}}T\right]\exp\left[-Q\eta^2/2k_{\mathrm{B}}T\right], \end{aligned} \tag{13}$$

where η and Q represent the rate and mass, respectively. Integrating the probability density function with respect to η, one obtains the canonical distribution with regard to $\{q,\dot{q}\}$. To obviate the contribution of η to the temperature-updating attempt in ST simulations, the scaling of η is necessary, similarly to the REMD[54], which is given by

$$\eta' = \sqrt{\frac{T_j}{T_i}}\eta \tag{14}$$

for the update from T_i to T_j.

Nevertheless, similar to the velocities, the scaling of the rate of Nosé-Hoover thermostats can be eliminated by setting $Q = (T_i/T_1)Q_0$. Substituting $m_k = (T_i/T_1)m_{0,k}$ and $Q = (T_i/T_1)Q_0$ in Eq. 13, the weight factor of the MSST simulation with Nosé-Hoover thermostats is given by

$$\begin{aligned} \Pi_{\text{NH-MSST}}(q,\dot{q},\eta,T_i) &= \exp\left[-\left(K_i(\dot{q})+E(q)\right)/k_{\text{B}}T_i + a(T_i)\right]\exp\left[-Q_0\eta^2/2k_{\text{B}}T_1\right] \\ &= \exp\left[-\left(\sum_{k=1}^{N}\frac{m_{0,k}\dot{\boldsymbol{q}}_k^2}{2k_{\text{B}}T_1}\right) - E(q)/k_{\text{B}}T_i + a(T_i)\right]\exp\left[-Q_0\eta^2/2k_{\text{B}}T_1\right]. \end{aligned} \tag{15}$$

Note that the distributions of $\dot{q}$ and η are independent of T_i.

In the general case, the mass of kth particle $m_{0,k}$ is scaled by arbitrary factor $\alpha_{k,i}$ at T_i. Then the kinetic energy at T_i is given by

$$K_i(\dot{q}) = \sum_{k=1}^{N}\frac{\alpha_{k,i}m_{0,k}\dot{\boldsymbol{q}}_k^2}{2}. \tag{16}$$

Temperature updates from $\{q,\dot{q},T_i\}$ to $\{q,\dot{q}',T_j\}$ may be made using Eq. 5 with the velocity scaling of

$$\dot{\boldsymbol{q}}_k' = \sqrt{\frac{\alpha_{k,i}}{\alpha_{k,j}}}\sqrt{\frac{T_j}{T_i}}\dot{\boldsymbol{q}}_k. \tag{17}$$

In summary, an MSST simulation can be performed as follows: (1) prepare an initial condition; (2) perform the canonical MD simulation at the current temperature; (3) attempt to update the temperature and mass values according to the probability given by Eq. 5 with Eq. 17; and (4) repeat steps (2) and (3) until the simulation ends.

Equations of motion for mass-scaled system

In the general case, the MSST method allows for the use of a variety of scaling factors. This can be of great utility in the simulation of systems with particles of widely varying mass. A special case of the MSST method involves the use of a single, uniform mass scaling factor. In that case, the mass scaling in the equation of motion of the Nosé-Hoover thermostat[52,53] is mathematically identical to changing the time step[35]. The equations of motion are given by

$$\frac{dq}{dt} = \frac{p}{m}, \tag{18}$$

$$\frac{dp}{dt} = F(q) - \eta p, \tag{19}$$

$$\frac{d\eta}{dt} = \frac{p^2/m - 3Nk_{\text{B}}T}{Q}, \tag{20}$$

where F denotes force and t is the time. For simplicity, we set $m_1 = m_2 = \cdots = m_N \equiv m$. These four equations are invariant under the transformation given by

$$m = \alpha m', \tag{21}$$
$$t = \sqrt{\alpha} t', \tag{22}$$
$$Q = \alpha Q', \tag{23}$$
$$q = q', \tag{24}$$
$$\eta = \frac{1}{\sqrt{\alpha}} \eta', \tag{25}$$

along with

$$\frac{dq}{dt} = \frac{1}{\sqrt{\alpha}} \frac{dq'}{dt'}, \tag{26}$$
$$p = \sqrt{\alpha} p', \tag{27}$$

where α denotes the scaling factor. Therefore, the time step $\Delta t'$ of a system with mass value m' corresponds to the time step

$$\Delta t = \sqrt{\alpha} \Delta t' \tag{28}$$

of a system with the mass value $m = \alpha m'$.

Based on the time evolution of a mass-scaled system, we examine the equivalence between the MSST simulation and a corresponding time-step-adjusting ST (TSA-ST) simulation. The TSA-ST simulation is defined as a conventional ST simulation with a varying time step

$$\Delta t^{\mathrm{TSA}}(T_i) = \frac{\Delta t^{\mathrm{MSST}}}{\sqrt{T_i/T_1}}, \tag{29}$$

where Δt^{MSST} is the time step of the equivalent MSST simulation. Further discussion of the equivalence focusing on the temperature-updating attempt is provided in the Appendix.

In order for the primed quantity to correspond to MSST, we set α to $1/(T_i/T_1)$. In other words, m' is (T_i/T_1)-times as heavy as m and $\Delta t'$ of the heavier system corresponds to $\Delta t = \Delta t'/\sqrt{T_i/T_1}$ with the normal mass m. We thus define an effective time step of MSST at T_i by

$$\Delta t_{\mathrm{eff}}(T_i) = \frac{\Delta t^{\mathrm{MSST}}}{\sqrt{T_i/T_1}}. \tag{30}$$

Equation 24 suggests that the MSST simulation with the time step Δt^{MSST} generates the same time evolution of the coordinate vector as the ST simulation with the time step $\Delta t_{\mathrm{eff}}(T_i)$ at T_i. We therefore refer to the ST simulation with the time step $\Delta t^{\mathrm{TSA}}(T_i) = \Delta t_{\mathrm{eff}}(T_i)$ as the TSA-ST simulation, the coordinate vector of which is equivalent to that of an MSST simulation with Δt^{MSST}, i.e.,

$$q^{\mathrm{MSST}} = q^{\mathrm{TSA}}. \tag{31}$$

In contrast to the position, the evolution of the velocity (or momentum) vector and rate of the Nosé-Hoover thermostat are reproduced up to a proportionality constant. For example, the velocity and rate of the Nosé-Hoover thermostat between the ST and MSST simulations can be related by

$$\frac{1}{\sqrt{T_i/T_1}}\dot{q}^{\mathrm{TSA}} = \dot{q}^{\mathrm{MSST}}, \tag{32}$$

$$\frac{1}{\sqrt{T_i/T_1}}\eta^{\mathrm{TSA}} = \eta^{\mathrm{MSST}}. \tag{33}$$

Application to a Lennard-Jones fluid

We employed a pairwise additive 12-6 Lennard-Jones (LJ) fluid as a useful test system. Hereafter, values are reported in the reduced unit, in which the well depth ϵ and the diameter σ are set to unity.

Free energy calculations

As mentioned above, the ST and MSST methods require free energy estimates as part of their input. We applied the MBAR[55], which is the equivalent of WHAM[56–58] in the limit that the bin width tends to zero, to the results of MSREMD simulations performed in a previous study[35]. Once the free energy is obtained, the reweighting techniques enable the thermal averages to be obtained at an arbitrary temperature at which simulations are not performed.

Numerical details

An in-house time-reversible integrator[59], which employed the Suzuki-Trotter decomposition corresponding to Integrator 1 in Ref. 60, was used. The Mersenne twister pseudo-random number generator[61] was employed.

In all simulations unless otherwise mentioned, eight temperatures, referred to as T_1 to T_8, are used, as in our previous work[35]: 1.000, 1.104, 1.219, 1.346, 1.486, 1.641, 1.812, and 2.000. As in the previous study, five hundred identical LJ particles were placed ($N = 500$) in a cube with a side length of 8.55 in the reduced length unit, using periodic boundary conditions and corresponding to a number density of $\rho = 0.800$. The LJ fluid is in the liquid phase[62] under these thermal conditions.

The particle mass was set to unity regardless of the temperature in the ST and TSA-ST simulations. In these simulations, the mass of the Nosé-Hoover thermostat was always 10, and η was scaled according to Eq. 14. In the MSST simulations, the mass at T_i was given by T_i/T_1 and that of the Nosé-Hoover thermostat was set to $Q_0 = 10$. The temperature-updating attempts were performed at intervals of 10 MD steps. The choice between two neighboring temperatures was made at random. The LJ forces were simply truncated at $r_c = 3$, and accordingly the LJ potential was shifted upward for $r_{ij} < r_c$.

Systems were prepared from a random initial configuration and equilibrated with the 8×10^6-step MSST simulation. We performed four simulations, namely ST with a long time step 0.01 (hereafter, referred to as LTS-ST or LTS), ST with a short time step $0.01/\sqrt{T_8/T_1}$

(STS-ST or STS), MSST with a time step 0.01, and corresponding TSA-ST simulations. These four simulations lasted for 8×10^7 steps. The time step in the STS-ST simulation represents the short one chosen for enhanced stability at high temperatures. In the TSA-ST simulation, the time step is set to $\Delta t = 0.01/\sqrt{T_i/T_1}$ at T_i (see Eq. 30), and other details are the same as for the LTS-ST simulation. We also performed another set of simulations, in which 25 temperatures are involved ranging 1 to 10 with exponential spacing. This wide range of temperature may sound somewhat extreme but is not unrealistic. For example, Ref. 63 employed the temperature range of 500 K to 5000 K during the replica-exchange MC simulations for an all-atom protein in an implicit membrane model. In addition, a short MSST simulation with $\Delta t = 0.0005$ and a TSA-ST simulation, of which the time step is given by $\Delta t = 0.0005/\sqrt{T_i/T_1}$ at T_i, are performed to demonstrate the equivalence between the two methods.

Evaluation of numerical instability

We evaluated the numerical instability using the trajectory inaccuracy in the following way[35,59]. The conserved quantity for the Nosé-Hoover MD simulation with the reference temperature $T_{\rm ref}$ is given by

$$E_{\rm cons}(t) = \sum_k^N \frac{\boldsymbol{p}_k^2}{2m_k} + E(q) + \frac{1}{2}Q\eta^2 + 3Nk_{\rm B}T_{\rm ref}\int_0^t \eta(t')dt'. \tag{34}$$

In practice, this quantity fluctuates, as a reflection of numerical errors. The average of the absolute fluctuation per MD step is defined as the trajectory inaccuracy

$$\Delta E_{\rm cons} \equiv \frac{1}{N_{\rm step}} \sum_{i=1}^{N_{\rm step}} |E_{\rm cons}(i\Delta t) - E_{\rm cons}((i-1)\Delta t)|, \tag{35}$$

where $N_{\rm step}$ is the number of MD steps. We ascribe the decrease in the numerical stability to the increase in $\Delta E_{\rm cons}$.

Application to model lipid bilayer

To further evaluate the effects of mass scaling on trajectory stability, we performed canonical simulations of a model lipid system using the solvent-free three-bead lipid model of Cooke et al[64,65]. We used parameters depicted in Ref. 65. For the simplicity, the unit length of σ and the unit energy ϵ were set to unity as well as $k_{\rm B}$. This model combines a hard core Weeks-Chandler-Andersen potential, a bond potential that has a finite divergence length, a bending potential, and an effective hydrophobic attraction. In our simulations, the box dimension was fixed at $17.6\,\sigma$ and the number of lipids was 512. A binary lipid mixture was simulated with 128 A lipids and 384 B lipids, and the decay ranges of the hydrophobic attraction between A ($w_{\rm AA}$), between B ($w_{\rm BB}$) are set to 1.6, while that between A and B ($w_{\rm AB}$) was set to 1.48. The system forms a bilayer with phase separations at $T = 1.1$[65,66].

The temperature was set to 1.1, 1,4, 2.0, or 4.0 with the time step Δt being 0.005, 0.0075, 0.01, or 0.015. The moderate time step of 0.01 is the value reported in Ref. 65 for simulations at $T = 1.1$. The time step of 0.015 is considered to be an ambitious time step. In standard simulations, the mass was set to unity. In mass-scaling simulations, the mass was set in proportion to the temperature with the mass unity at $T = 1.1$. We performed or tried to perform 1×10^5-step simulations for all combinations of the above temperatures and time steps.

Application to Trpcage

In order to evaluate the utility of mass scaling, we performed standard and mass scaling canonical MD simulations of Trpcage[48] with a time step of 0.5 and 1.0 fs, using the GPU-accelerated OpenMM[67–69]. The temperature was kept at 273 K or 819 K with Langevin dynamics with the friction coefficient of 1.0 ps^{-1} . For the simulation at the higher temperature, we tested two mass scaling schemes. In the first scheme, each atom mass was multiplied by $819/273 = 3$. In the other scheme, only the hydrogen mass was multiplied by 3 with other masses kept at their original values. We evaluated the stability of Langevin dynamics trajectories in the spirit of Eq. 20 in Ref. 70. In this work, the leading term of the increment of the conserved quantity

$$\Delta H_{\mathrm{cons}} = \Delta q \left(\frac{f(t) + f(t + \Delta t)}{2} \right) + \Delta E \tag{36}$$

was used, where $\Delta q = q(t + \Delta t) - q(t)$, $\Delta E = E(t + \Delta t) - E(t)$ and $f(t)$ represents the force. A modification of ff99SB previously employed by Simmerling et al. to the study of the folding of Trpcage[71] with GBOBC2[72] was used to parametrize these simulations. The solute and solvent dielectrics were set to 1.0 and 78.5, respectively. Note that, as Langevin dynamics was employed, the exact correspondence between TSA-ST and MSST discussed above with Nosé-Hoover thermostat is not guaranteed.

Additionally, we performed the standard ST and MSST simulations. In the MSST simulation, only the hydrogen mass was scaled. The time step was set to 0.5 fs in the standard ST simulation and to 1.0 fs in the MSST simulation. Exchange attempts were made every 50 ps with the Gibbs sampler scheme[73]. Simulations lasted for 30 ns. Five replicates of simulations starting with the extended structure were performed for each.

In this application, we obtained separate sets of weights for ST and MSST simulations, as, considering $\partial f / \partial T = -\langle E \rangle / k_{\mathrm{B}} T^2$, the shift in the average potential energy due to the varying time step[45,60,70] can significantly affect the weights. We first performed three simulations with the Wang-Landau approach for ST and MSST so as to obtain approximate samples at each temperature. The first rough estimates were obtained by application of MBAR[55] to these results. We then performed three replicates of simulations with these fixed weights. Observing the random walk in these simulations, we further refined the weights with the application of MBAR. These refined weights were used for the final production runs reported below. The weights obtained were significantly different between the two simulations.

RESULTS AND DISCUSSION

LJ fluid simulations

We first confirm that the probability density function for velocity is independent of temperature for MSST and dependent on temperature for all other methods (Figure 1). This was performed by evaluating the probability distribution function of velocity in the x-dimension for LTS-ST, STS-ST, MSST, and TSA-ST simulations, taking the velocity at half time steps as per the recommendation in Ref. 60. The probability distributions for visiting each temperature were nearly flat (Table 1), having a slight discrepancy from $P_{\mathrm{S}}^{(\mathrm{perfect})}(T_i) = 0.125$, a perfectly flat distribution, whereby the random walk in temperature was realized[74]. Considering the autocorrelation time[75–77] (see, in particular, Eq. (4.27) in Ref. 76) of the temperature index, the errors roughly exceed 0.001, thereby indicating that $P_{\mathrm{S}}(T_i)$ is the same between all simulations.

The numerical instability in terms of trajectory inaccuracy, ΔE_{cons} (Eq. 35), at each temperature observed in these four simulations shows that ST simulations become substantially more unstable than MSST simulations as temperatures increase (Figure 2). The slight increase in inaccuracy with rising temperature in the MSST simulation could be attributed to the steeper potential energy surfaces encountered at higher potential energy values. In short, we infer that the MSST method enables us to perform more stable ST simulations with a long time step, because the trajectory inaccuracy is substantially reduced by this method at high temperature.

To further examine the stability, we examine the time series of instantaneous temperatures ($2K/3N$ or $2K_i/3N$) in simulations employing 25 reference temperatures. As shown in Fig. 3 the LTS-ST simulation with the time step of 0.01 exhibits unreasonable behavior from around the 8180th MD step, while the MSST and too careful STS-ST simulations were stable over 10^7 steps. This result demonstrates the superior stability of the MSST method.

The agreement between the physical quantities of the four methods is confirmed in Tables 2 and 3 that list the average kinetic and potential energies, respectively, for all the reference temperatures. In addition, the radial distribution functions were also found in agreement between the four methods [data not shown].

As an indicator of the efficiency of the simulations, we obtained the autocorrelation time (τ) of the potential energy for the four simulations (Figiure 4). In this regard, the binning method[75,76] was employed and, in particular, τ was calculated based on Eq. (4.27) in Ref. 76. In this calculation, the unit of τ is the number of samples and an increase in its value can be interpreted as a reduction in the efficiency of the simulations[75,76]. In other words, the shorter autocorrelation time suggests the faster simulation. The energy autocorrelation times in the MSST an TSA-ST simulations are in relative agreement with each other, indicating the equivalence in coordinates between the two methods. Additionally, the value of the autocorrelation time in MSST coincided with that of LTS-ST at lowest temperature (Figiure 4), demonstrating the similar efficiency of MSST to LTS-ST at the lowest temperature, at which the physical quantities are usually of most interest.

Figure 5 shows the scaled autocorrelation time τ for the potential energy versus T. The scaling was performed according to the ratios in the real or effective time steps of each simulation to the LTS-ST simulation. The effective time step of the MSST method is

defined in Eq. 30. For the LTS-ST, STS-ST, MSST, and TSA-ST simulations, the factors are 1, $\frac{1}{\sqrt{T_8}}(=\frac{1}{\sqrt{2}})$, $\frac{1}{\sqrt{T_i/T_1}}$, and $\frac{1}{\sqrt{T_i/T_1}}$, respectively. Figure 5 clearly indicates that $\tau_i^{E,\mathrm{LTS}}$, $(1/\sqrt{2})\tau_i^{E,\mathrm{STS}}$, $(1/\sqrt{T_i})\tau_i^{E,\mathrm{MSST}}$, and $(1/\sqrt{T_i})\tau_i^{E,\mathrm{TSA}}$ are in agreement. This demonstrates that the autocorrelation time merges into one master curve by scaling, showing universal behavior as a function of temperature.

Figure 6 shows the autocorrelation time of potential energy for the LJ fluid simulation employing 25 temperatures. The difference in the efficiency between the MSST and STS-ST simulation as measured by τ_1^E is approximately as large as a factor of 3. Recall that in this case the LTS-ST simulation was unstable. Additionally, the agreement between the scaled values (black diamonds) and values of MSST (green crosses) supports the validity of the scaling argument illustrated in Fig. 5. This result establishes that the MSST method is superior to the conventional ST method, especially when the temperature range is wide.

We examine the equivalence between MSST and TSA-ST simulations based on LJ fluid simulations with a very short time step. Both of the simulations were initiated at T_1, where they use the identical mass and time step, with the same initial conditions and random number seed. Figure 7 illustrates the x-component velocity of the particle as a function of the time step. Even though the two methods have different values for their velocities, the properly scaled value $v_x^{\text{TSA-ST}}/\sqrt{T_i/T_1}$, corresponding to Eq. 32, coincides with the value of v_x^{MSST}. Similarly it is clearly shown that $\eta^{\text{STA-ST}}/\sqrt{T_i/T_1}$, which corresponds to Eq. 33, merges into the curve of η^{MSST} (Figure 8). These figures clearly establish the relationship between the MSST and TSA-ST simulations. In addition, both v_x and η have smaller values at high temperatures in MSST than in TSA-ST simulations. This indicates that the MSST simulation has slower dynamics at high temperatures, whereby the simulations tend to be more stable with the fixed long time step.

Solvent-free model lipid bilayer simulations

All simulations of the model lipid bilayer employing mass scaling were stable; however, some of the simulations without mass scaling exhibited unstable behavior (see Table 4). As the time step of 0.01 is the value reported in Ref. 65 for simulations at $T = 1.1$, this time step should be a moderate time step. The time step of 0.015 is an ambitious time step, with which the integration can be performed at $T = 1.1$ with a small monotonic drift in E_{cons} of around 5 % at 10^5 MD steps (data not shown). The time steps of 0.005 and 0.0075 are chosen as sober time steps. The table shows that, even with the time step of 0.0075, the simulation at $T = 4.0$ is unstable, indicating that the time step of 0.005 is necessary for the conventional ST method. With the ambitious time step of 0.015, the normal ST simulation breaks down even at $T = 1.4$. On the other hand, the mass-scaling simulations with time step of 0.015 stable over the wide range of temperature, although a small drift in E_{cons} was observed as found at $T = 1.1$.

Figures 9 and 10 illustrate the conserved quantity of Nosé-Hoover thermostats normalized by its initial value, $E_{\mathrm{cons}}/E_{\mathrm{cons}}(0)$, for the normal and mass scaling simulations, respectively, at $T = 2.0$. It is shown that mass scaling makes the integration with the time step of 0.015 feasible, although $E_{\mathrm{cons}}/E_{\mathrm{cons}}(0)$ exhibits a small drift with this large time step (as at $T = 1.1$). The conserved quantity with the time step of 0.01 shows a small yet apparent

drift without mass scaling (see Figure 9). The fact that the mass scaling improved the conservation properties demonstrates that sound tempering simulations are possible with mass scaling.

Protein simulations

With an integration time step of 1.0 fs, simulations at higher temperature exhibit large fluctuations in the conserved quantity (Figure 11). Nevertheless, the mass scaled simulations demonstrate a general decrease in fluctuations when compared to conventional simulation with the 1.0 fs time step. Simulations with scaled hydrogen masses show a significant relief in fluctuations of the conserved quantity, suggesting that the hybrid approach of MSST and ST should be useful for enhanced sampling with increased stability.

The average energy and helix content were evaluated for the ST and MSST with hydrogen mass scaling (Figure 12), which show similar convergence, as the total time is the same. However, ST requires a time step of 0.5 fs compared with a time step of 1 fs using MSST. With the initial value being 14.9 Å, the lowest backbone atom RMSD values to the NMR structure PDB:1L2Y were 1.9, 1.7, 2.0, 1.7, and 2.4 Å in the five ST simulations and were 1.5, 1.6, 1.5, 2.5, and 1.6 Å for the five MSST simulations, indicating that both of these methods successfully folded the peptide.

CONCLUSIONS

The mass-scaling simulated tempering (MSST) method is found to have superior stability relative to the conventional simulated tempering method in MD simulation. The enhanced stability allows for longer integration time steps and more efficient simulations. Applying mass scaling to all particles in proportion to temperature causes the velocity distributions to become identical regardless of temperature. This homogeneity makes the velocity scaling unnecessary upon temperature updates, thus leading to a simple and elegant algorithm. We also show that MSST may be applied to subsets of particles in a system, as is done for hydrogen atoms in the simulation of Trpcage. This approach makes simulation more stable with a minor increase in the system mass, which may be of particular interest to the simulation of biomolecules.

It is possible to implement a more general MSST approach for biomolecular simulation involving the mass scaling in accordance with the atomic species as well as the temperature. The new method would be expected to work well with coarse-grained models as well as atomistic models. Although the efficiency of the method is measured by the autocorrelation time, measurement of ergodicity[77,78] could be examined as an additional measure of efficiency in simulation of complex systems. Finally we note that while our efforts were specifically directed at the canonical ensemble; however, the rigorous formalism and evaluation of the MSST method with other thermostats or ensembles would be straightforward and fruitful.

ACKNOWLEDGMENTS

This work was supported in part by a Grant-in-Aid for Young Scientists (B) under Grant No. 26790083 (to TN). TN gratefully acknowledges the financial support of Ritsumeikan University for his extended stay at Boston Univsersity. JES acknowledges the support of a Japan Society for the Promotion of Science (JSPS) Invitation Fellowship during his stay at Nagoya University and the generous support of the National Science Foundation (CHE-1362524). GAP is supported by a National Science Foundation Graduate Research Fellowship under Grant DGE-1545957. Some of the computations were performed at the Research Center for Computational Science, Okazaki, Japan. NVIDIA corporation is gratefully acknowledged for a GeForce GTX Titan X via an NVIDIA hardware grant which was used to perform simulations of Trpcage.

References

1. U. H. E. Hansmann and Y. Okamoto, in *Annual Reviews of Computational Physics VI*, edited by D. Stauffer (World Scientific, Singapore, 1999), pp. 129–157.

2. A. Mitsutake, Y. Sugita, and Y. Okamoto, Biopolymers **60**, 96 (2001).

3. Y. Sugita and Y. Okamoto, in *Lecture Notes in Computational Science and Engineering*, edited by T. Schlick and H. H. Gan (Springer, 2002), pp. 304–332, http://arxiv.org/abs/cond-mat/0102296.

4. A. Morriss-Andrews and J.-E. Shea, Annual Review of Physical Chemistry **66**, 643 (2015).

5. A. P. Lyubartsev, A. A. Martsinovski, S. V. Shevkunov, and P. N. Vorontsov-Velyaminov, The Journal of Chemical Physics **96**, 1776 (1992).

6. E. Marinari and G. Parisi, Europhysics Letters (EPL) **19**, 451 (1992).

7. C. J. Geyer, in *Computing Science and Statistics, Proceedings of the 23rd Symposium on the Interface*, edited by E. M. Keramidas (Interface Foundation of North America, 1991), pp. 156–163.

8. K. Hukushima and K. Nemoto, Journal of the Physics Society Japan **65**, 1604 (1996).

9. R. H. Swendsen and J.-S. Wang, Physical Review Letters **57**, 2607 (1986).

10. J.-S. Wang and R. H. Swendsen, Progress of Theoretical Physics Supplement **157**, 317 (2005).

11. B. A. Berg and T. Neuhaus, Physics Letters B **267**, 249 (1991).

12. B. A. Berg and T. Neuhaus, Physical Review Letters **68**, 9 (1992).

13. F. Wang and D. P. Landau, Physical Review Letters **86**, 2050 (2001).

14. F. Wang and D. P. Landau, Physical Review E **64**, 056101 (2001).

15. J. Kim, J. E. Straub, and T. Keyes, The Journal of Chemical Physics **126**, 135101 (2007).

16. J. Kim, T. Keyes, and J. E. Straub, The Journal of Chemical Physics **130**, 124112 (2009).

17. J. Kim, J. E. Straub, and T. Keyes, The Journal of Physical Chemistry B **116**, 8646 (2012).

18. A. Laio and M. Parrinello, Proc. Natl. Acad. Sci. USA **99**, 12562 (2002).

19. A. Mitsutake and Y. Okamoto, Physical Review E **79**, 047701 (2009).

20. A. Mitsutake and Y. Okamoto, The Journal of Chemical Physics **130**, 214105 (2009).

21. A. Mitsutake, The Journal of Chemical Physics **131**, 094105 (2009).

22. T. Nagai and Y. Okamoto, Physical Review E **86**, 056705 (2012).

23. T. Nagai, Y. Okamoto, and W. Janke, Journal of Statistical Mechanics: Theory and Experiment **2013**, P02039 (2013).

24. T. Nagai, Y. Okamoto, and W. Janke, Condensed Matter Physics **16**, 23605 (2013).

25. C. Tsallis, Journal of Statistical Physics **52**, 479 (1988).

26. J. Kim and J. E. Straub, The Journal of Chemical Physics **133**, 154101 (2010).

27. J. Kim, T. Keyes, and J. E. Straub, The Journal of Chemical Physics **132**, 224107 (2010).

28. Q. Lu, J. Kim, and J. E. Straub, The Journal of Physical Chemistry B **116**, 8654 (2012).

29. Q. Lu, J. Kim, and J. E. Straub, Journal of Chemical Physics **138** (2013).

30. U. H. Hansmann, Y. Okamoto, and F. Eisenmenger, Chemical Physics Letters **259**, 321 (1996).

31. N. Nakajima, H. Nakamura, and A. Kidera, The Journal of Physical Chemistry B **101**, 817 (1997).

32. Y. Sugita and Y. Okamoto, Chemical Physics Letters **314**, 141 (1999).

33. S. Park and V. Pande, Physical Review E **76**, 016703 (2007).

34. P. H. Nguyen, Y. Okamoto, and P. Derreumaux, The Journal of Chemical Physics **138**, 061102 (2013).

35. T. Nagai and T. Takahashi, The Journal of Chemical Physics **141**, 114111 (2014).

36. T. Nagai and T. Takahashi, Proceedings of Computational Science Workshop 2014 (CSW2014); JPS Conf. Proc. **5**, 011009 (2015).

37. Y. Sugita, A. Kitao, and Y. Okamoto, The Journal of Chemical Physics **113**, 6042 (2000).

38. C. H. Bennett, Journal of Computational Physics **19**, 267 (1975).

39. R. Pomès and J. McCammon, Chemical Physics Letters **166**, 425 (1990).

40. B. Mao, Biophysical journal **60**, 611 (1991).

41. B. Mao, G. M. Maggiora, and K. C. Chou, Biopolymers **31**, 1077 (1991).

42. K. A. Feenstra, B. Hess, and H. J. C. Berendsen, Journal of Computational Chemistry **20**, 786 (1999).

43. P. H. Nguyen, The Journal of Chemical Physics **132**, 144109 (2010).

44. I.-C. Lin and M. E. Tuckerman, The Journal of Physical Chemistry B **114**, 15935 (2010).

45. C. W. Hopkins, S. Le Grand, R. C. Walker, and A. E. Roitberg, Journal of Chemical Theory and Computation **11**, 1864 (2015).

46. H. Zheng, S. Wang, and Y. Zhang, Journal of Computational Chemistry **30**, 2706 (2009).

47. E. Tsuchida, The Journal of Chemical Physics **134**, 044112 (2011).

48. J. W. Neidigh, R. M. Fesinmeyer, and N. H. Andersen, Nature structural biology **9**, 425 (2002).

49. N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, The Journal of Chemical Physics **21**, 1087 (1953).

50. M. Allen and D. Tildesley, *Computer simulation of liquids* (Oxford: Clarendon Press, New York, 1989).

51. H. C. Andersen, The Journal of Chemical Physics **72**, 2384 (1980).

52. S. Nosé, Molecular Physics **52**, 255 (1984).

53. W. Hoover, Physical Review A **31**, 1695 (1985).

54. Y. Mori and Y. Okamoto, Journal of the Physical Society of Japan **79**, 074001 (2010).

55. M. R. Shirts and J. D. Chodera, The Journal of Chemical Physics **129**, 124105 (2008).

56. A. Ferrenberg and R. Swendsen, Physical Review Letters **63**, 1195 (1989).

57. S. Kumar, J. M. Rosenberg, D. Bouzida, R. H. Swendsen, and P. A. Kollman, Journal of Computational Chemistry **13**, 1011 (1992).

58. Z. Tan, E. Gallicchio, M. Lapelosa, and R. M. Levy, The Journal of Chemical Physics **136**, 144102 (2012).

59. G. J. Martyna, M. E. Tuckerman, D. J. Tobias, and M. L. Klein, Molecular Physics **87**, 1117 (1996).

60. S. G. Itoh, T. Morishita, and H. Okumura, The Journal of Chemical Physics **139**, 064103 (2013).

61. M. Matsumoto and T. Nishimura, ACM Transactions on Modeling and Computer Simulation (TOMACS) **8**, 3 (1998).

62. H. Okumura and F. Yonezawa, The Journal of Chemical Physics **113**, 9162 (2000).

63. H. Kokubo and Y. Okamoto, Biophysical Journal **96**, 765 (2008).

64. I. R. Cooke, K. Kremer, and M. Deserno, Physical Review E **72**, 011506 (2005).

65. M. Deserno, Macromolecular Rapid Communications **30**, 752 (2009).

66. B. J. Reynwar and M. Deserno, Biointerphases **3**, FA117 (2008).

67. M. S. Friedrichs, P. Eastman, V. Vaidyanathan, M. Houston, S. Legrand, A. L. Beberg, D. L. Ensign, C. M. Bruns, and V. S. Pande, Journal of Computational Chemistry **30**, 864 (2009).

68. P. Eastman and V. S. Pande, Journal of Computational Chemistry **31**, 1268 (2010).

69. P. Eastman, M. S. Friedrichs, J. D. Chodera, R. J. Radmer, C. M. Bruns, J. P. Ku, K. A. Beauchamp, T. J. Lane, L.-P. Wang, D. Shukla, et al., Journal of Chemical Theory and Computation **9**, 461 (2013).

70. G. Bussi and M. Parrinello, Physical Review E **75**, 056707 (2007).

71. C. Simmerling, B. Strockbine, and A. E. Roitberg, Journal of the American Chemical Society **124**, 11258 (2002).

72. A. Onufriev, D. Bashford, and D. A. Case, Proteins: Structure, Function, and Bioinformatics **55**, 383 (2004).

73. J. D. Chodera and M. R. Shirts, The Journal of Chemical Physics **135**, 194110 (2011).

74. The temperature-updating acceptance rates ranged from 29% to 35% in all simulations. These values are larger than the corresponding values in the REMD and MSREMD simulations obtained in the previous study (14% to 19%)[35]. This increase in the acceptance ratios is consistent with Refs. 79–82 and often credited as the reason for the superior sampling efficiency of ST over REMD methods. Nevertheless, this may not necessarily be the case, especially in the vicinity of a phase transition[83]. A more in-depth discussion of this point would be interesting, but is beyond the scope of this work.

75. H. Flyvbjerg and H. G. Petersen, The Journal of Chemical Physics **91**, 461 (1989).

76. B. A. Berg, *Markov Chain Monte Carlo Simulations and Their Statistical Analysis* (World Scientific, Singapore, 2004), ; http://www.worldscibooks.com/physics/5602.html.

77. D. Frenkel and B. Smit, *Understanding molecular simulation: from algorithms to applications* (Elsevier, 2001).

78. R. D. Mountain and D. Thirumalai, Physica A: Statistical Mechanics and its Applications **210**, 453 (1994).

79. A. Mitsutake and Y. Okamoto, Chemical Physics Letters **332**, 131 (2000).

80. X. Huang, G. R. Bowman, and V. S. Pande, The Journal of Chemical Physics **128**, 205106 (2008).

81. C. Zhang and J. Ma, The Journal of Chemical Physics **129**, 134112 (2008).

82. S. Park, Physical Review E **77**, 016709 (2008).

83. C. E. Fiore and M. G. E. da Luz, Physical Review E **82**, 031104 (2010).

84. W. Kabsch and C. Sander, Biopolymers **22**, 2577 (1983).

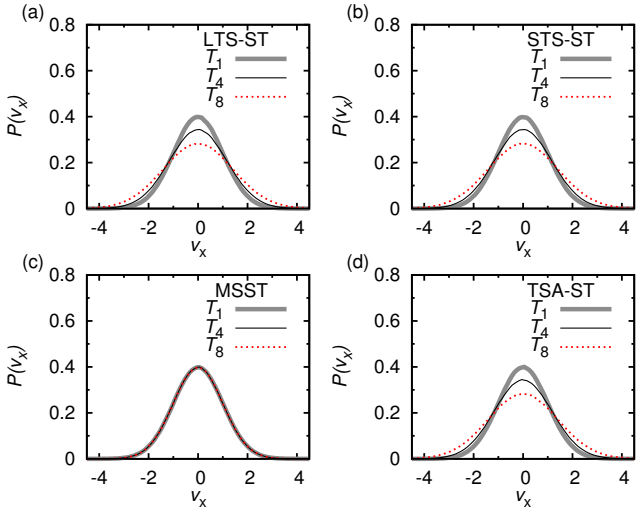


Figure 1: Probability density functions of v_x, $p(v_x)$, obtained in (a) LTS-ST, (b) STS-ST (c) MSST, and (d) TSA-ST simulations. Gray thick, black thin, and red dashed lines represent the functions for T_1, T_4 and T_8, respectively. Those at other temperatures were suppressed for the sake of clarity, as were error bars. The lines completely overlap in the MSST simulation [see (c)].

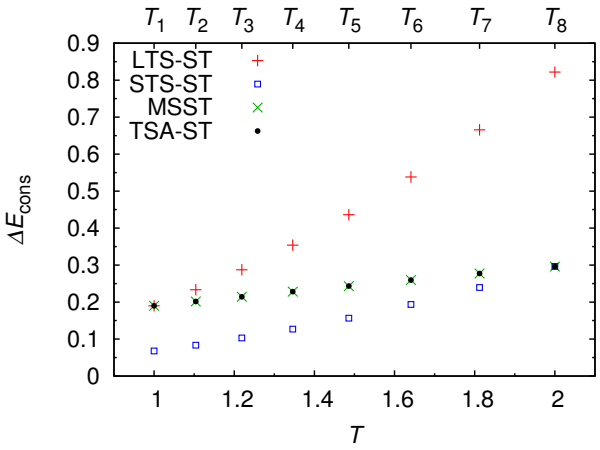


Figure 2: Numerical instability, as measured by ΔE_{cons}, plotted against temperature. As the error bars are small, they were suppressed for the sake of clarity. Red plus signs, blue squares, green crosses, and black point marks represent the LTS-ST, STS-ST, MSST, and TSA-ST simulations, respectively.

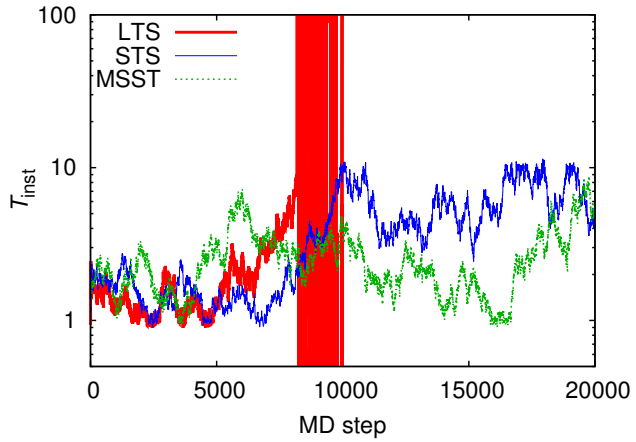


Figure 3: Instantaneous temperatures T_{inst} are plotted against MD step. Red thick, blue thin, and green dot lines marks LTS-ST, STS-ST and MSST simulations, respectively. Ordinate is shown in the logarithmic scale. The LTS-ST simulation begins to show unreasonable behavior from the 8200th step, and is only plotted to 10000th step.

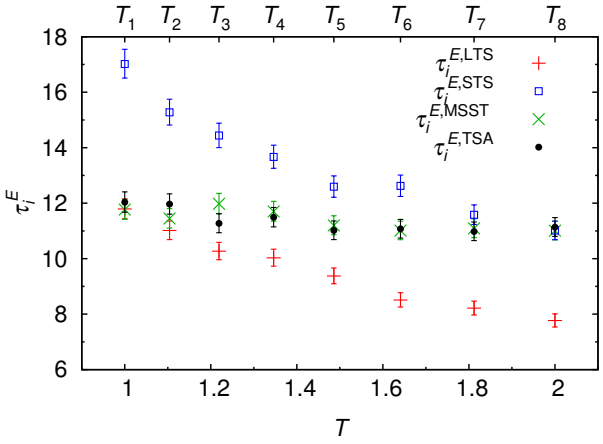


Figure 4: Autocorrelation time for potential energy plotted against temperature. Red plus signs, blue boxes, green crosses, and black point marks represent the values for the LTS-ST ($\tau_i^{E,\mathrm{LTS}}$), STS-ST ($\tau_i^{E,\mathrm{STS}}$), MSST ($\tau_i^{E,\mathrm{MS}}$), and TSA-ST ($\tau_i^{E,\mathrm{TSA}}$) simulations, respectively.

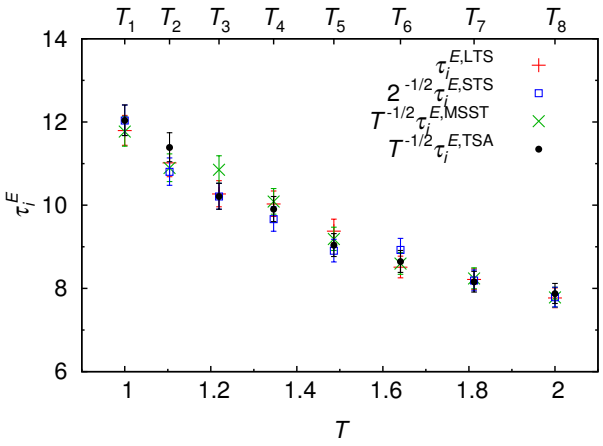


Figure 5: Scaled autocorrelation time for potential energy plotted against temperature. Red plus signs, blue boxes, green crosses, and black point marks represent $\tau_i^{E,\mathrm{LTS}}$, $(1/\sqrt{2})\tau_i^{E,\mathrm{STS}}$, $(1/\sqrt{T_i})\tau_i^{E,\mathrm{MSST}}$, and $(1/\sqrt{T_i})\tau_i^{E,\mathrm{TSA}}$, respectively. The four values at each temperature are in good agreement.

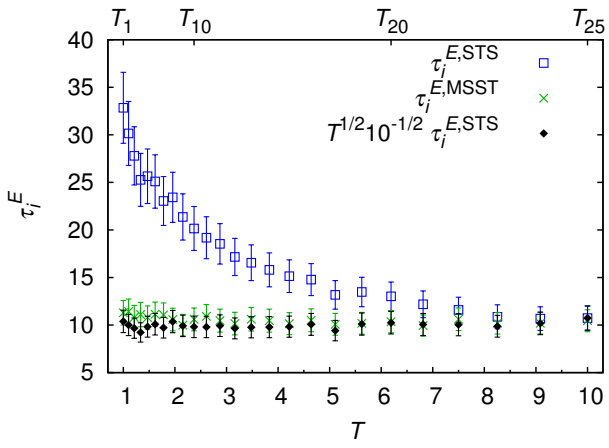


Figure 6: Autocorrelation time for potential energy plotted against temperature for the simulations with 25 temperatures from 1 to 10. Blue boxes and green crosses represent the STS-ST ($\tau_i^{E,\mathrm{STS}}$) and MSST ($\tau_i^{E,\mathrm{MS}}$) simulations, respectively. Black diamonds represent $\sqrt{T_i}/\sqrt{10}\,\tau_i^{E,\mathrm{STS}}$, scaled in accordance to the effective and real time steps. LTS-ST simulations (not shown) were unstable using the too large time step.

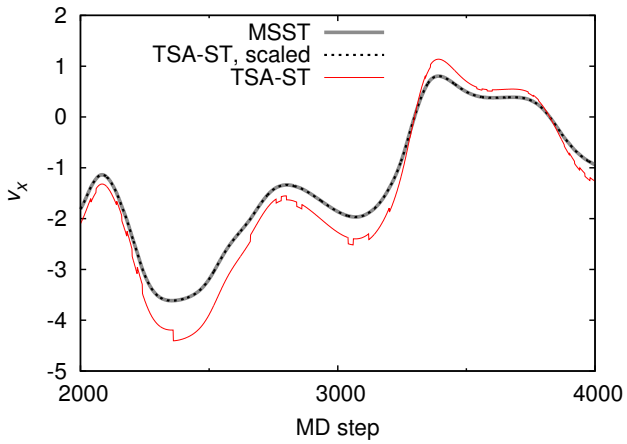


Figure 7: x-component velocities for the first particle plotted as a function of the MD step for the TSA and MSST simulations. Thick gray and dotted black lines represent the velocity of the MSST and scaled velocity of the TSA-ST ($\frac{1}{\sqrt{T_i/T_1}}v_x^{\mathrm{TSA}}$) simulations, respectively. Red line represents the unscaled velocity for the TSA-ST simulation (v_x^{TSA}). The abrupt changes in $v_x^{\text{TSA-ST}}$ reflect scaling at accepted temperature updates.

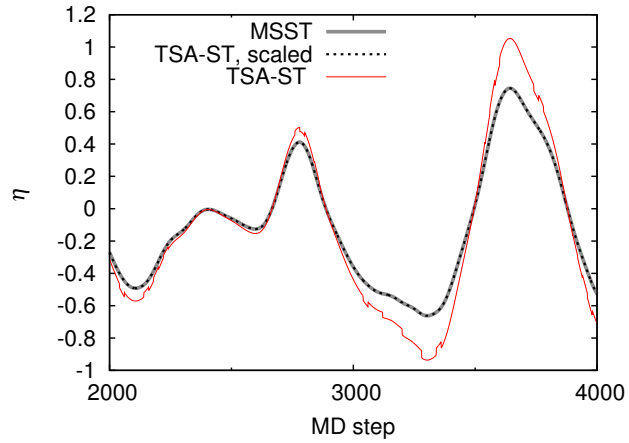


Figure 8: Rates of Nosé-Hoover thermostats plotted as a function of the MD step for the MSST and TSA-ST simulations. Thick gray line represents the rate of the Nosé-Hoover thermostat for the MSST, and dotted black line the scaled rate for the TSA-ST simulation ($\frac{1}{\sqrt{T_i/T_1}}\eta^{\mathrm{TSA}}$). Red line represents the unscaled velocity for the TSA-ST simulation (η^{TSA}). The abrupt changes in η^{TSA} are a consequence of scaling at accepted temperature updates.

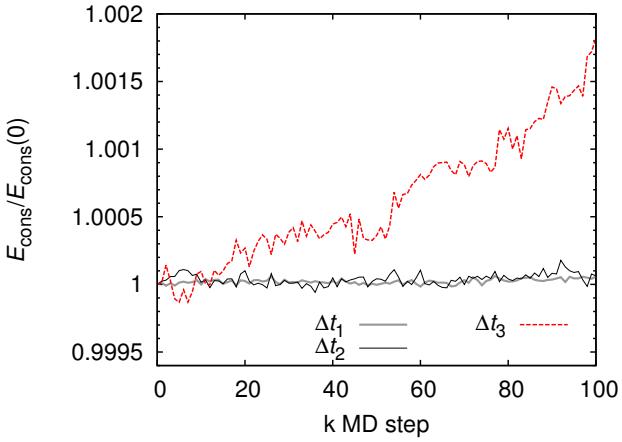


Figure 9: Normalized conserved quantity $E_{\mathrm{cons}}/E_{\mathrm{cons}}(0)$ is plotted against MD step for the normal simulations at $T = 2.0$. The unit of abscissa is kilo MD step. Gray thick, black thin, and red dash lines represent the results for $\Delta t_1 = 0.005$, $\Delta t_2 = 0.0075$, and $\Delta t_3 = 0.01$, respectively. Simulation with $\Delta t = 0.015$ (not shown) was unstable. At this temperature, a drift is observed even with the time step of $\Delta t = 0.01$.

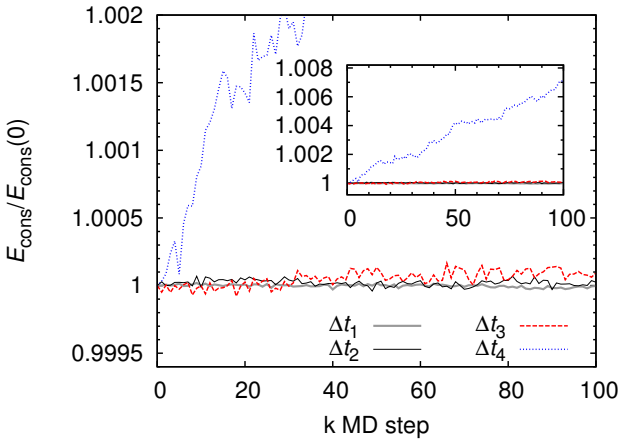


Figure 10: Normalized conserved quantity $E_{\rm cons}/E_{\rm cons}(0)$ is plotted against MD step for the mass scaling simulations at $T = 2.0$. The unit of abscissa is kilo MD step. Gray thick, black thin, red dash, and blue dot lines represent the results for $\Delta t_1 = 0.005$, $\Delta t_2 = 0.0075$, $\Delta t_3 = 0.01$, and $\Delta t_4 = 0.015$, respectively. The inset is the same plot with a different range. A small drift was observed for the ambitious time step of $\Delta t = 0.015$.

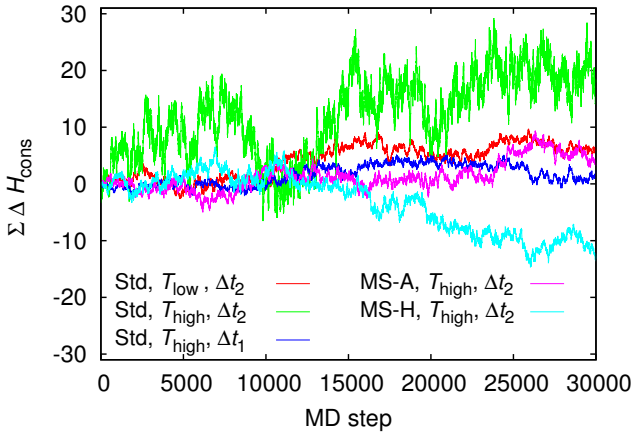


Figure 11: Conserved quantity as a function of MD step. Red and green lines mark the standard canonical simulation with $\Delta t_2 = 1.0$ fs at $T_{\rm low} = 273$ K and $T_{\rm high} = 819$ K, respectively. Blue line marks standard simulation with $\Delta t_1 = 0.5$ fs at 819 K. Magenta line represents the simulation with $\Delta t_2 = 1.0$ fs with the mass scaling applied to all atoms, while the cyan line represents the simulation with $\Delta t_2 = 1.0$ fs with the mass scaling applied to hydrogen atoms only.

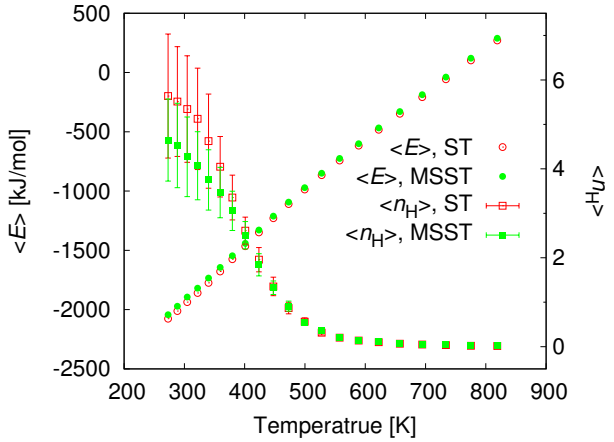


Figure 12: Average potential energy $\langle E \rangle$ and average helix content $\langle n_{\mathrm{H}} \rangle$ as a function of temperature. Open and closed squares represent the average number of residues judged as helix by DSSP[84]. The bars show the standard deviation of the replicates of simulations. Open and closed circles mark the average potential energy obtained with ST and MSST simulations, respectively. The size of error was comparative to the size of mark.

Table 1: Empirical probability $P(T_i)$ of visiting temperature T_i in the four LJ fluid simulations. Considering the autocorrelation time for the temperature index, the typical errors exceed the approximate value of 0.001.

	LTS-ST	STS-ST	MSST	TSA-ST
T_1	0.140	0.135	0.139	0.138
T_2	0.135	0.130	0.132	0.132
T_3	0.128	0.125	0.126	0.126
T_4	0.120	0.120	0.120	0.120
T_5	0.116	0.117	0.116	0.117
T_6	0.119	0.121	0.120	0.120
T_7	0.122	0.126	0.124	0.124
T_8	0.121	0.125	0.124	0.123

Table 2: Average kinetic energy $\langle K \rangle$ obtained with the four simulations at all reference temperatures. The kinetic energy was calculated with the velocities at half time steps[60]. Errors were evaluated using the jackknife method with 1000 bins. Exact average kinetic energy $\frac{3}{2}NT$ is also given.

T_l	$\frac{3}{2}NT_l$	LTS	STS	MSST	TSA
1.000	750.000	749.979 ± 0.009	749.997 ± 0.012	749.997 ± 0.009	750.016 ± 0.009
1.104	828.000	828.027 ± 0.014	828.029 ± 0.018	828.015 ± 0.014	828.013 ± 0.015
1.219	914.250	914.255 ± 0.016	914.216 ± 0.021	914.250 ± 0.017	914.241 ± 0.017
1.346	1009.500	1009.495 ± 0.017	1009.497 ± 0.023	1009.479 ± 0.029	1009.476 ± 0.019
1.486	1114.500	1114.502 ± 0.018	1114.514 ± 0.025	1114.525 ± 0.022	1114.509 ± 0.022
1.641	1230.750	1230.750 ± 0.019	1230.749 ± 0.026	1230.757 ± 0.024	1230.729 ± 0.024
1.812	1359.000	1358.989 ± 0.021	1359.004 ± 0.026	1358.984 ± 0.027	1359.033 ± 0.026
2.000	1500.000	1500.002 ± 0.015	1499.993 ± 0.019	1499.986 ± 0.021	1499.965 ± 0.020

Table 3: Average potential energy $\langle E \rangle$ obtained with the four methods at all reference temperatures. Errors were evaluated using the jackknife method with 1000 bins.

T_l	LTS	STS	MSST	TSA
1.000	-2519.604 ± 0.022	-2519.500 ± 0.026	-2519.630 ± 0.022	-2519.567 ± 0.022
1.104	-2474.296 ± 0.024	-2474.226 ± 0.025	-2474.305 ± 0.025	-2474.319 ± 0.025
1.219	-2425.878 ± 0.026	-2425.790 ± 0.027	-2425.830 ± 0.027	-2425.858 ± 0.026
1.346	-2374.052 ± 0.027	-2373.959 ± 0.031	-2374.000 ± 0.031	-2374.011 ± 0.029
1.486	-2318.681 ± 0.029	-2318.611 ± 0.032	-2318.651 ± 0.032	-2318.634 ± 0.033
1.641	-2259.383 ± 0.029	-2259.212 ± 0.035	-2259.245 ± 0.035	-2259.294 ± 0.036
1.812	-2195.888 ± 0.032	-2195.771 ± 0.036	-2195.787 ± 0.036	-2195.785 ± 0.037
2.000	-2128.297 ± 0.035	-2128.027 ± 0.040	-2128.104 ± 0.040	-2128.154 ± 0.040

Table 4: Summary of simulations of model lipid bilayer noting if simulation was stable ($\circ$) or unstable ($\times$) after 10^5 time steps.

	$\Delta t = 0.005$	$\Delta t = 0.0075$	$\Delta t = 0.01$	$\Delta t = 0.015$
$T = 1.1$	$\circ$	$\circ$	$\circ$	$\circ$
$T = 1.4$	$\circ$	$\circ$	$\circ$	$\times$
$T = 2.0$	$\circ$	$\circ$	$\circ$	$\times$
$T = 4.0$	$\circ$	$\circ$	$\times$	$\times$

Appendix: Equivalence of temperature updating in the MSST and TSA-ST methods

Assume that at time $t = 0$, without loss of generality, the MSST and corresponding TSA-ST simulations are at T_i and at the states that meet the following conditions:

$$q^{\mathrm{TSA}}(0) = q^{\mathrm{MSST}}(0), \tag{37}$$

$$\dot{q}^{\mathrm{TSA}}(0) = \sqrt{T_i/T_1}\dot{q}^{\mathrm{MSST}}(0), \tag{38}$$

$$\eta^{\mathrm{TSA}}(0) = \sqrt{T_i/T_1}\eta^{\mathrm{MSST}}(0). \tag{39}$$

As these conditions satisfy Eqs. 31–33 and the same time evolution is expected based on the Nosé-Hoover equations of motion. In this Appendix, we demonstrate that the equivalence in the time evolution of the coordinate between the two methods is preserved even after temperature updates.

Let us consider temperature updates from T_i to T_j. First, we note that both simulations have the same potential energy values, such that the Metropolis criteria must produce the same results, if the seed of the random number generator is the same. Given that the temperature-updating attempts are accepted, the system is coupled to T_j. In TSA-ST simulations $\dot{q}$ and η are scaled by $\sqrt{T_j/T_i}$, which yields

$$q^{\mathrm{TSA}}(0) = q^{\mathrm{MSST}}(0), \tag{40}$$

$$\dot{q}^{\mathrm{TSA}}(0)' = \sqrt{T_j/T_i}\dot{q}^{\mathrm{TSA}}(0) = \sqrt{T_j/T_1}\dot{q}^{\mathrm{MSST}}(0), \tag{41}$$

$$\eta^{\mathrm{TSA}}(0)' = \sqrt{T_j/T_i}\eta^{\mathrm{TSA}}(0) = \sqrt{T_j/T_1}\eta^{\mathrm{MSST}}(0). \tag{42}$$

These equations suggest that Eqs. 31–33 are met even after the accepted temperature update to the new temperature T_j. Therefore, the two simulations should have the same time evolution even with temperature updates.