

UvA-DARE (Digital Academic Repository)

A one-way shooting algorithm for transition path sampling of asymmetric barriers

Brotzakis, Z.F.; Bolhuis, P.G.

DOI

[10.1063/1.4965882](https://doi.org/10.1063/1.4965882)

Publication date

2016

Document Version

Final published version

Published in

Journal of Chemical Physics

License

Article 25fa Dutch Copyright Act (<https://www.openaccess.nl/en/policies/open-access-in-dutch-copyright-law-taverne-amendment>)

[Link to publication](#)

Citation for published version (APA):

Brotzakis, Z. F., & Bolhuis, P. G. (2016). A one-way shooting algorithm for transition path sampling of asymmetric barriers. *Journal of Chemical Physics*, 145(16), Article 164112. <https://doi.org/10.1063/1.4965882>

General rights

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <https://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, P.O. Box 19185, 1000 GD Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (<https://dare.uva.nl>)

A one-way shooting algorithm for transition path sampling of asymmetric barriers

Z. Faidon Brotzakis and Peter G. Bolhuis^{a)}

Van 't Hoff Institute for Molecular Sciences, University of Amsterdam, P.O. Box 94157, 1090 GD Amsterdam, The Netherlands

(Received 14 July 2016; accepted 7 October 2016; published online 31 October 2016)

We present a novel transition path sampling shooting algorithm for the efficient sampling of complex (biomolecular) activated processes with asymmetric free energy barriers. The method employs a fictitious potential that biases the shooting point toward the transition state. The method is similar in spirit to the aimless shooting technique by Peters and Trout [J. Chem. Phys. **125**, 054108 (2006)], but is targeted for use with the one-way shooting approach, which has been shown to be more effective than two-way shooting algorithms in systems dominated by diffusive dynamics. We illustrate the method on a 2D Langevin toy model, the association of two peptides and the initial step in dissociation of a β -lactoglobulin dimer. In all cases we show a significant increase in efficiency. *Published by AIP Publishing.* [<http://dx.doi.org/10.1063/1.4965882>]

I. INTRODUCTION

Straightforward molecular dynamics of activated molecular processes, such as chemical reactions, biomolecular isomerizations, association processes, or phase transitions, is inefficient, as much time is wasted on sampling the stable states, in comparison to the transition itself. While many biasing methods exist that address the exponential time scale problem occurring in such activated processes, they typically rely on good reaction coordinates (e.g., umbrella sampling,¹ flooding,² local elevation,³ adaptive bias force,⁴ and metadynamics⁵) or give up access to kinetic and mechanistic details (such as replica exchange molecular dynamics⁶). Transition path sampling (TPS) was developed to bypass these problems by focusing on the transition paths between stable states.^{7–9} The TPS algorithm employs importance sampling to generate an ensemble of unbiased dynamical trajectories that connect an initial with a final stable state, and is therefore independent of an *a priori* choice of order parameters or reaction coordinate that describes the transition. Rather, the reaction coordinate can be extracted from the sampled path ensemble. The method has been successfully applied to various activated processes (see Refs. 10–12 for a review).

The standard TPS algorithm⁸ takes an existing path, and creates a new trial path from it. The resulting trial path is accepted using a Metropolis-Hastings criterion based on detailed balance, designed to maintain the equilibrium path ensemble distribution. This generic TPS setup allows for much flexibility, as the way how to generate a new trial path from an existing path is entirely up to the user. Even in the seminal publications on TPS,⁷ different approaches were considered, such as frame by frame moves, or dynamical schemes evolving the entire pathway simultaneously. The vanilla type, standard TPS algorithm relies on the so-called shooting move,⁹ which selects a random frame from an existing initial trajectory of

fixed duration, alters the momenta slightly, and integrates the equation of motion forward as well as backward in time, until the total duration of the trajectory has been reached. For the shooting algorithm the acceptance criterion is simply determined by whether it connects the two stable states. (The original approach also included the shifting move. This move shifts the origin of the paths, but does not create new decorrelated pathways, at least for deterministic MD). The shooting moves are particularly effective since they allow for a quick decorrelation of the subsequent pathways, while still giving reasonable good acceptance probability as long as the trial shooting points are not too far from the transition states. While the shooting move has been employed for the majority of cases,^{10–12} other TPS approaches can be very useful as well, such as the noise history approach.¹³

Following the introduction of the original shooting algorithm, many improvements were made. The transition interface sampling (TIS)¹⁴ based on flexible shooting moves allowed restricting the path length to the necessary minimum, thus minimizing the computational effort. Biasing the shooting point toward the barrier region also is very useful, as it prevents ill-fated shots from near the stable states and improves the acceptance ratio.^{12,15} Another important development was the aimless shooting approach by Peters and Trout,¹⁶ which was originally developed to be applied in conjunction with a maximum likelihood based reaction coordinate analysis advocated by the same authors, but can be used independently as a valid shooting algorithm. The idea of the aimless shooting method is to keep the trial shooting point distribution close to the transition barrier. The reason for this goal is twofold. First, the acceptance ratio for the shooting is higher when the shooting point is close to the transition state. Second, the likelihood maximization relies on information close to the dividing surface. The aimless shooting method achieves these goals by basing the trial shooting point on the previous shooting point, shifting it by a fixed number of frames in the forward or backward direction. When this shift happens

^{a)}p.g.bolhuis@uva.nl.

to bring the shooting point closer to the transition state, the acceptance becomes more likely, thus providing a natural tendency to stay close to the barrier, thus circumventing the need for predefined order parameters to bias in.

At the same time, it became clear that the standard two-way shooting move was less effective for complex systems, in particular, biomolecular systems, where the transition paths could become exceedingly long, much longer than the Lyapunov time scale. This means that (deterministic) dynamical pathways effectively become diffusive. In the original TPS publications it was already clear that for diffusive dynamics one had to employ the so-called one-way or stochastic shooting method.¹⁷ This method attempts to create only a forward or a backward *partial* path from a uniformly randomly chosen trial shooting point rather than shooting in both directions. When the forward(backward) path is acceptable, that is, when it reaches the final (initial) state, the trial path replaces only part of the existing path, while the complementary part of the previous path is retained. While clearly several shots are needed to create an entirely new path, the one-way shooting algorithm is more efficient for diffusive or stochastic dynamics. Moreover, even in case of deterministic molecular dynamics (coupled to a stochastic thermostat) in which the molecular chaos sets in long before the average length of the path is reached, one-way shooting is preferred.¹⁸ In fact, for biomolecular systems it is currently the standard approach. The effectiveness of the path sampling can be tracked using a so-called path tree.¹⁹

The one-way shooting method has also several drawbacks. First, it requires more successful shots to decorrelate paths (although not more computer time). More seriously, it suffers in efficiency for asymmetric barriers. Asymmetric barriers occur, for instance, when the system on one side of the main barrier is trapped in an intermediate state, while it can easily reach the stable state on the other side. This means that the paths on the trapped side become much longer (note that this is, of course, also a problem for the two-way shooting algorithm). When uniform one-way shooting is used, this asymmetry leads to many more shooting attempts on one side of the barrier with respect to the other. Due to the asymmetry, most shooting points are chosen on the trapped side. A biasing function that selects shooting points around the transition state using an order parameter can ameliorate this situation. However, then one needs to know *a priori* what order parameter to use to parameterize the transition state, and TPS was developed just to avoid that. Clearly one would prefer to use an algorithm such as aimless shooting, which automatically finds the transition state region. The current work provides such an algorithm, which we denote “spring shooting.” Inspired by aimless shooting, the method chooses a new shooting point by shifting the old shooting point by a number of frames, and accepts or rejects this shift while applying a bias, the spring. This bias is different for a forward shot with respect to a backward shot. For a forward shot the shooting point is pulled to lower frame number by the “spring,” for a backward shot the situation is the other way around. The opposing forces balance when the shooting point oscillates around the transition state.

The paper is organized as follows: we start with introduction and derivation of the algorithm. We illustrate method using a 2D Langevin model, and show that the efficiency improves with respect to the uniform shooting. We then further illustrate the new algorithms on the association reaction of two diphenylalanine (FF) solvated dipeptides. We end with an application of a large system, the associating/dissociation reaction of the β -lactoglobulin dimer.

II. METHODS

A. Derivation of the spring shooting algorithm

A TPS simulation performs a Monte Carlo random walk in trajectory space. A dynamical trajectory $\mathbf{x}(L) = \{x_0, x_1, \dots, x_L\}$ is discretized in time by a time step Δt into $L + 1$ slices (or frames) x_τ . Each time frame $x_\tau = \{\mathbf{r}_\tau, \mathbf{p}_\tau\}$ contains all positions r and momenta p of all N particles in the system at time $t = \tau\Delta t$. The probability $\mathcal{P}_{\mathcal{AB}}[\mathbf{x}(L)]$ for a path of a fixed length L connecting the initial state A and the final state B is given by

$$\mathcal{P}_{\mathcal{AB}}[\mathbf{x}(L)] = Z_{AB}^{-1} h_A(x_0) \mathcal{P}[\mathbf{x}(L)] h_B(x_L). \quad (1)$$

$\mathcal{P}[\mathbf{x}(L)]$ is the dynamical probability of the discretized path $\mathbf{x}(L)$, while $h_\Omega(x)$ denotes an indicator function that is unity when $x \in \Omega$, i.e., x is inside a stable state Ω , and zero otherwise. The product $h_A(x_0)h_B(x_L)$ thus guarantees that the probability is non-zero only when a path starts from A and ends in B . The normalization factor Z_{AB} is akin to a partition function. The dynamical path probability $\mathcal{P}[\mathbf{x}(L)]$ is given by

$$\mathcal{P}[\mathbf{x}(L)] = \rho(x_0) \prod_{\tau=0}^{L-1} p(x_\tau \rightarrow x_{(\tau+1)}), \quad (2)$$

where $p(x_\tau \rightarrow x_{(\tau+1)})$ represents the Markovian probability for transitions from a phase point at time frame τ to one at time frame $\tau + 1$ (e.g., a delta function for molecular dynamics, or a Gaussian for Langevin dynamics⁷), and $\rho(x_0)$ denotes the stationary distribution of x_0 . All trajectories that connect the two defined stable states form the *path ensemble*, which plays a key role in TPS as the representation of the rare event process under study. A definition of reaction coordinate is not necessary in TPS, but the reactant A and product B regions should be defined properly by order parameters. These order parameters should be chosen such that the two states can not only be distinguished from each other but also that each state lies inside the basin of attraction of that stable state.¹²

The shooting algorithm⁷ can efficiently sample the path ensemble by selecting a random slice, changing the momenta slightly, and shooting off a new direction forward and backward in time. Each path that still connects A with B can be accepted in the Monte Carlo procedure. The general acceptance rule for the shooting move is^{8,12}

$$P_{acc}[\mathbf{x}^{(o)} \rightarrow \mathbf{x}^{(n)}] = h_A(x_0^{(n)}) h_B(x_L^{(n)}) \min \left[1, \frac{\mathcal{P}[\mathbf{x}^{(n)}(L^{(n)})] P_{gen}[\mathbf{x}^{(n)} \rightarrow \mathbf{x}^{(o)}]}{\mathcal{P}[\mathbf{x}^{(o)}(L^{(o)})] P_{gen}[\mathbf{x}^{(o)} \rightarrow \mathbf{x}^{(n)}]} \right], \quad (3)$$

where the min function returns the lower of its arguments and $P_{gen}[\mathbf{x}^{(n)} \rightarrow \mathbf{x}^{(o)}]$ denotes the generating probability for creating a new trajectory from an old one, making use of microscopic reversibility. One can show that for the shooting move this acceptance rule reduces to

$$P_{acc}[\mathbf{x}^{(o)} \rightarrow \mathbf{x}^{(n)}] = h_A(x_0^{(n)})h_B(x_L^{(n)}) \min \left[1, \frac{p_{sel}(\tau', \mathbf{x}^{(n)})}{p_{sel}(\tau, \mathbf{x}^{(o)})} \right], \quad (4)$$

where $p_{sel}(\tau, \mathbf{x})$ denotes the probability to select the shooting point. In the simplest case, this selection probability is uniform, and is just determined by the length $p_{sel}(\tau, \mathbf{x}) = 1/L$. Note that when the generation of the trajectories is halted upon reaching a stable state, as is done in flexible shooting, the length of the paths varies and the acceptance criterion becomes dependent on the ratio $L^{(o)}/L^{(n)}$.^{12,14}

However we are free to choose a different selection probability, for instance,

$$p_{sel}(\tau, \mathbf{x}) = \frac{f(\lambda(x_\tau))}{\sum_{i=0}^L f(\lambda(x_i))} \quad (5)$$

where the function $f(\lambda(x_i))$ is based on the order parameter $\lambda(x)$ which in turn depends solely on the coordinates of the time frame.

To sample from this distribution one can generate the shooting points by uniform selection, do the shooting move, and then accept with the criterion Eq. (4), or, alternatively, generate the shooting point directly from this distribution and perform the shooting, and accept with $P_{acc}^{shot}[\mathbf{x}^{(o)} \rightarrow \mathbf{x}^{(n)}] = h_A(x_0^{(n)})h_B(x_L^{(n)})$. To generate the shooting point from the distribution Eq. (5) one can use e.g., rejection (von Neumann) sampling, or other approaches.^{20,21}

The shooting move is thus separated into two stages. The first is the generation of the shooting point, by a Metropolis Monte Carlo scheme, in which undesired shooting points can be quickly discarded, with a acceptance ratio $P_{acc}^{sp}[\tau \rightarrow \tau'] = \min [1, p_{sel}(\tau', \mathbf{x}^{(n)})/p_{sel}(\tau, \mathbf{x}^{(o)})]$. The second stage is the creation and acceptance/rejection of the trial path. The total move acceptance ratio can be seen as product of the two stages,

$$\begin{aligned} P_{acc}[\tau \rightarrow \tau'; \mathbf{x}^{(o)} \rightarrow \mathbf{x}^{(n)}] \\ &= P_{acc}^{sp}[\tau \rightarrow \tau'] P_{acc}^{shot}[\mathbf{x}^{(o)} \rightarrow \mathbf{x}^{(n)}] \\ &= h_A(x_0^{(n)})h_B(x_L^{(n)}) \min \left[1, \frac{p_{sel}(\tau', \mathbf{x}^{(n)})}{p_{sel}(\tau, \mathbf{x}^{(o)})} \right], \quad (6) \end{aligned}$$

which is indeed the same as Eq. (4). In previous work such an approach has been used to great effect.¹⁵

Eq. (6) holds also for the one-way shooting algorithm, in which one randomly chooses to generate a forward or a backward partial path starting from the shooting point. The complementary partial part is retained from the old trajectory. Note that one-way shooting requires some stochasticity in the dynamics, so that the shooting point itself does not have to be altered. Indeed, altering the shooting point like in standard two-way TPS would lead to unrealistic dynamics along the entire path.

The spring shooting algorithm is especially developed for use with the one-way algorithm. Similar to the aimless shooting algorithms, the spring shooting selects the shooting

point not based on coordinates but only based on the time frame index,

$$p_{sel}(\tau, \mathbf{x}) = p_{sel}(\tau). \quad (7)$$

The essence of the spring shooting algorithm is to treat the forward and backward shooting moves as independent moves. Similar to aimless shooting, the spring shooting starts by shifting the shooting point with respect to the successful shooting point on the previous path by a certain number of frames. Whereas in aimless shooting this shift is symmetric, in the spring method it is selected with an exponential function,

$$p_{sel}(\tau) = c \exp(sk\tau), \quad (8)$$

where τ is the (absolute) frame index, k a force constant determining the magnitude of the bias, and $s \in \{-1, 1\}$ is determined by the direction of shooting, $s = -1$ for forward shooting, and $s = 1$ for backward shooting, and c is the normalization. Since we now allow all times (in principle) this normalization constant is given by $c = 1/\sum_{-b}^b \exp(sk\tau)$, where the boundary b of the allowed time frame window be made arbitrarily large. While this normalization constant vanishes for $b \rightarrow \infty$, it cancels out in the acceptance criterion itself,²²

$$P_{acc}^{sp}[\tau \rightarrow \tau'] = \min \left[1, \frac{\exp(sk\tau')}{\exp(sk\tau)} \right] = \min[1, e^{sk\Delta\tau}], \quad (9)$$

where $\Delta\tau = \tau' - \tau$ is the number of shifted frames from the previous shooting point. As a large $\Delta\tau$ either yields an exponentially small acceptance ratio or is likely to produce a failed shot, we limit the choice of $\Delta\tau$ between the interval $[-\Delta\tau_{max}, \Delta\tau_{max}]$, analogous to the maximum allowed displacement in a regular MC translational move. When the trial shooting point falls outside the current path the acceptance probability becomes zero, and the move is rejected.

As the spring shooting algorithm treats the forward and backward shots separately, the complete algorithm is as follows:

1. Select with equal probability a forward or a backward move. Set $s = -1$ in case of forward move, $s = 1$ in case of backward.
2. Select a uniform shift $\Delta\tau$ in shooting point from the interval $[-\Delta\tau_{max}, \Delta\tau_{max}]$. Add this shift to the previous shooting point frame index $\tau' = \tau + \Delta\tau$. Reject the entire move if the index is outside of the current path.
3. Accept the trial shooting point τ' according to Eq. (9), otherwise reject the entire move.
4. Create partial path (forward or backward) employing molecular dynamics with a stochastic thermostat. Halt the path when a stable state is reached. Reject the move when the total trial path length exceeds a maximum length L_{max} . Glue the path to the complementary existing path, and accept according to the standard criterion $P_{acc}[\mathbf{x}^{(o)} \rightarrow \mathbf{x}^{(n)}] = h_A(x_0^{(n)})h_B(x_L^{(n)})$.

The advantage of this approach is that unfavorable shooting points are discarded without extra cost. The exponential shooting selection function can be viewed as a kind of external field pushing (or pulling) the shooting point forward (or backward). The shooting points for forward moves are preferably taken from frames before the previous shooting

point, and the backward movers preferably after the previous shooting point. In this way pathways are decorrelated as much as possible, without wasting time creating partial paths that do not contribute to the decorrelation. Note that the acceptance criterion for the spring move is very simple, and in contrast to the standard one-way flexible path length algorithm, does not depend on the instantaneous (fluctuating) path length. Note also that the algorithm rejects trial paths which become longer than L_{max} , which is set to prevent memory or storage problems, or as an indication that the path generation went awry, e.g., got stuck in a long-lived intermediate state.

The algorithm requires an initial shooting point location. While in principle it does not matter where this point is chosen, in practice it is more efficient to start close to the barrier. When the initial trajectory comes from gluing two trajectories together (e.g., obtained in a committor analysis), the initial shooting point naturally should be chosen close to the configuration from where these trajectories were initiated. In other cases it might be wise to establish a proxy for the barrier e.g., the maximum energy along the initial path. Even visual inspection might help to find a suitable shooting point location. Nevertheless, we stress that even when starting with an entirely wrong initial shooting point, the algorithm ensures that the shooting point diffuses to the top of the barrier eventually.

The algorithm has two crucial parameters: the maximum frame shift $\Delta\tau_{max}$ and the spring constant k . In the examples we investigate how the performance of the algorithm depends on the choice of these two parameters.

B. Simulation details

1. 2D Langevin model

To test the performance and validity of the spring shooting algorithm we employ the method first on a toy model consisting of a particle evolving according to Langevin dynamics in a simple 2D asymmetric potential. We choose the potential to be

$$V(x, y) = -6e^{-(4+x)^2-y^2} - 12e^{-3(9+x)^2-y^2} + 0.0177778(0.0625x^4 + y^4). \quad (10)$$

This potential is visualized in Fig. 1 with a contour plot, and shows two minima, one at $V(\{x = -8.9554, y = 0\}) = -4.78204$ and one at $V(\{x = 3.9767, y = 0\}) = -5.71887$. A saddle point is located at $V(\{x = 7.90805, y = 0\}) = 4.34544$. For clarity the 1D projection $\beta F(x) = -\ln \int dy \exp[-\beta V(x, y)]$ onto x is also shown, where the y variable is integrated out using Boltzmann averaging. The inverse temperature $\beta = 1/k_B T$, with T the reduced (artificial) temperature and k_B the Boltzmann constant, determines how rare the transition between minima is. In this 2D potential a single particle moves according to full Langevin dynamics. To integrate the equation of motion we employ the BAOAB algorithm described in Ref. 23. We set time step $\Delta t = 0.05$, and a friction of $\gamma = 1$. The temperature was $\beta = 10$, chosen to ensure a low rate constant. Particle configurations are saved each 10th time step. Frames in the trajectories are thus separated by $\Delta t_{frame} = 0.5$. The maximum path length is set to $L_{max} = 10^5$ frames.

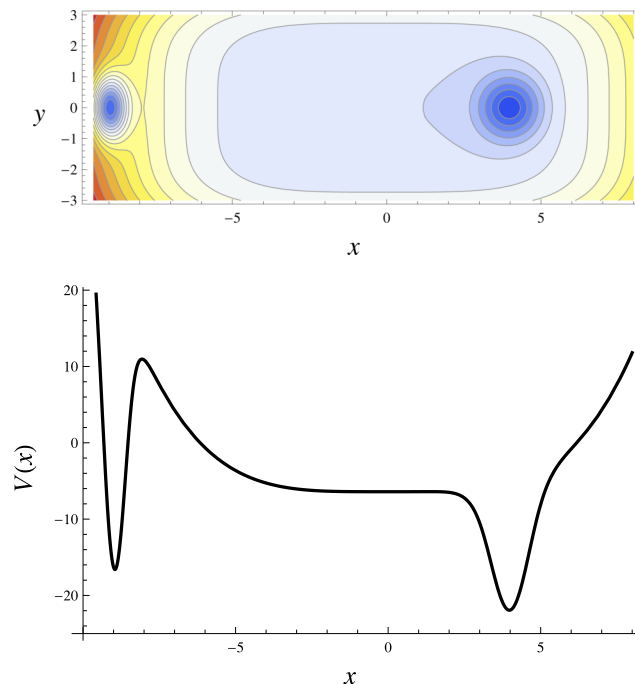


FIG. 1. Top: 2D Potential contour plot. Contours are separated by $1 k_B T$. Bottom: the Boltzmann integrated projection on the x -coordinate.

2. FF dimer

All atomistic molecular dynamics simulations as well as the system preparation for the FF dimer were performed with the Gromacs 4.5.4 package.²⁴ All atom interactions were defined using the amber99sb-ildn²⁵ and TIP3P force fields.²⁶ The FF segment was taken from the sequence KLVFFA from the amyloid-beta peptide (AB residues 16-21) (PDB 2Y29²⁷) and capped with ACE and NME to render, respectively, the N and C terminal neutral. Two copies of the FF monomer were put at a distance of 1 nm from each other in a cubic box of $30 \times 30 \times 30$ Å, and energy minimized using the conjugate gradient method. After solvation of the box and a second energy minimization, we performed a short equilibration of water (10 ps NPT in ambient conditions) with the peptide positions restrained. The solvated system was equilibrated for 1 ns in ambient conditions in the NPT ensemble and thereafter was subjected to a long 200 ns NPT simulation. All bonds were constrained with the Lincs algorithm. A cutoff of 1 nm was used for the non-bonded interactions. The particle mesh Ewald method treated the electrostatic interactions, with a Fourier spacing of 0.12 nm and a 1 nm cutoff for the short range electrostatic interactions. Neighbor lists were updated every 10 fs with a cutoff of 1 nm and the time step was 2 fs.²⁵ Newton's equations of motion were integrated with the leap-frog algorithm. In the NPT simulations the Bussi thermostat²⁸ with a coupling time constant of 0.2 ps controlled the temperature, while the Parrinello-Rahman barostat²⁹ with a coupling time constant of 1.0 ps kept the pressure constant.

An initial path was taken from the 200 ns MD run. TPS was performed with a home written script capable of applying both one-way uniform and spring shooting, generating paths of flexible length. In both types of shooting, the maximum path length L_{max} was set to 2000 frames, and

frames were saved every 5 ps. The TPS algorithm generated trajectories in ambient conditions ($T = 298$ K, $P = 1$ atm) in the NPT ensemble, with the above mentioned settings. Order parameters such as minimum distance and solvent accessible surface area (SASA) were obtained with the GROMACS analysis tools.

3. β -lactoglobulin dimer

All-atom molecular dynamics simulations as well as the system preparation for the β -lactoglobulin dimer were performed with Gromacs 4.6.7 package²⁴ using GPUs. All atom interactions were defined using the amber99sb-ildn²⁵ and TIP3P force fields.²⁶ The β -lactoglobulin (β -lac) dimer system was taken from the protein research databank (PDB 2AKQ). The dimer was put in a dodecahedral box and energy was minimized using the conjugate gradient method. After solvation of the box with 20 787 water molecules and a second energy minimization, we performed a short equilibration of water (10 ps NPT in ambient conditions) with the protein position restrained. The solvated system was equilibrated for 1 ns in ambient conditions in the NPT ensemble and thereafter was subjected to a long 200 ns NPT simulation. All other settings were as mentioned above in the FF case.

In the course of the 200 ns MD, the β -lactoglobulin dimer remained in its native bound state (see Fig. 15 of the Appendix). The native contacts were identified as those residue pairs that stayed within a minimum heavy atom distance of 0.4 nm for at least 90% in the 200 ns NPT trajectory. These native contact pairs are listed in Table I of the Appendix. Only 8 residue pairs are shown to fulfill the above criterion. These 8 residue pairs as well as four native hydrogen bonds (between backbone NH and CO of residues 146-150, 148-148, 150-146) characterized by Sakurai and Goto³⁰ define the stable native (contact) state (NC).

In order to initialize the TPS simulation we need a single path undergoing the dissociation. Therefore we performed a meta-dynamics simulation using the PLUMED package³¹ with the above MD settings. The collective variable for MetaD was the center of mass distance of the protein's CAs, using a Gaussian hill with a height of 0.4 kJ/mol and width of $\sigma = 0.1$ kJ/mol, which are deposited every 2 ps. The resulting trajectory of the meta-dynamics run indeed undergoes a dissociation event. The trajectory was saved every 20 ps. For completeness we give the heavy atom minimum distance of the proteins along the trajectory in the Appendix (see Fig. 16). From this biased dimer dissociation trajectory we launched from a particular configuration (frame 108, corresponding to 2140 ps along the trajectory) a series of 10 trial trajectories with random velocities. These trajectories were performed at ambient conditions in the NPT ensemble, using the settings mentioned above. The particular frame 108 turned out to have a reasonable chance to end in the native state (NC), as well as escape the native state. Two of these trajectories, ending in different stable states, were glued to yield the desired initial unbiased path. Ideally this trajectories would start in the native state and end in as in the unbound state (U), which we defined by a minimum distance between the two proteins larger than 1 nm. However, an MD trajectory

from this particular configuration was extremely unlikely to provide trajectories ending in both state NC and state U (see Fig. 17 in the Appendix), as the system is trapped in an intermediate state, where only a fraction of native contacts is intact, thus preventing the full dissociation to state U. We therefore decided to first sample pathways between the NC state and an intermediate state NN with only one native contact formed.

We performed TPS simulations of the native state to non-native state (NC \rightarrow NN) at ambient conditions ($T = 300$, $P = 1$ atm) in the NPT ensemble, using the same home written scripts as for the FF dimer TPS, using three different type of one-way shooting schemes, a uniform shooting, Gaussian bias and a spring shooting. The maximum path length for this transition was set to $L_{max} = 1500$ and frames were saved every 10 ps. For the Gaussian bias shooting, we implemented a Gaussian bias in the selection of the shooting point at NC = 4 with a width $\sigma = 1$. For the spring shooting move TPS, we performed simulations of $k = 0.1, 0.5$ and $\Delta\tau_{max} = 10, 30, 50, 70, 90$.

4. Analysis of the path ensemble

Home-written scripts analyzed the path sampling results to produce the path tree, the decorrelated path, the path length distribution, and the path density.¹⁵ We construct the path density by choosing two order parameters (e.g., dipeptide minimum distance and solvent accessible surface area for the FF dimer) and binning each frame of each trajectory in the path ensemble to a 2D grid. Every path can only contribute to a specific bin once, even if visited multiple times. Note that accepted paths can occur multiple times in the ensemble, depending on whether the next trial moves have been rejected.

III. RESULTS AND DISCUSSION

A. 2D Langevin model

Using an initial path constructed from two trajectories initiated at the saddle point, and ending at state A

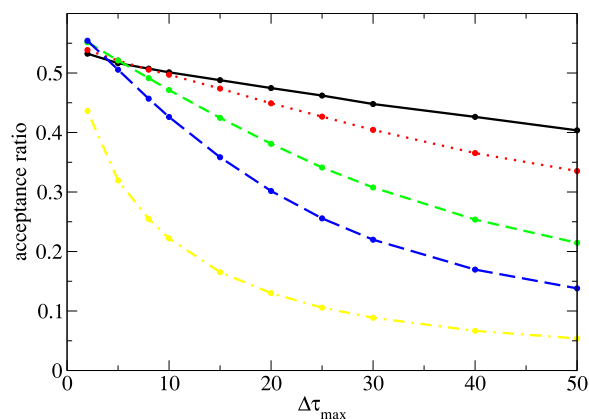


FIG. 2. Acceptance ratio as a function of maximum frame shift, for different spring constants. Curves correspond to $k = 0.01$ (black solid), 0.02 (red dotted), 0.05 (green dashed), 0.10 (blue long-dashed), 1.00 (yellow dotted-dashed), 5.00 (brown dotted-long dashed).

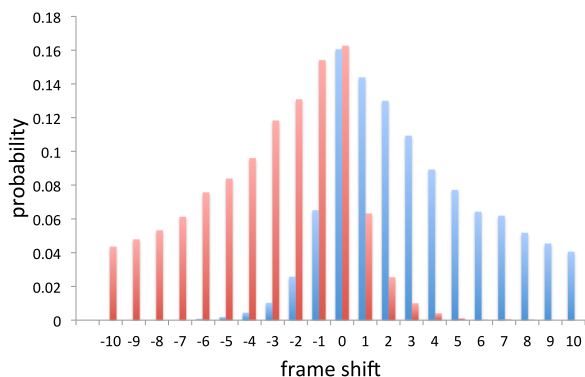


FIG. 3. Normalized histograms of the instantaneous frame shift for forward (red) and backward shots (blue) of the accepted paths for $\Delta\tau_{max} = 10$ and $k = 1.0$.

and B, respectively, we conduct Transition Path Sampling simulations with the spring shooting algorithm. We perform a series of runs with different spring constants $k = 0.01, 0.02, 0.05, 0.1, 1.0, 5.0$. For each value of k we perform a series of runs with frame shifts $\Delta\tau_{max} = 2, 5, 8, 10, 15, 20, 25, 30, 40, 50$. In all cases the number of trial shots is 10^6 . In addition we perform a simulation using uniform one-way shooting, and a simulation employing aimless shooting.

The overall shooting acceptances are shown in Fig. 2. As expected, the acceptance ratio goes down when the frame shift is increased. Moreover, increasing the spring constant also decreases the acceptance ratio, since the biasing potential will reject shooting point trials more often. Still the overall acceptance never drops below 5%.

As both the backward and forward shooting trial moves contribute to the overall shooting acceptance, it is insightful to separate the contributions of both these sub-moves. In Fig. 3 the acceptance of both moves is plotted as a function of the instantaneous frame shift for a $k = 1.00$ and maximum frame shift $\Delta\tau_{max} = 10$. Here one clearly sees that the backward move biases to positive frame shift, while the forward move biases to negative frame shift.

To show that the biasing potential in the spring shooting move has no influence on the resulting path ensembles, we plot the path length distribution of all simulations together in Fig. 4. As a reference we also plotted the uniform one-way shooting

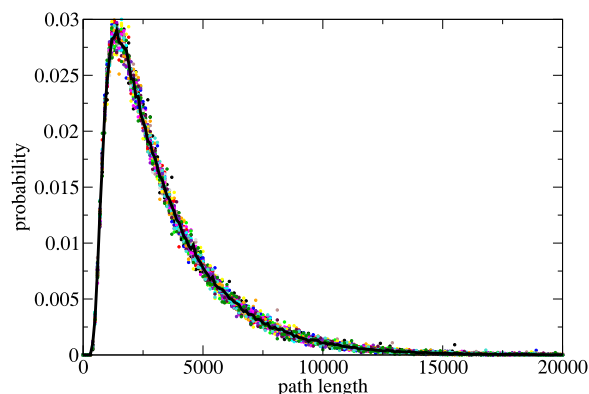


FIG. 4. Path length distribution for all 60 TPS simulations. The uniform one-way shooting result is shown as a solid line for reference.

result. Clearly, the distribution is the same for all settings, and only differs in accuracy due to sampling efficiency.

To establish the efficiency of the spring shooting algorithm, we need a measure of the decorrelation between paths. There is not a straightforward recipe for this. For the one way shooting algorithm a good measure is the amount of sampling that takes one from one decorrelated path to the other. This decorrelation path can be visualized using a path tree, see Fig. 5. This figure shows the path tree for a uniform one-way shooting algorithm. Starting from the top each horizontal line indicates an accepted shooting attempt. Red lines indicate forward shots, and green lines indicate backward shots. The thin black vertical lines indicate the shooting point location on the previous path. Each accepted new path thus consists of the newly formed green/red path, together with the complementary part of the previous path. The blue horizontal lines indicates the part of the paths that contributes to path decorrelation. Clearly, the longer the blue trajectory is, the better it is. Note also that a vertical jump in the blue line means that paths that do not contribute to this decorrelation are skipped. In this case whole green and red parts are skipped when the vertical lines are long, indicating that the uniform shooting algorithm might be not so efficient.

The blue decorrelation trajectory is the part of the path ensemble that is the least changed during the path sampling. This trajectory effectively performs a random walk on top of the (diffusive) barrier. Clearly, the longer it is, the more it is

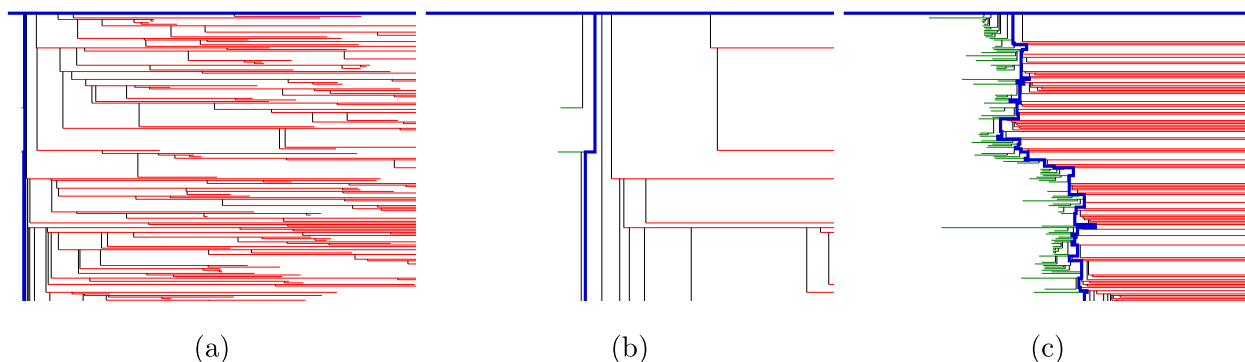


FIG. 5. (a) Path tree based on uniform shooting. Backward paths are indicated in green, forward in red. Forward paths had to be cut in order to fit in the figure. (b) Zoom in on the blue decorrelation path. Clearly, many forward shots were skipped, as they did not contribute. (c) Spring shooting tree shows much longer decorrelation path (blue). Also here forward paths had to be cut to fit in the figure.

able to explore path space, as subsequent paths are diffusing away from each other, eventually providing a representative sampling of the relevant path space. One might object that this choice of the decorrelation is rather arbitrary, and that it is more important to establish when the path ensemble becomes representative. Indeed, the decorrelation metric does not give information about how well the path ensemble is sampled. For instance, if the path ensemble oscillates between two narrow reaction channels, this will (most likely) not show up in the decorrelation measure. We stress, however, that this holds for any shooting algorithm. The aim of our analysis is to compare the efficiency of the different shooting algorithms. Besides, it is not straightforward to come up with a system independent measure of the representative sampling of the path ensemble. We therefore believe that the least changed path is a reasonable system independent measure for decorrelation, to establish the difference in efficiency of the different sampling algorithms.

Taking the (horizontal) length of the blue path as a measure for decorrelation, we establish this length for all spring method simulations. The results are given in Fig. 6(a). Here, clearly the decorrelation increases with the spring constant k . This is because the spring forces the shooting point into a direction that makes it more likely that the new trajectory contributes to the “blue” decorrelation path. Note that increasing the spring constant above $k = 1.0$ does hardly alter the decorrelation curve, as then basically all trial moves in the “wrong” direction will be rejected. For each spring constant the decorrelation curve behaves non-monotonically as a function of the maximum frame shift. While for small maximum frame shifts the decorrelation increases with $\Delta\tau_{max}$, the decorrelation reaches a maximum before starting to go down. This is because small maximum frame shifts are more likely to yield accepted new paths, but contribute only little, while large maximum frame shifts contribute much more to decorrelation, but have a smaller acceptance probability. Clearly there is a $\Delta\tau_{max}$ at which the decorrelation is maximized. This maximum shifts to smaller $\Delta\tau_{max}$ for larger spring constant k . Note that the change in maximum is not that large: from $\Delta\tau_{max} \approx 20$ at $k = 0.01$ to $\Delta\tau_{max} \approx 10$ at $k = 5$. From this plot the largest decorrelation occurs for $k \geq 1$ and $\Delta\tau_{max} \approx 10$. Of course, this optimum will be system specific.

Also shown in the plot is the decorrelation occurring in the uniform one way shooting algorithm (horizontal gray dotted line) and the aimless shooting method (horizontal purple dotted line). From this it seems that the aimless shooting always wins with respect to the spring shooting. This is because each successful aimless shot will contribute to the decorrelation, whereas in the spring shooting, there is still possibility to not contribute.

However, the path decorrelation alone is not the determining factor. Efficiency is also determined by CPU time. In Fig. 6(b) the CPU time for each simulation is plotted. Here the aimless shooting simulations are clearly the most expensive simulations, as for each shot both forward and backward paths need to be evaluated, which can both be rather long if the shooting point is on the shallow side of the asymmetric the barrier. The CPU time is already smaller

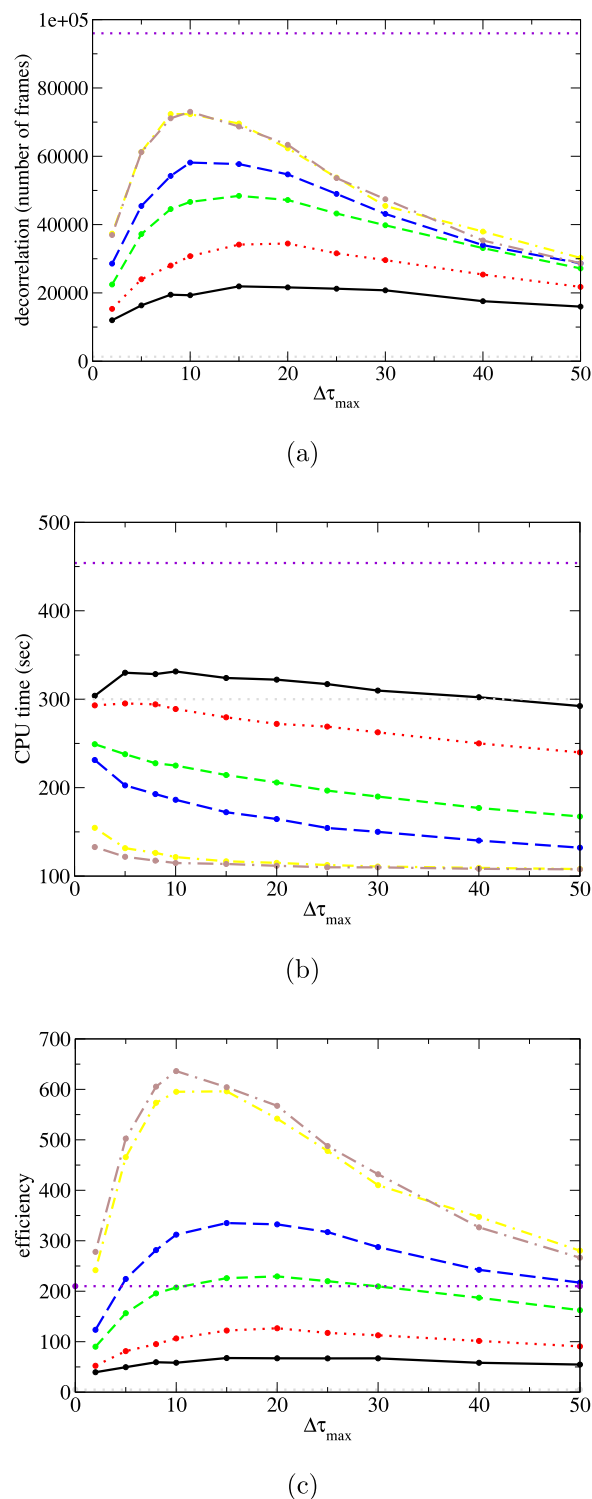


FIG. 6. Efficiency of shooting algorithms. (a) Decorrelated path length as a function of maximum frame shift for different spring constants. (b) CPU time in seconds. (c) Efficiency as the decorrelation per CPU time. The curves correspond to spring constants $k = 0.01$ (black solid), 0.02 (red dotted), 0.05 (green dashed), 0.10 (blue long-dashed), 1.00 (yellow dotted-dashed), and 5.00 (brown dotted-long dashed). Also shown are results for uniform shooting (gray horizontal dotted line), and for aimless shooting (purple dotted line).

for the uniform one-way shooting since then only half of the trajectory needs to be computed for every shot. The CPU time drastically decreases for the spring shooting move, becoming smaller with both k and $\Delta\tau_{max}$. For the high values of $k \geq 1$

the CPU is only 25% of the aimless shooting effort. This is caused by the early rejections, due to the spring bias, as well as due to the one-way nature of the shooting.

The most efficient algorithm shows the highest decorrelation per unit CPU time. In Fig. 6(c) we plot the ratio of the decorrelation and the CPU time. While the shape of the curves remain roughly as in Fig. 6(a), now the aimless shooting efficiency drops substantially. For $k > 0.05$ the spring shooting algorithm already becomes more effective than aimless shooting (for this set up), and for the $k = 5$, $\Delta\tau_{max} = 10$ case, the efficiency is more than 3 times that of aimless shooting and more than 100 times as efficient as the uniform shooting move.

To understand the origin of the differences between the efficiency of the different shooting algorithms we compare the distributions of the x-value of shooting points in Fig. 7. Clearly the uniform one-way algorithm shoots mostly from areas far away from the transition state, thus leading to low decorrelation, and large non-contributing stretches of partial paths. The aimless shooting on the other hand focuses on the transition state. However, it is more expensive as it is a two-way shooting method, failing more often than a one-way shooting. The spring shooting algorithms fall in between these two extremes, closer to the aimless shooting distribution for $k = 1$ than for $k = 0.05$. The conclusion is thus that one wants to be close to the aimless shooting distribution, which happens for relatively strong spring constant k , and not a too high maximum frame shift $\Delta\tau_{max}$.

An alternative way to identify the efficiency is to construct a correlation function of a property along the paths,

$$C(n) = \frac{\langle \delta g(0) \delta g(n) \rangle}{\langle \delta g(0) \delta g(0) \rangle}, \quad (11)$$

where n is the index of the shooting trial, the angular brackets denote average over the path ensemble taking each path as a time origin, and $\delta g(n) = g(\mathbf{x}^n) - \overline{g(\mathbf{x})}$, with the overbar denoting the mean over all paths. This leads to an exponentially decaying correlation function. We extract the decay time of this correlation function by establishing the shooting trial index at which the correlation function reaches a value of $1/e$.

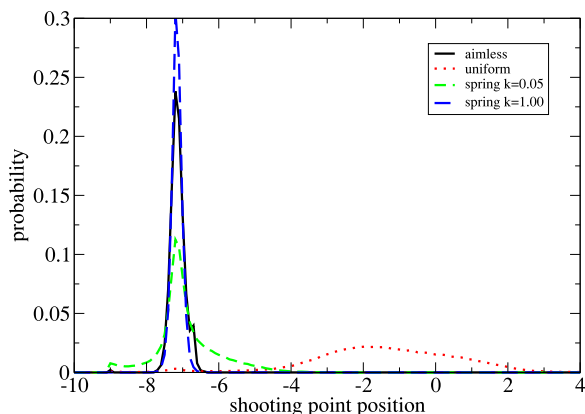


FIG. 7. Shooting point distribution as a function of the x-coordinate for the aimless shooting (black solid line), uniform shooting (red dotted line), spring shooting with constant $k = 0.05$ (green dashed line) and $k = 1.00$ (blue long dashed line). The aimless and spring algorithms are much more peaked, and closer to the saddle point compared to the uniform sampling.

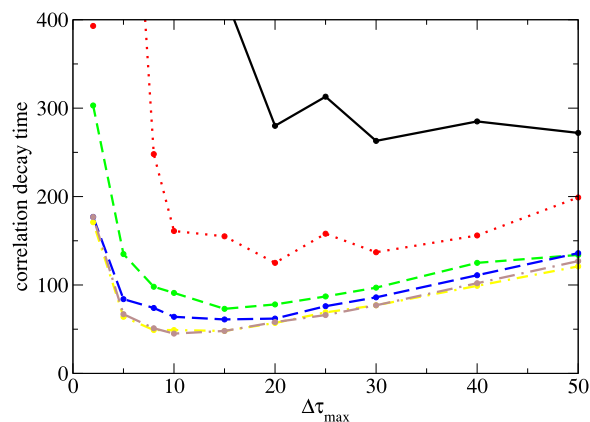


FIG. 8. Correlation decay time for the maximum energy along the path. Curves are for spring constants $k = 0.01$ (black solid), 0.02 (red dotted), 0.05 (green dashed), 0.10 (blue long-dashed), 1.00 (yellow dotted-dashed), and 5.00 (brown dotted-long dashed).

For the correlation function of the maximum energy along the path,^{9,32} the decay time is plotted in Fig. 8 for different k values as function of the maximum frame shift $\Delta\tau_{max}$. Similar to the decorrelated path length, this function shows also an optimum for $k > 1$ and $\Delta\tau_{max} = 10$.

B. FF dimer association

In the next illustrative example we focus on the association dissociation reaction between two phenylalanine dipeptides. Because of their hydrophobicity these dipeptides do attract each other, but also sometimes dissociate spontaneously, a state that is stabilized by entropy. We performed a 200 ns straightforward MD run, at 300 K and 1 atm. As shown in Fig. 9(a) in this time many transitions occur between the bound and unbound states, indicating that the binding unbinding is not a rare event on that time scale. Still, while the relaxation time is on the order of a ns, most of the transitions are rather sharp and occur on a time scale of around 100 ps, thus characterizing them as activated processes on the ps time scale. Fig. 9(b) shows the free energy, obtained as the negative logarithm of the probability distribution along the minimum distance order parameter r_{min} . Indeed this plot shows a well defined minimum at the bound state around $r_{min} = 0.2$ nm, and a very broad shallow minimum for the unbound state for $r_{min} > 0.6$ nm, separated by a small barrier at $r_{min} = 0.4$ nm. While the FE barrier is small, it is in principle asymmetric, with a steep binding force on the left side, and a shallow diffusive entropic force on the other side. The fact that the FE increases again after the minimum of the unbound state U has been reached is due to the finite size of the system. For larger system sizes the minimum would shift to larger distances. Moreover, the choice of the minimum distance as the order parameter also causes the free energy to increase already at smaller r_{min} than if we would have chosen the center of mass distance. We define the stable states solely on the minimum distance r_{min} , with the bound state B by $r_{min} \leq 0.22$ and the unbound state U by $r_{min} \geq 1.2$.

Here our aim is to test the efficiency of the spring method. We perform a series of runs employing the spring shooting

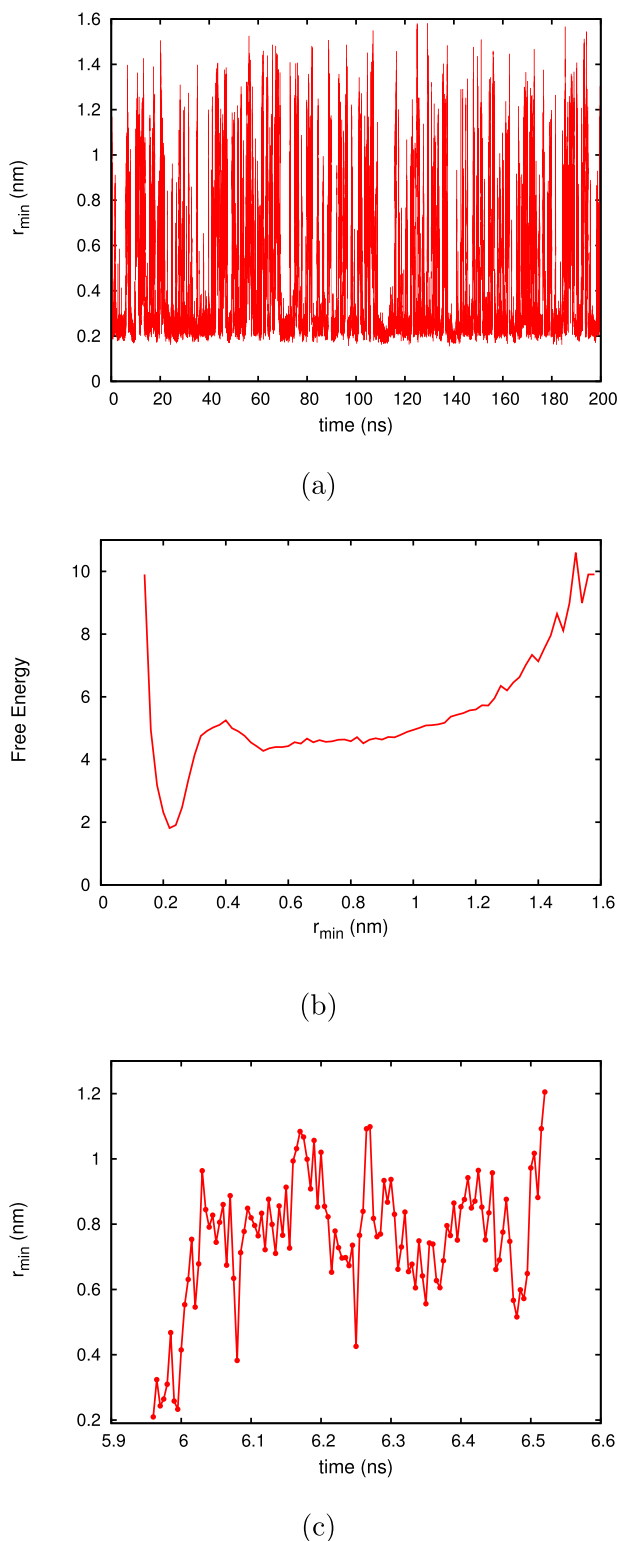


FIG. 9. (a) All atom FF dipeptide minimum distance time trace of 200 ns NPT MD, (b) free energy plot as a function of minimum distance, and (c) initial trajectory of the TPS, connecting bound (B) to unbound (U) state.

algorithm for different values of the spring constant k as well as for different $\Delta\tau_{\max}$. To initialize the TPS we extract a partial trajectory from the 200 ns straightforward MD run. This trajectory is plotted in Fig. 9(c) and connects state B ($r_{\min} \leq 0.22$) and U ($r_{\min} \geq 1.2$)

For the spring shooting move TPS, we performed simulations of multiple spring constants values $k = 0.1, 1$ and $\Delta\tau_{\max} = 10, 30, 50, 70, 90, 110, 150, 200, 250, 400$.

As mentioned above for the 2D toy model, the spring shooting move should eventually sample the same equilibrium path ensemble, independent of its parameters. Therefore, properties of the path ensemble such as the distribution of path lengths and path densities should also be independent of the employed shooting scheme. Fig. 10 illustrates that the path length distribution remains unchanged upon changing the spring constant k and $\Delta\tau_{\max}$'s, the equilibrium path distribution. The path length distribution is peaked at 200 ps and has an average path length of 340 ps. This distribution has a typical Poisson shape, with a long tail up to 2 ns.

At low spring $k = 0.1$ and $\Delta\tau_{\max} = 10$ the distribution deviates slightly from the equilibrium one (with a peak at $P = 0.03$ and not at 0.045). This is probable due to the slow equilibration of the path ensemble, as, since the maximum shift in shooting point position is small, more sampling is needed for sufficient diffusion of shooting points and therefore to generate all possible pathways of the equilibrium distribution.

As for the 2D toy model, the efficiency of the spring shooting method depends on the spring constant k and the maximum allowed shift $\Delta\tau_{\max}$. We define the efficiency of the spring shooting algorithm as the total decorrelated path length divided by the total number of computed MD steps (as a measure for the CPU time). For the FF dipeptide binding transition, the efficiency in terms of decorrelation length over the total number of MD steps of the TPS (including accepted and rejected forward or backward paths) is plotted in Fig. 11. The efficiency shows a maximum at $\Delta\tau_{\max} = 150$, corresponding to 750 ps path, since the frame saving frequency was 5 ps. Increasing the spring constant from $k = 0.1$ to $k = 1$ does not change the efficiency much, indicating that the efficiency is already saturated (see Fig. 6).

Fig. 11 shows that at small $\Delta\tau_{\max}$ the efficiency is small, which is attributed to the reduced decorrelation of new trajectories, caused by the small allowed shooting point shifts. Upon increasing $\Delta\tau_{\max}$ the efficiency increases as larger shooting point shifts are allowed in each trial move, and the chance of large contribution to decorrelation increases. The efficiency reaches its maximum around $\Delta\tau_{\max} \approx 150$ –200. In

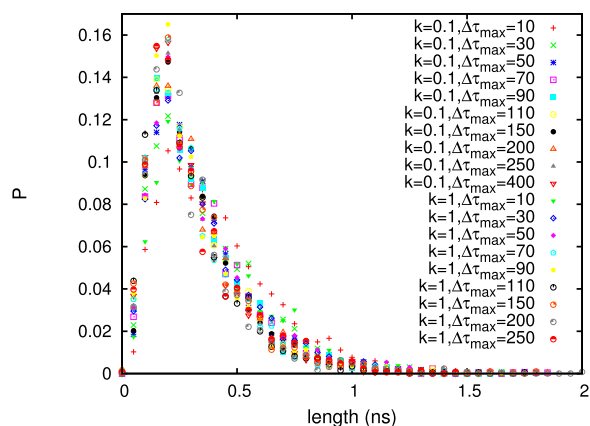


FIG. 10. Path length distribution for different spring constants k and $\Delta\tau_{\max}$.

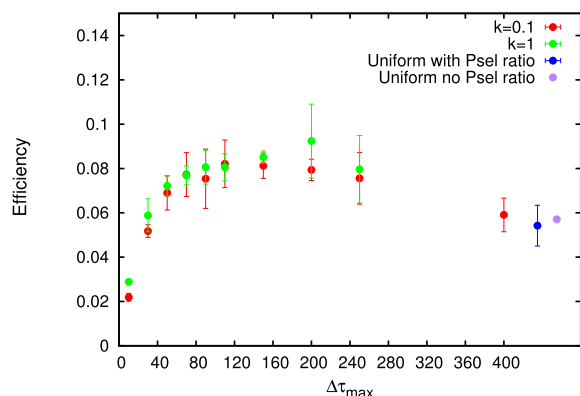


FIG. 11. Efficiency of spring shooting algorithm as a function of and $\Delta\tau_{max}$ for different spring constants $k = 0.1$ (red), 1 (green).

the limit of large $\Delta\tau_{max}$'s the shooting efficiency decreases again, and becomes comparable to the uniform shooting.

In the spring shooting move, after accepting the newly generated shooting point (Eq. (9)), the generated trajectory needs to end in the correct state in order for the trial move to be accepted within a reasonable amount of time, not exceeding a maximum time frames L_{max} . As stated Sec. II, the maximum time was set to 10 ns, much longer than the path lengths occurring the sampled path ensemble.

The comparison between the efficiency uniform and spring shooting algorithm might be influenced by the fact that uniform shooting rejects paths based on the ratio of the path lengths, whereas the spring move does not. Therefore, when comparing the efficiency of the spring shooting with the uniform shooting, we report both the efficiency for the regular uniform shooting scheme (“uniform with Psel ratio”) as well as the efficiency for the uniform shooting where this rejection step is omitted (“uniform no Psel ratio”). The latter shooting scheme does not fulfill detailed balance and was implemented only to compare to the spring shooting algorithm. One would expect the uniform shooting without the path length ratio comparison to be much more efficient than the uniform shooting with the criterion as the latter rejects more trajectories ($P_{acc} = 40\%$) because of the length criterion compared to the former ($P_{acc} = 45\%$) leading to

less decorrelation per CPU-time. However, we find that only 30% of the rejected paths are rejected due to the length criterion. When the length criterion is off, the acceptance ratio drops only by 10%. From this we deduce that the trajectories rejected because of the length criterion, would result in a rejection anyway, by ending up in the wrong states. Indeed, the uniform shooting incorporating the length criterion has only a marginally smaller efficiency than the uniform shooting without the length criterion.

From the sampled path ensemble we can construct path density plots, shown in Fig. 12. The different spring shooting parameters do not alter the equilibrium properties of the transition. To show that we plotted the difference in distribution in Fig. 12(c). The difference is of the order of 10%. The bound state (B) is located below a minimum distance of 0.22 nm between the dipeptides. In the bound state there exist configurations of different hydration, i.e., ones with solvent accessible surface below 8.6 nm^2 (see Fig. 12(b) bottom left) and others with higher SASA. The unbound state (U) is characterized by a minimum dipeptide distance larger than 1.2 nm from each other. Here the solvent accessible surface area (SASA) is around 10.5 nm^2 . The L shape of the path density plot indicates that when the dipeptides approach each other below a certain threshold, the SASA starts to decrease, as expected for an association reaction. From the unbound state (top right area of the path density plot of Fig. 12) the peptides bind non-specifically through on-pathway intermediates with a SASA in the range of $9.5\text{--}10.5 \text{ nm}^2$, highlighted on the lower left side of the path densities Fig. 12. Thereafter, they reach a desolvated bound state (B) which has a low SASA ($<8.6 \text{ nm}^2$) and is characterized by single phenyl-phenyl ring interactions, or both phenyl-phenyl interactions when peptides align in a parallel fashion, or by alignment of the peptides through hydrogen bond/hydrophilic interaction stabilization. The path density plots also suggest a dynamical bottleneck for the association located at $r_{min} = 0.4 \text{ nm}$ and a SASA of 10.5 nm^2 .

C. Dissociation of β -lactoglobulin dimer

As for the FF dimer system the spring shooting algorithm is only marginally more efficient than the uniform shooting, in

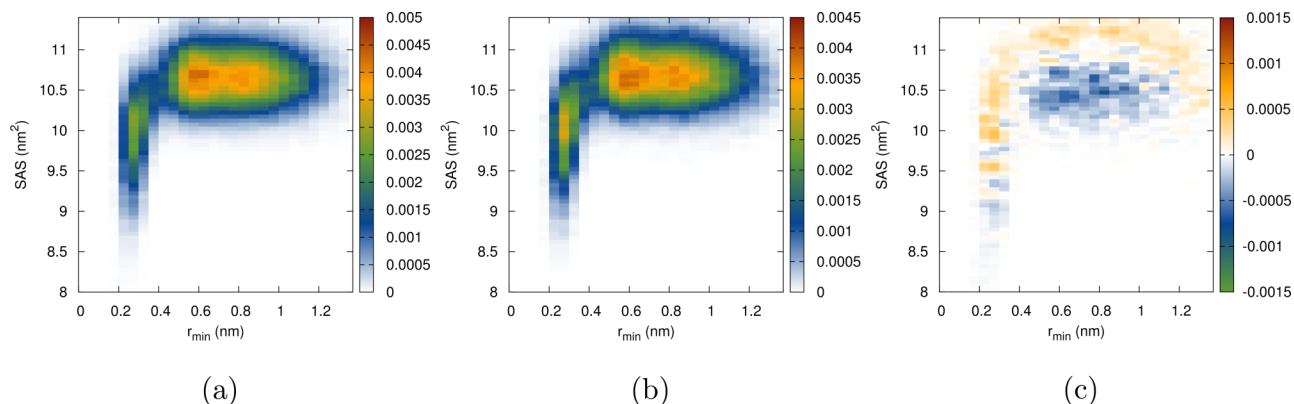


FIG. 12. Path density plot of all atom dipeptide minimum distance versus solvent accessible surface for (a) $k = 0.1$, $\Delta\tau_{max} = 10$, (b) $k = 0.1$, $\Delta\tau_{max} = 100$. (c) Difference between these path density plots.

this subsection we consider a more complex protein-protein dissociation problem. We performed TPS on the first step in the dissociation process of the β -lactoglobulin dimer, the transition from the native state (NC) to an intermediate state (NN). The TPS simulation was initialized from a predefined dissociating path (see Sec. II).

Here, we discuss only the difference in sampling of the uniform shooting, the Gaussian bias shooting, and the spring shooting, in terms of their sampling trees, shooting points, and efficiency. A more detailed molecular interpretation of the TPS results will appear elsewhere.

In Fig. 13, we plot the path tree for the NC \rightarrow NN transition. Although the sampling is not exhaustive, clearly the uniform shooting algorithm generates a higher abundance of accepted backward paths. This corresponds to a systematic selection of shooting points closer to the NC state (see Fig. 14). This is caused by the presence of an asymmetric barrier where the rate limiting step takes place for relatively low number of contacts, and the dimer spends long parts of the backward trajectory in near native states before finally locking into native bound state (NC).

The spring shooting move ameliorates this effect and nearly equally creates both forward and backward paths. This is further corroborated by the shooting point as a function of accepted trials, where the selected shooting points are not driven by the natural entropy of the system, close to the native state, but are wandering to the middle of the transition. The efficiency, expressed as the decorrelation length over number of saved frames, of the uniform shooting move, the Gaussian bias shooting move and the spring shooting move of $k = 0.5$ and $\Delta\tau_{max} = 70$ is 0.00066, 0.01699, and 0.0445 respectively. Clearly the Gaussian bias shooting move and the spring shooting are two orders of magnitude more efficient than the uniform shooting. Comparing the Gaussian bias with

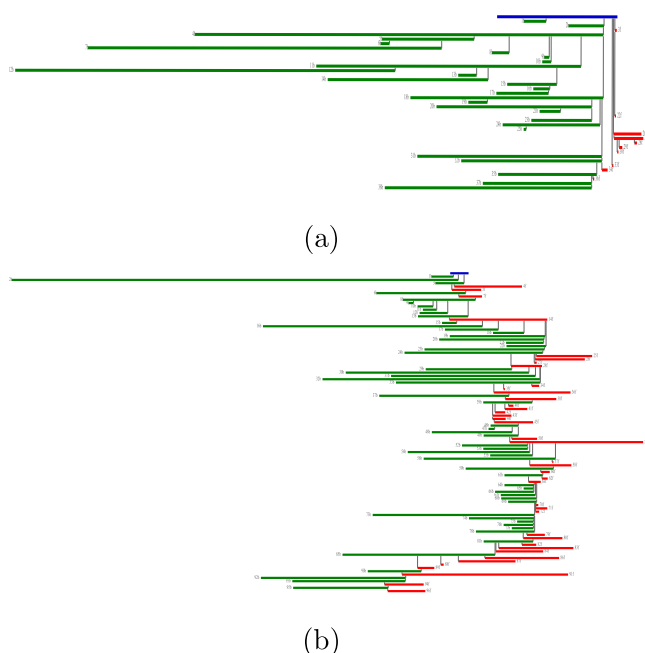


FIG. 13. Tree for the NC \rightarrow NN transition using (a) uniform shooting, (b) spring shooting for $k = 0.5$ and $\Delta\tau_{max} = 70$.

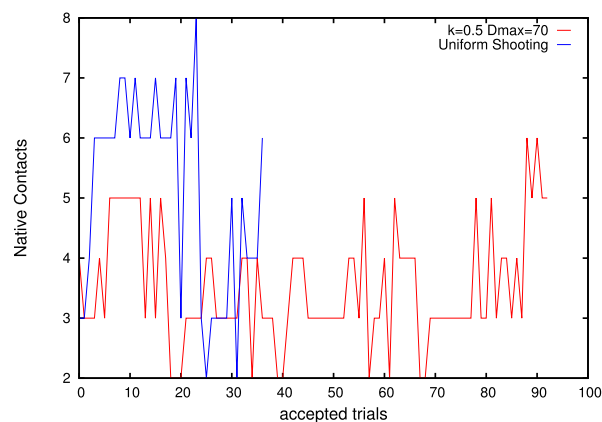


FIG. 14. Number of native contacts of the shooting points of accepted reactive NC \rightarrow NN trajectories sampled by TPS, using spring shooting for $k = 0.5$ and $\Delta\tau_{max} = 70$ (red) and uniform shooting (blue).

the spring shooting of $k = 0.5$ and $\Delta\tau_{max} = 70$, we find that the spring shooting is about twice more efficient compared to the Gaussian bias shooting. Note that the choice of k and $\Delta\tau_{max}$ might be not optimal, as no effort has been made to maximize efficiency for this transition.

IV. CONCLUSION

We introduced a novel shooting algorithm for the transition path sampling framework that is aimed at efficiently sampling transitions in complex systems with diffusive behavior and asymmetric barriers. We showed that the new spring shooting is more efficient than uniform shooting, and also than the aimless shooting algorithm. In addition, the new algorithm might also be useful to other path sampling methods such as transition interface sampling,¹⁴ although there the efficiency gain is probably smaller because of the use of interfaces. Note however that the efficiency analysis is not a general statement of superiority. There will be many cases in which other methods are more efficient. Nevertheless we believe that this new algorithm will help to improve the sampling of rare events in complex systems such as ligand binding and protein association reactions.

ACKNOWLEDGMENTS

The authors thank David H. Swenson for the critical reading of the manuscript. This work is supported by NanoNextNL, a micro- and nanotechnology consortium of the Government of the Netherlands and 130 partners. We acknowledge the support from the Nederlandse Organisatie voor Wetenschappelijk Onderzoek (NWO) for the use of supercomputer facilities.

APPENDIX: INITIALIZATION OF THE β -LACTOGLOBULIN DIMER PATHS

In this section we provide additional information on the β -lactoglobulin dimer simulations. Table I lists the contacts found in the native dimer, together with their occupancy, based on the 200 ns trajectory. Here we defined a contact when the

TABLE I. Occupancy of residue contacts within the course of a 100 ns NPT MD.

Residue pair	Occupancy (%)
150-146	100
148-148	100
146-150	100
148-147	99.98
147-148	99.84
149-146	99.56
146-149	99.44
33-33	90.74
150-29	79.46
29-150	78.66
137-141	75.96
151-29	70.66
141-137	69.78

minimum heavy atom distance was less than 4.0 Å. The first 8 pairs are considered the native contact pairs. Fig. 15 plots the number of native contacts and the hydrogen bonds of the native dimer state in the 200 ns NPT trajectory.

Fig. 16 shows the minimum distance between the β -lactoglobulin proteins in a metadynamics simulation. Fig. 17

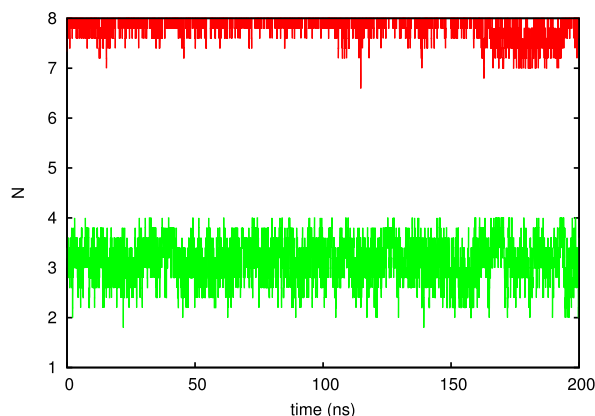


FIG. 15. The number of native contacts (red) and native hydrogen bonds (green) along a 200 ns NPT trajectory at ambient conditions.

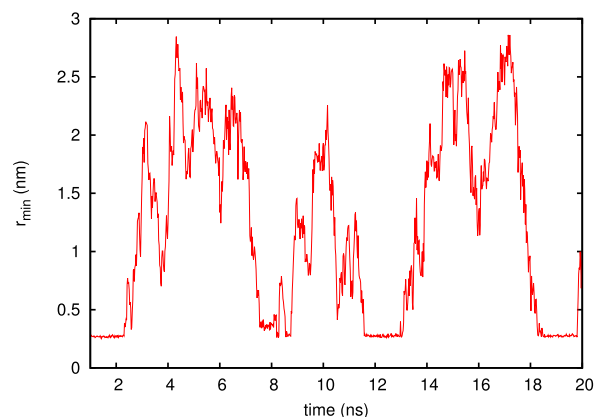


FIG. 16. Heavy atom minimum distance between the two proteins along a 20 ns meta-dynamics simulation.

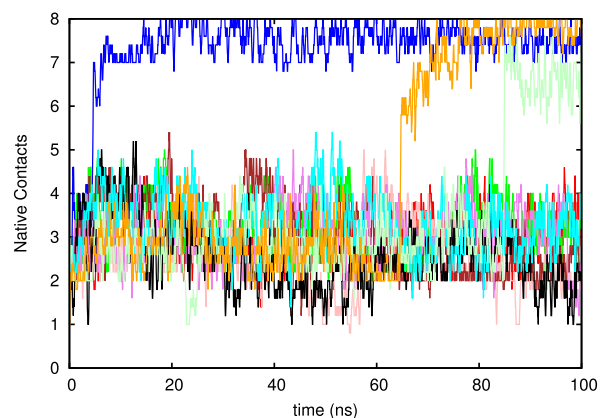


FIG. 17. The number of native contacts as a function of time for a series of ten 100 ns trajectories starting from the same configuration but with different momenta.

shows the number of native contacts for the ten simulations starting from frame 108 of the metadynamics run.

- ¹G. M. Torrie and J. P. Valleau, *Chem. Phys. Lett.* **28**, 578 (1974).
- ²H. Grubmüller, *Phys. Rev. E* **52**, 2893 (1995).
- ³T. Huber, A. Torda, and W. van Gunsteren, *J. Comput.-Aided Mol. Des.* **8**, 695 (1994).
- ⁴E. Darve and A. Pohorille, *J. Chem. Phys.* **115**, 9169 (2001).
- ⁵A. Laio and M. Parrinello, *Proc. Natl. Acad. Sci. U. S. A.* **99**, 12562 (2002).
- ⁶Y. Sugita and Y. Okamoto, *Chem. Phys. Lett.* **314**, 141 (1999).
- ⁷C. Dellago, P. G. Bolhuis, F. S. Csajka, and D. Chandler, *J. Chem. Phys.* **108**, 1964 (1998).
- ⁸P. G. Bolhuis, D. Chandler, C. Dellago, and P. L. Geissler, *Annu. Rev. Phys. Chem.* **53**, 291 (2002).
- ⁹C. Dellago, P. G. Bolhuis, and P. L. Geissler, *Adv. Chem. Phys.* **123**, 1 (2002).
- ¹⁰C. Dellago and P. G. Bolhuis, *Top. Curr. Chem.* **268**, 291 (2007).
- ¹¹C. Dellago and P. G. Bolhuis, *Adv. Polym. Sci.* **221**, 167 (2008).
- ¹²P. G. Bolhuis and C. Dellago, *Rev. Comput. Chem.* **27**, 111 (2010).
- ¹³G. Crooks and D. Chandler, *Phys. Rev. E* **64**, 026109 (2001).
- ¹⁴T. S. van Erp, D. Moroni, and P. G. Bolhuis, *J. Chem. Phys.* **118**, 7762 (2003).
- ¹⁵J. Juraszek and P. G. Bolhuis, *Biophys. J.* **95**, 4246 (2008).
- ¹⁶B. Peters and B. L. Trout, *J. Chem. Phys.* **125**, 054108 (2006).
- ¹⁷C. Dellago, P. G. Bolhuis, and D. Chandler, *J. Chem. Phys.* **108**, 9236 (1998).
- ¹⁸P. G. Bolhuis, *J. Phys.: Condens. Matter* **15**, S113 (2003).
- ¹⁹P. G. Bolhuis, *Biophys. J.* **88**, 50 (2005).
- ²⁰J. von Neumann, *Collective Works* (Pergamon Press, Oxford, 1963), Vol. V.
- ²¹C. Robert and G. Casella, *Monte Carlo Statistical Methods* (Springer-Verlag, New York, 2004).
- ²²Note that here the limit is only taken after the construction of the acceptance criterion. An analogous situation arises when deriving the acceptance ratio for a regular translational MC move (in an external field). The distribution function is an integral over the (possibly infinite) space, but since in the Metropolis acceptance criterion the two distributions have the same normalization constant, the ratio remains finite. Note also that the choice of the $\Delta\tau_{max}$ is independent of b . The $\Delta\tau_{max}$ is analogous to the maximum allowed displacement in a regular MC translational move.
- ²³B. Leimkuhler and C. Matthews, *J. Chem. Phys.* **138**, 174102 (2013).
- ²⁴S. Pronk, S. Páll, R. Schulz, P. Larsson, P. Bjelkmar, R. Apostolov, M. R. Shirts, J. C. Smith, P. M. Kasson, D. van der Spoel, B. Hess, and E. Lindahl, *Bioinformatics* **29**, 845 (2013).
- ²⁵K. Lindorff-Larsen, S. Piana, K. Palmo, P. Maragakis, J. L. Klepeis, R. O. Dror, and D. E. Shaw, *Proteins: Struct., Funct., Bioinf.* **78**, 1950 (2010).
- ²⁶W. L. Jorgensen, J. Chandrasekhar, J. D. Madura, R. W. Impey, and M. L. Klein, *J. Chem. Phys.* **79**, 926 (1983).
- ²⁷J. Colletier, *Proc. Natl. Acad. Sci. U. S. A.* **108**, 16938 (2011).
- ²⁸G. Bussi, D. Donadio, and M. Parrinello, *J. Chem. Phys.* **126**, 014101 (2007).
- ²⁹M. Parrinello, *J. Appl. Phys.* **52**, 7182 (1981).
- ³⁰K. Sakurai and Y. Goto, *J. Biol. Chem.* **277**, 25735 (2002).
- ³¹M. Bonomi, D. Branduardi, G. Bussi, C. Camilloni, D. Provasi, P. Raiteri, D. Donadio, F. Marinelli, F. Pietrucci, R. A. Broglia, and M. Parrinello, *Comput. Phys. Commun.* **180**, 1961 (2009).
- ³²C. Dellago, P. Bolhuis, and D. Chandler, *J. Chem. Phys.* **110**, 6617 (1999).