Enhanced path sampling using subtrajectory Monte Carlo moves

Daniel T. Zhang,[1] Enrico Riccardi,[2] and Titus S. van Erp[1]
[1)] *Norwegian University of Science and Technology, Department of Chemistry, NO-7491 Trondheim, Norway*
[2)] *Department of Informatics, UiO, Gaustadalléen 23B, 0373 Oslo, Norway*

(Dated: 20 September 2022)

Path sampling allows the study of rare events like chemical reactions, nucleation, and protein folding via a Monte Carlo (MC) exploration in path space. Instead of configuration points, this method samples short molecular dynamics (MD) trajectories with specific start- and end-conditions. As in configuration MC, its efficiency highly depends on the types of MC moves. Since the last two decades, the central MC move for path sampling has been the so-called shooting move in which a perturbed phase point of the old path is propagated backward and forward in time to generate a new path. Recently, we proposed the subtrajectory moves, stone-skipping (SS) and web-throwing (WT), that are demonstrably more efficient. However, the one-step crossing requirement makes them somewhat more difficult to implement in combination with external MD programs or when the order parameter determination is expensive. In this article, we present strategies to address the issue. The most generic solution is a new member of subtrajectory moves, wire fencing (WF), that is less thrifty than the SS, but more versatile. This makes it easier to link path sampling codes with external MD packages and provides a practical solution for cases where the calculation of the order parameter is expensive or not a simple function of geometry. We demonstrate the WF move in a double-well Langevin model, a thin film breaking transition based on classical force fields, and a smaller ruthenium redox reaction at the ab initio level in which the order parameter explicitly depends on the electron density.

I. INTRODUCTION

Rare event simulation techniques aim to sample events that require an exceedingly long CPU/wall time to be simulated with standard molecular dynamics (MD). In classical full atom simulations of protein folding, for example, the longest reported[1] MD runs generated by the special-purpose molecular dynamics Anton 1 supercomputer are around 1 ms, allowing the study of fast-folding proteins. The most recently released Anton 3 supercomputer is even able to generate 100 μs/day for a million atom system.[2] Despite this remarkable speed, it is still not fast enough to study the folding of all proteins. For instance, the tryptophan synthase β_2 subunit has an experimentally measured[3] folding rate of $k_f = 0.001$ s^{-1}. Hence, the protein needs on average 1000 seconds to fold. The Anton 3 computer would thus need 27,379 wall time years to generate one single transition. For ab initio MD (AIMD) the situation is even worse as the quantum mechanical force evaluation is orders of magnitude slower than computing the gradient of a classical force field potential. In addition, no special purpose AIMD computers exist today.

Yet, rare event simulations allow the calculation of rate constants and the study reaction mechanisms orders of magnitude faster than MD, oftentimes without sacrificing any molecular-level resolution.[4] (Replica exchange) transition interface sampling (RE)TIS[5,6] is such a method that exploits the idea of transition path sampling (TPS)[7] to focus the CPU time on the actual barrier crossing event via a Monte Carlo (MC) sampling of MD paths.

RETIS and TIS employ a series of path sampling simulations, each sampling a different path ensemble. The path ensembles differ with respect to a minimal progress requirement, i.e. the number of interfaces (defined by fixed values of the reaction coordinate/order parameter) that has to be crossed.[8] Combining the results of all path ensembles allows the computation of rate constants and other properties with an exponentially reduced CPU time compared to a single MD simulation.

For instance, a classical simulation study on methane hydrate formation[9] using TIS and RETIS reports on a crystallization rate of 10^{-17} nuclei per second per simulation volume. In other words, in a system as small as those used in atomistic simulations, the process for forming a single critical nucleus takes physically 3 years. Naturally, the hypothetical wall time for reaching this with MD is astronomical for any supercomputer. Likewise, RETIS simulations[10] reproduced the rate constant of water dissociation at the AIMD level in reasonable agreement with experiments suggesting it happens once per 11 hours for each water molecule.[11,12] As it required 30 minutes to produce 1 ps MD time in the 32 water molecules system, a naive straightforward AIMD approach would need 0.7 billion centuries of wall time to generate a single dissociation.

Despite being orders of magnitude times faster than plain MD, simulations like the above are still computationally expensive and can require months to years to obtain satisfactory statistical accuracy. A further increase in efficiency is therefore desirable. There are essentially three approaches to achieve this: i) reducing the cost of the MC moves, ii) reducing the number of required trajectories, and iii) parallelization of the algorithms. Partial path sampling (PPTIS)[13] and milestoning[14] can be viewed as realizations of i) by sampling more restrictive path ensembles with a reduced average path length. Unfortunately, this introduces additional approximations. Strategies ii) and iii), on the other hand, allow for a

speed-up while still producing exact results, identical to those from hypothetical unattainably long MD simulations. In fact, RETIS successfully employs strategy ii) by complementing the shooting moves with replica exchange moves between path ensembles. RETIS is thus more CPU efficient compared to TIS. However, regarding strategy iii), TIS has the advantage that path ensembles can be run in parallel completely independently, while replica exchange moves require the progress of the sampling in the path ensembles to be synchronized such that processing units do not have to wait for each other. As a result, RETIS might not always outperform TIS based on wall time, which is the reason why the previously mentioned hydrate formation study was partly based on TIS.[9] The recently introduced ∞RETIS algorithm[15] is expected to solve this issue for future studies based on a fundamentally new replica exchange technique for cost-unbalanced replicas.

In fact, ∞RETIS implicitly applies the cost-free replica exchange moves an infinite number of times after each shooting move. Still, replica exchange moves alone are not ergodic and should, therefore, only be used in combination with another MC move like shooting.[16] To further push strategy ii), the principle MC move should be changed to reduce both the rejection rate and the resemblance between accepted paths. This is exactly what subtrajectory moves aim to establish. These MC moves resemble PPTIS[13] and milestoning[14] in the sense that they evolve via series of shorter paths (subtrajectories/subpaths), but differently to those methods, these subpaths are just intermediates between sampled paths that are extended to their full lengths. Sampled paths, therefore, have no configuration point in common with the previous path and the statistical inefficiency is typically reduced by a factor equal to the number of intermediate subtrajectories. So while the creation of a full new path becomes more expensive, this is more than offset by the fact that far fewer trajectories are needed to achieve a certain statistical accuracy. In addition, the approach can be combined with a high-acceptance protocol, which minimizes the number of rejections. As a result, most path ensembles obtain a nearly 100% acceptance.[17]

The two moves presented in Ref. 17, stone skipping (SS) and web throwing (WT), however, have one element, the one-step crossing condition, which can hinder the practical implementation with external MD programs or when the calculation of the order parameter is computationally expensive. In SS and WT, the subtrajectories are launched from a configuration point of a previous (sub) path that is just before or after the path ensemble's interface. At this configuration point, velocities are generated such that the interface is crossed again within a single time step. The velocity randomization and one-step crossing test is reiterated several times until the condition is fulfilled. The procedure is based on the idea that generation of new random velocities followed by a one-step crossing test is relatively cheap compared to generating MD steps, especially if the test can be performed without new force calculations. This might not always be the case. Present path sampling codes[18–21] use external MD codes for performing the MD steps. PyRETIS version 2 has for instance couplings to Gromacs,[22] Lammps,[23] openmm,[24] and CP2K.[25] In order to reduce the number of stop/restart calls to these programs, a "time step" in the RETIS program is often several (10-1000) MD steps by the external MD engine. This complicates the one-step crossing condition as it actually involves not one, but several steps which is costly and not easy to predict without actually performing these steps. Another issue arises when the calculation of the order parameter is expensive such as those used in nucleation studies.[26,27]

In this article, we discuss several approaches to tackle this issue. The most generically applicable solution is a new member of the subtrajectory family called wire fencing (WF). The approach is slightly more wasteful with respect to the number of MD steps compared to SS, but very versatile and does not require any code modifications of the external engines. We illustrate the WF move on three model systems, a simple 1D double-well potential, a Gromacs thin film breakage application, and a CP2K study on ruthenium redox reactions.

II. SUBTRAJECTORY MOVES

The schematic main idea of the three subtrajectory moves is shown in Fig. 1. These are the stone skipping

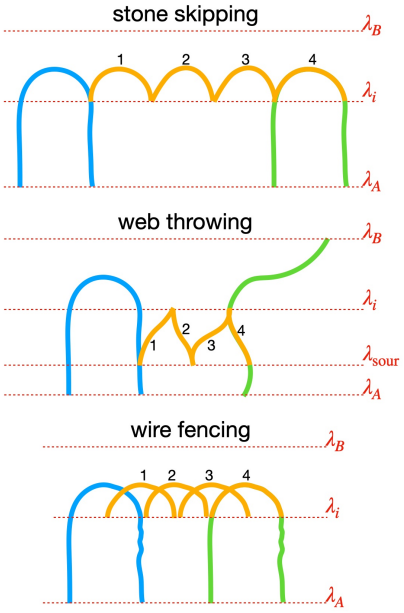


FIG. 1. Cartoon representation of the three subtrajectory moves: stone skipping, web throwing, and wire fencing. The old path is shown in blue. Four subtrajectories are shown in orange. The final new path consists of the fourth subtrajectory and its extensions colored in green.

(SS), web throwing (WT), and the new wire fencing (WF) move. The commonality is that an arbitrary number of partial trajectories (subtrajectories/subpaths) are generated before the completion of a new full trajectory. The subtrajectories obey different start- and end-conditions and are, due to this, considerably shorter than full trajectories. The subtrajectories are not part of the sampling, but are just intermediate steps between one full trajectory to another. The $[i^+]$ path ensemble that is being sampled in Fig. 1 consists of paths starting at λ_A, crossing λ_i at least once, and ending at either λ_A or λ_B. In Fig. 1, the old full trajectory is colored blue. In the example, the new trajectory is generated via 4 subtrajectories. The first subtrajectory is obtained from a shooting move from the old trajectory. Then, the next subtrajectory is generated from the previous one until the number of predetermined subtrajectories (4 in this case, colored in orange) is reached. The final subtrajectory is extended backward and forward in time until reaching a stable state. The new full trajectory comprises the last subtrajectory and the extensions colored in green. The difference between the three moves lies in the way the shooting of subpaths is executed.

The SS move resembles a flat stone that collides with the water's surface after a skillful throw. The move starts by selecting randomly any of the crossing points of the old path with λ_i, generates new velocities that also establish a crossing, and then proceeds until λ_B is crossed or λ_i is crossed again. The process is then repeated by selecting the subpath's last crossing with λ_i for shooting off the next subpath. Finally, the last subpath is extended and possibly accepted or rejected.[17]

The WT move has been named after a gesture of the famous Marvel character swinging between skyscrapers. Here, an additional interface needs to be defined, *the surface of unlikely return* (SOUR), at the state A side of the λ_i interface. If this interface, λ_{sour}, is crossed towards the direction of state A, it is assumed to be highly unlikely that the MD trajectory will end up in state B rather than A (defined by the last interface, λ_B, and the first interface, λ_A, respectively). The first subpath is then shot from a random crossing point with either λ_i or λ_{sour} at a path segment of the old path that connects these two interfaces. After the velocities of the system's atoms are re-set, like in the SS move, the subpath is continued till λ_{sour} or λ_i is crossed, but is only kept if the subpath connects λ_{sour} and λ_i again like the segment of the old path. If not, the subpath is rejected and a new crossing point is taken randomly from the same segment. If both λ_{sour} and λ_i are crossed, the subpath replaces the segment. The process is repeated until the selected number of subpaths, accepted or rejected, has been completed. The final accepted subpath is extended in both time-directions to make a full new path. Note that a rejection of a subpath does not imply a rejection of the MC move itself, but just redirects the process of achieving a new path from an old path. The time-direction is chosen such that from λ_{sour} the trajectory is propagated backward in time and from λ_i forward in time. Due to the placement of λ_{sour}, it is nearly guaranteed that the backward extension reaches state A. As λ_i is also crossed, it is ensured that the path is valid for the $[i^+]$ ensemble, though it might still be rejected due to a final acceptance/rejection step, as required by detailed balance.[28]

The WF move, further discussed in Sec. VII, differs with the other moves by its location of the shooting points. In the WF move, these might be any point with a corresponding value of the reaction coordinate that is larger than λ_i and lower than λ_B (or λ_{cap} if a so-called *cap interface* is set, see Sec. VII). From this point, no specific requirements are needed for the velocities so that they are most conveniently generated from a Maxwell-Boltzmann distribution for the temperature of interest. From the new phase point, MD steps are generated forward and backward in time until λ_B (or λ_{cap}) or λ_i is crossed. The subpath is accepted unless it reaches λ_B (or λ_{cap}) in both time-directions. In that case, it would be rejected and the next shot is taken again from the latest accepted subpath or the previous segment of the old path if no accepted subpaths yet exist. After finishing the number of desired subpaths, the last accepted one is extended to the stable states, like in SS and WT. While the WF move is slightly more wasteful with respect to the MD moves compared to SS, the velocity generation is much simpler which can have both practical and fundamental advantages compared to SS and WT. These are further discussed in Sec. VI. The name of the WF move is derived from the visual resemblance between the set of full paths and subpaths and the top of a wire fence.

The subtrajectory moves go against strategy i) as these MC moves require more MD steps than just the number of MD steps for generating a new path. These moves are nevertheless more efficient because they utilize strategy ii): the statistical inefficiency of the sampling is reduced and, therefore, fewer trajectories are needed to achieve a desired statistical error. Like with the standard shooting move, a final acceptance/rejection step should ensure that the correct statistical distribution of paths is sampled. However, due to the complexity of the subtrajectory move, the design and mathematical validation of the acceptance rule is substantially more complex and is derived from the so-called superdetailed balance[29] principle.

III. SUPERDETAILED BALANCE

The term superdetailed balance was first introduced within the context of configurational bias MC (CBMC),[29–31] which is an effective method to study the adsorption of polymers in nanoporous materials such as zeolites. In this algorithm, polymers are removed and then regrown atom by atom such that any overlap between the polymer and the zeolite's walls and other polymers is avoided. In this growth process, several attempted branch formations are tested and potentially rejected. Therefore, a specific final accepted configuration could, in principle,

be obtained from the old configuration via an infinite number of ways (construction paths). As a result, the Metropolis-Hastings[32] rule for deriving acceptance probabilities becomes impractical as it requires the knowledge on the generation probabilities of all these branches, accepted and rejected, that need to be summed up. This issue is overcome in CBMC using the superdetailed balance principle, which can be formulated in terms of a construction path χ and its inverse $\overline{\chi}$.[17] That is, we not only require detailed balance between any possible old state and new state, but we require this for any specific route that connects these two states:

$$P_{\rm acc} = \min\left[1, \frac{P(\text{path}^{(n)})P_{\rm gen}(\text{path}^{(n)} \to \text{path}^{(o)} \text{ via } \overline{\chi})}{P(\text{path}^{(o)})P_{\rm gen}(\text{path}^{(o)} \to \text{path}^{(n)} \text{ via } \chi)}\right] \quad (1)$$

where $P_{\rm gen}(\text{path}^{(o)} \to \text{path}^{(n)} \text{ via } \chi)$ is the generation probability to generate the new state (path in our case) from the old state via construction path χ and $P_{\rm gen}(\text{path}^{(n)} \to \text{path}^{(o)} \text{ via } \overline{\chi})$ is the generation probability to generate the old state from the new state via the reverse construction path $\overline{\chi}$.

In subtrajectory moves, the "construction" path does not only describe the MD extensions of the final path, but also the sequence of subtrajectories including the failed ones. For SS and WT, the unsuccessful velocity generations, that do not obey the one-step crossing condition, should also be considered as part of the construction path χ. In other words, χ consists of several steps and the generation probability "via χ" is given by the product of generation probabilities of each step.

For each construction path χ there should exist an unique reverse construction path $\overline{\chi}$. Roughly said, when χ represents a sequence of algorithmic steps, $\overline{\chi}$ will typically consist of the reverse steps in reverse order. However, some groups of consecutive steps might actually happen in the same order. In fact, there is no unique way to define "a reverse", but for a given definition there will be a one-to-one relation between any possible χ and its reverse $\overline{\chi}$ and with that, valid acceptance/rejection rules can be derived based on the superdetailed balance, Eq. 1.

Yet, the definition of the reverse should be chosen such that the acceptance probability is computable and not negligibly small in the majority of cases. Therefore, the mathematical definition for the inverse is taken such that the probabilities of most of the algorithmic steps in the expressions for $P_{\rm gen}(\text{path}^{(o)} \to \text{path}^{(n)} \text{ via } \chi)$ and $P_{\rm gen}(\text{path}^{(n)} \to \text{path}^{(o)} \text{ via } \overline{\chi})$ will cancel.

For instance, if we represent the construction path as a vector containing the different steps in chronological order, χ could look like

$$\chi = [s^0, t^1, t^2, s^3, s^4, t^5, s^6] \quad (2)$$

which shows that there were 6 subtrajectories generated of which there were 3 failed trials t^1, t^2 and t^5. The initial step involves cutting out the very first subtrajectory s^0 from the old path, while the final step implies not only the generation of the last subtrajectory s^6 but also its extension to a full trajectory. The reverse construction path in this case is conveniently defined as

$$\overline{\chi} = [s^6, s^4, t^5, s^3, s^0, t^1, t^2] \quad (3)$$

So the order of the steps is not completely reversed, but the reverse order takes place on groups of consecutive steps, a group being a successful subtrajectory with all its failed trials that follow. The reason for this inverse is that Eq. 2 shows that trial trajectory t^5 can be generated starting from s^4, but this is not necessarily the case from s^6. Reversely, as s^6 was generated from s^4, they share a common configuration point, which makes it possible to generate s^4 from s^6. There is, however, no reason whatsoever that s^6 and t^5 share a common configuration point. Hence, if we would consider the reverse to be $\overline{\chi} = [s^6, t^5, s^4, \ldots]$, $P_{\rm gen}(\text{path}^{(n)} \to \text{path}^{(o)} \text{ via } \overline{\chi})$ would most likely be zero as $\overline{\chi}$ itself cannot be generated. In contrast, the inverse based on the grouped reordering, Eq. 3, contains generation probabilities like the probability to generate t^5 given s^4 which appear both in χ and $\overline{\chi}$. Therefore, all the generation probabilities of failed trajectories cancel in Eq. 1. Likewise, all failed velocity generations in SS and WT that do not obey the one-step crossing condition cancel out for the same reason as shown in Ref. 17.

Excluding all the failed steps that will cancel in Eq. 1, we can write for $P_{\rm gen}(\text{path}^{(o)} \to \text{path}^{(n)} \text{ via } \chi)$:

$$\begin{aligned} P_{\rm gen}(\text{path}^{(o)} \to \text{path}^{(n)} \text{ via } \chi) \propto\ & P_{\rm sel}(s^0|\text{path}^{(o)})\times \\ & P_{\rm sel}(r^{0,3}|s^0)P_{\rm gen}(v^{0,3})P_{\rm MD}(s^3|x^{0,3})\times \\ & P_{\rm sel}(r^{3,4}|s^3)P_{\rm gen}(v^{3,4})P_{\rm MD}(s^4|x^{3,4})\times \\ & P_{\rm sel}(r^{4,6}|s^4)P_{\rm gen}(v^{4,6})P_{\rm MD}(s^6|x^{4,6})\times \\ & P_{\rm sel}(\text{td})P_{\rm MD}(\text{path}^{(n)}|s^6) \end{aligned} \quad (4)$$

Here, $P_{\rm sel}(s^0|\text{path}^{(o)})$ is the probability for selecting s^0 from the old $\text{path}^{(o)}$ and $P_{\rm sel}(r^{0,3}|s^0)$ is the selection probability of choosing point $r^{0,3}$ from the subpath s^0 as the shooting point. Since $r^{0,3}$ is a shooting point to go from s^0 to s^3, it is a configuration point that s^0 and s^3 have in common. $P_{\rm gen}(v^{0,3})$ is the probability for generating the velocities $v^{0,3}$ which are the velocities of s^3 at the corresponding configuration point $r^{0,3}$. $P_{\rm MD}(s^3|x^{0,3})$ is the chance that starting from phase point $x^{0,3} = (r^{0,3}, v^{0,3})$, the MD integrator produces subpath s^3 by integrating the equations of motion forward and backward in time. The MD integrator can be based on actual Newtonian MD, Langevin, Brownian, etc. Likewise, $P_{\rm MD}(\text{path}^{(n)}|s^6)$ is the chance that the new $\text{path}^{(n)}$ is produced by extending the final subpath s^6. Finally, $P_{\rm sel}(\text{td})$ is the selection probability for the time-direction along the new path. Note that the time-direction along the subpaths is irrelevant in WT and WF. In SS, subpaths do have a sort of direction as the next shooting always takes place at the last λ_i crossing.[17]

For the reverse construction path, Eq. 3, we can write

$$\begin{aligned} P_{\rm gen}(\text{path}^{(n)} \to \text{path}^{(o)} \text{ via } \overline{\chi}) \propto P_{\rm sel}(s^6|\text{path}^{(n)})\times \\ P_{\rm sel}(r^{6,4}|s^6)P_{\rm gen}(v^{6,4})P_{\rm MD}(s^4|x^{6,4})\times \\ P_{\rm sel}(r^{4,3}|s^4)P_{\rm gen}(v^{4,3})P_{\rm MD}(s^3|x^{4,3})\times \\ P_{\rm sel}(r^{3,0}|s^3)P_{\rm gen}(v^{3,0})P_{\rm MD}(s^0|x^{3,0})\times \\ P_{\rm sel}(\text{td})P_{\rm MD}(\text{path}^{(o)}|s^0) \end{aligned} \quad (5)$$

Now, it becomes apparent that most terms will cancel out in Eq. 1 when we take the ratio between Eq. 5 and Eq. 4. First of all, the time-direction is chosen with a 50% probability such that $P_{\rm sel}(\text{td}) = 0.5$. Then, we can use the fact that a path probability can be written in terms of a phase point probability times the MD generation probability

$$P(\text{path}) = \rho(x)P_{\rm MD}(\text{path}|x) \quad (6)$$

where $\rho(x)$ is the phase space equilibrium density for any phase point x that is part of the path.[17] For a phase point $x = (r, v)$ this can be split into

$$\rho(x) = \rho_r(r)\rho_v(v) \quad (7)$$

where ρ_r and ρ_v are, respectively, the configuration (Boltzmann) distribution and the velocity Maxwell-Boltzmann distribution (possibly subjected to bond- and angle constraints if applicable). Further, as generating new velocities in Eqs. 4 and 5 is based on the velocity distribution, $P_{\rm gen}(v) = \rho_v(v)$, we can substitute all $P_{\rm MD}$ terms in Eqs. 4 and 5, e. g.:

$$\begin{aligned} P_{\rm gen}(v^{4,6})P_{\rm MD}(s^6|x^{4,6}) &= \frac{P_{\rm gen}(v^{4,6})P(s^6)}{\rho(x^{4,6})} \\ &= \frac{\rho_v(v^{4,6})P(s^6)}{\rho(x^{4,6})} = \frac{P(s^6)}{\rho_r(r^{4,6})} \\ P(s^6)P_{\rm MD}(\text{path}^{(n)}|s^6) &= P(\text{path}^{(n)}) \end{aligned} \quad (8)$$

Applying these operations to Eqs. 4 and 5, we get

$$\begin{aligned} P_{\rm gen}(\text{path}^{(o)} \to \text{path}^{(n)} \text{ via } \chi) \propto P_{\rm sel}(s^0|\text{path}^{(o)})P_{\rm sel}(\text{td})\times \\ P_{\rm sel}(r^{0,3}|s^0)P_{\rm sel}(r^{3,4}|s^3)P_{\rm sel}(r^{4,6}|s^4)\times \\ P(s^3)P(s^4)P(\text{path}^{(n)})/[\rho_r(r^{0,3})\rho_r(r^{3,4})\rho_r(r^{4,6})] \\ P_{\rm gen}(\text{path}^{(n)} \to \text{path}^{(o)} \text{ via } \overline{\chi}) \propto P_{\rm sel}(s^6|\text{path}^{(n)})P_{\rm sel}(\text{td})\times \\ P_{\rm sel}(r^{6,4}|s^6)P_{\rm sel}(r^{4,3}|s^4)P_{\rm sel}(r^{3,0}|s^3)\times \\ P(s^4)P(s^3)P(\text{path}^{(o)})/[\rho_r(r^{6,4})\rho_r(r^{4,3})\rho_r(r^{3,0})] \end{aligned} \quad (9)$$

In the ratio of these two equations more terms will cancel out as $r^{\alpha,\beta} = r^{\beta,\alpha}$. Further, $P_{\rm sel}(\text{td}) = 0.5$ as stated before. In all subtrajectory moves, $P_{\rm sel}(r|s^\alpha)$ is either a fixed number (SS and WT) or it depends on s^α, but not on r (WF). In SS, the shooting point is selected from the last crossing with λ_i and therefore $P_{\rm sel}(r|s^\alpha) = 2$ (the phase point just before or after λ_i). In WT, it is randomly chosen from a crossing with either λ_i or $\lambda_{\rm sour}$ and therefore $P_{\rm sel}(r|s^\alpha) = 4$. With stochastic dynamics one can also opt to choose only the inner points[17] such that $P_{\rm sel}(r|s^\alpha) = 2$. In WF any point of the subpath that lies between λ_i and λ_B (or $\lambda_{\rm cap}$) can be chosen. In all these cases the $P_{\rm sel}(r|s^\alpha)$ terms with identical s^α cancel out in the ratio. That means that the only terms that remain depend on the first and last subpath (s^0 and s^6), or on the full paths ($\text{path}^{(o)}$ and $\text{path}^{(n)}$):

$$\begin{aligned} \frac{P_{\rm gen}(\text{path}^{(n)} \to \text{path}^{(o)} \text{ via } \overline{\chi})}{P_{\rm gen}(\text{path}^{(o)} \to \text{path}^{(n)} \text{ via } \chi)} = \\ \frac{P_{\rm sel}(s^6|\text{path}^{(n)})P_{\rm sel}(r^{6,4}|s^6)P(\text{path}^{(o)})}{P_{\rm sel}(s^0|\text{path}^{(o)})P_{\rm sel}(r^{0,3}|s^0)P(\text{path}^{(n)})} = \\ \frac{P_{\rm sel}(r^{6,4}|\text{path}^{(n)})P(\text{path}^{(o)})}{P_{\rm sel}(r^{0,3}|\text{path}^{(o)})P(\text{path}^{(n)})} = \frac{P(\text{path}^{(o)})/M^{(n)}}{P(\text{path}^{(n)})/M^{(o)}} \end{aligned} \quad (10)$$

where in the third expression we contracted the selection probabilities involving the two-steps (first selecting s^0 or s^6, then selecting $r^{0,3}$ or $r^{6,4}$) to the chance of selecting the very first successful crossing point from the existing full path. Finally, the latter was replaced by $1/M^{(n)}$ and $1/M^{(o)}$ where $M^{(n)}$ and $M^{(o)}$ are the numbers of different equally probable possibilities to select a shooting point for generating a subtrajectory from the new and old full path, respectively.

If we substitute Eq. 10 into Eq. 1, we obtain a rather simple expression for the acceptance:

$$P_{\rm acc} = \min\left[1, \frac{M^{(o)}}{M^{(n)}}\right] \quad (11)$$

In SS, $M^{(o)}$ and $M^{(n)}$ are simply proportional to the number of crossing points of the old and new paths with λ_i, while for WT these are proportional to the number segments that can be cut out of these trajectories that connect $\lambda_{\rm sour}$ and λ_i.[17] In WF, these relate to the number of points between λ_i and λ_B. If a so-called *cap*-interface is defined, $M^{(o)}$ and $M^{(n)}$ relate to the number of points between λ_i and $\lambda_{\rm cap}$ excluding any points lying on a segment $\lambda_{\rm cap} \to \lambda_{\rm cap}$ without crossing λ_i.

Eq. 11 can also be combined with an early rejection scheme as was introduced in Ref. 5. In the standard approach, one would complete the MC move, compute the acceptance probability, Eq. 11, take a uniform random number α between 0 and 1, and then accept if $\alpha < P_{\rm acc}$ and reject otherwise. In the early rejection scheme, the random number α is taken first and the move is rejected as soon as $M^{(n)} > M^{(o)}/\alpha$. In normal shooting, this provides a considerable speed up since long paths have a high chance to get rejected. Using the early rejection scheme a lot of unnecessary MD steps can be avoided as these paths can be stopped whenever they exceed the predetermined maximum length. Yet, for the subtrajectory moves the *high-acceptance* scheme is preferable as we discuss in Sec V. In the next section we show why the

subtrajectory moves allow us to sample fewer trajectories than with standard shooting via a reduction of the statistical inefficiency.

IV. STATISTICAL INEFFICIENCY

The principal property that is computed in the $[i^+]$ ensemble is the local crossing probability $P_A(\lambda_{i+1}|\lambda_i)$. This is the history dependent conditional probability that the system, given it crosses λ_A and then crosses λ_i, crosses λ_{i+1} before λ_A. In the *post hoc* analysis, this local crossing probability is simply the fraction of sampled path in the $[i^+]$ ensemble that happen to cross λ_{i+1} in addition to λ_i. Once these are accurately enough determined, the global crossing probability $P_A(\lambda_B|\lambda_A)$ is obtained from:[5,8]

$$P_A(\lambda_B|\lambda_A) = \prod_{i=0}^{n-1} P_A(\lambda_{i+1}|\lambda_i) \tag{12}$$

where $\lambda_0 = \lambda_A$ and $\lambda_n = \lambda_B$. The above expression is exact since the local crossing probabilities include the full history dependence ($\lambda_A \to \lambda_i$) in their condition.[33] An alternative approximate expression for the global crossing probability is used in partial path TIS[13] in which the amount of spatial memory is reduced though not set to zero, as in milestoning.[14] The global crossing probability gives the rate of the transition when multiplied with f_A, the conditional flux through λ_A.

In TIS, the flux is calculated by straightforward MD where the system is prepared in state A and then the number of crossings with λ_A per time unit is computed. If a spontaneous transition to state B takes place, which is unlikely for a rare event, the simulation is paused, reinitiated in state A and then continued. RETIS computes the flux term differently as it does not use a single continuous MD simulation. Instead, the $[0^-]$ path ensemble is introduced to explore the A state, and the flux is derived from the average path lengths in $[0^-]$ and $[0^+]$.[6] In addition to rate constants, the overall crossing probability can also be used to compute permeability coefficients[34] and activation energies.[35,36]

Considering the j-th path in the simulation for path ensemble $[i^+]$, the main output of sample j (the generated path) that is relevant for the computation of the crossing probability is simply the observation of whether it crosses λ_{i+1} or not. We can describe this by a characteristic function h_j which equals 1 if λ_{i+1} is crossed and 0 otherwise. The simulation estimate of the local crossing probability, $p(m)$, after m MC moves is then expressed as

$$p(m) = \frac{1}{m}\sum_{j=0}^{m-1} h_j \approx P_A(\lambda_{i+1}|\lambda_i) \tag{13}$$

where the index counter starts from zero for mathematical convenience.

For finite m, the value of $p(m)$ will not be exact and the absolute error, ϵ_a, is defined as the standard deviation of the mean $\sigma_{p(m)}$. This is essentially the standard deviation in possible $p(m)$ results if the simulation experiment would be carried out multiple times. Mathematically we can write this as

$$\epsilon_a = \sigma_{p(m)} = \sqrt{\langle (p(m)-p)^2 \rangle} \tag{14}$$

where $p = p(\infty) = P_A(\lambda_{i+1}|\lambda_i)$ and the brackets $\langle\cdot\rangle$ refer to the perfect ensemble sampling average. This can be viewed as the hypothetical average that is obtained after repeating the simulation an infinite number of times starting with initial conditions that are randomly drawn form a perfect statistical equilibrium distribution. In other words, we have $\langle p(1)\rangle = \langle p(m)\rangle = p$. Further, since detailed balance MC moves conserve the equilibrium distribution,[29] the absolute value of the index j is irrelevant and $\langle h_0\rangle = \langle h_1\rangle = \langle h_j\rangle = p$ and $\langle h_j h_k\rangle = \langle h_0 h_{k-j}\rangle$ for any j, k. Using this, one can show that:[37]

$$\sigma^2_{p(m)} = \frac{\sigma^2_{p(1)}}{m}\mathcal{N}, \quad \mathcal{N} = [1 + 2n_c] \tag{15}$$

where $\mathcal{N}$ is called the statistical inefficiency and n_c is the correlation number which is the integral of the correlation function $C(j)$:

$$n_c = \sum_{j=1}^{\infty} C(j), \quad C(j) = \frac{\langle (h_0 - p)(h_j - p)\rangle}{\langle (h_0 - p)^2\rangle} \tag{16}$$

As the output h_j of a single sample is either 1 with a probability p or 0 with a probability $(1-p)$, the sample standard deviation $\sigma_{p(1)}$ can be simplified

$$\begin{aligned}\sigma^2_{p(1)} &= \big\langle (p(1)-p)^2 \big\rangle = \big\langle (h_0 - p)^2 \big\rangle \\ &= p(1-p)^2 + (1-p)(0-p)^2 = p(1-p)\end{aligned} \tag{17}$$

Via Eqs. 15, 16, and 17, we can write for the relative error:

$$\epsilon_r = \frac{\epsilon_a}{p} = \sqrt{\frac{1-p}{p}\frac{\mathcal{N}}{m}} \tag{18}$$

Eq. 18 shows that for a fixed number of MC moves m, the larger the local crossing probability $p = P_A(\lambda_{i+1}|\lambda_i)$, the lower the relative error. Hence, the result in simulation $[i^+]$ converges faster when the difference between λ_i and λ_{i+1} is small, but this will obviously increase the number of path ensembles needed. Analytical results on model systems suggest that the optimum placement of interfaces in TIS is achieved when $p \approx 0.2$ for all ensembles.[37] In RETIS the optimum is expected to be slightly higher as this would lead to more successful swaps. Likewise, the optimum is also slightly higher if the weighted histogram analysis method (WHAM)[38] is used instead of single-point matching to determine the total crossing probability. In this approach, the crossing statistics of path ensemble $[i^+]$ is not limited to the fraction of paths crossing λ_{i+1}, but also the fractions for crossing λ_{i+2}, λ_{i+3}, etc. are used to get a slightly more accurate estimate of Eq. 12.[39,40]

If the sampling between successive MC moves is completely uncorrelated, we have that $\langle (h_0 - p)(h_j - p)\rangle = \langle (h_0 - p)\rangle \cdot \langle (h_j - p)\rangle = 0 \cdot 0 = 0$. This would imply that $C(j) = n_c = 0$ and $\mathcal{N} = 1$. In this case, if $p = 0.2$, there are about $m = 400$ trajectories required to obtain an $\epsilon_r = 10\%$ error. For $\mathcal{N} > 1$, one would need $m/\mathcal{N} = m_u = 400$ to get the same error. Here, m_u is called the number of effectively uncorrelated samples.

In general, $C(j) \neq 0$ except for the limit $j \to \infty$ as correlation decays. If a MC move is rejected at step j, then the previous sample is kept and recounted such that sample j is identical to sample $j-1$. Hence, if there are j consecutive rejections, sample j is identical to sample 0 causing correlation over multiple steps. Even if the j-th step is accepted, it tends to have some similarity with the previous sample. Therefore, there is a high probability for $h_j = h_{j-1}$, even if the samples are not identical. The correlations lead to a sampling output $(h_0, h_1, h_2, \ldots)$ with long rows of consecutive zeros and consecutive ones.

To illustrate this effect with a mathematical example: suppose that the MC move has a probability π_R to remain unchanged such that $h_j = h_{j-1}$ and a probability $\pi_M = 1 - \pi_R$ to actually make a move that potentially (but not necessarily) changes the output: the new sample yields $h_j = 1$ with a probability p and $h_j = 0$ with a probability $(1-p)$. As shown in the appendix, for this mathematical model the statistical inefficiency equals:

$$\mathcal{N} = \frac{2 - \pi_M}{\pi_M} \tag{19}$$

This shows that for a typical MC acceptance probability of 50%, the effect of rejections alone causes the statistical inefficiency to be equal to 3. The situation is usually worse in complex systems and also more difficult to identify than merely by the presence of rows of consecutive ones or zeros. For instance, inter- and intramolecular changes of reactants could temporarily boost or reduce the probability of a transition. The same kind of fluctuations in the temporary transition probability can be caused by the local solvent structure and the position and orientation of catalytic molecules. These describe degrees of freedom that are orthogonal to the reaction coordinate.

We can examine this by a slightly more complex model where we assume that there are two phases α and β, described by the orthogonal degrees of freedom, which occur with probabilities P_α and $P_\beta = 1 - P_\alpha$. Let p_α and p_β be the corresponding local crossing probabilities along the reaction coordinate for these phases such that: $p = P_\alpha p_\alpha + P_\beta p_\beta$. Analogous to the above, let π_ρ be the chance to not update the phase, and $\pi_\mu = 1 - \pi_\rho$ be the chance to freshly choose between phase α or β with respective probabilities P_α and P_β. As shown in the appendix, in this case the statistical inefficiency equals:

$$\mathcal{N} = \frac{2K_s - \pi_\mu(2K_s - 1)}{\pi_\mu} \tag{20}$$

where K_s is a system parameter that does not depend on the type of MC move:

$$K_s = \frac{P_\alpha P_\beta (p_\alpha - p_\beta)^2}{p(1-p)} = \frac{(p - p_\alpha)(p_\beta - p)}{p(1-p)} \tag{21}$$

Note that $K_s = 0$ whenever $p_\alpha = p_\beta$, which gives $\mathcal{N} = 1$. This would be the case if all TIS interfaces are placed at isocommittor surfaces, which partly supports the hypothesis of Ref. 41 that stated that path sampling simulations are most efficient if the reaction coordinate λ equals the committor. However, although this surely minimizes the statistical inefficiencies, the mean path lengths in the path ensembles also depend on the choice of the reaction coordinate λ. If this is included in the analysis, the hypothesis is at least not generally true.[33]

Now assume that not all generated paths are saved and analyzed, but instead only every N_s-th path is kept. While this will cause a reduction in the number of samples from m to m/N_s, it does not necessarily reduce the number of uncorrelated samples m_u as the statistical inefficiency between saved samples is also reduced. In particular, the "remain" probability between saved samples changes from π_ρ to $\pi_\rho^{N_s}$ and, therefore, the "move" probability changes from π_μ to $1 - \pi_\rho^{N_s} = 1 - (1 - \pi_\mu)^{N_s}$. The statistical inefficiency between saved samples is henceforth:

$$\mathcal{N}(N_s) = \frac{2K_s - (1 - (1 - \pi_\mu)^{N_s})(2K_s - 1)}{1 - (1 - \pi_\mu)^{N_s}} \tag{22}$$

Eq. 22 shows that the statistical efficiency indeed goes down with increasing N_s up to an asymptote equal to 1. Taking the power series up to first order in π_μ, we see that the initial downfall is inversely linear:

$$\mathcal{N}(N_s) \approx \frac{2K_s - N_s\pi_\mu(2K_s - 1)}{N_s\pi_\mu} \approx \frac{\mathcal{N}(1)}{N_s} \tag{23}$$

where we assumed $N_s\pi_\mu \ll 1$. As a result, saving every N_s-th path instead of all paths will not affect much the post-simulation analysis in terms of accuracy. The reduction in the number of data points from m to m/N_s is compensated by a lower statistical inefficiency such that the number of uncorrelated samples m_u remains nearly unchanged. While this allows for obvious data storage savings, reducing both the memory and time for writing to disk, it also paves the way to reduce MD steps as is shown in Fig. 2. The figure illustrates a hypothetical MC sequence in path sampling of six consecutive paths, labeled 0 to 5, where the shooting point has an order parameter larger than λ_i. If only every fifth path is saved, only paths 0 and 5 are considered as in Fig. 2-a). Although the intermediate paths contribute for their decorrelation, it is clear that many MD steps can be omitted, as exploited by the subtrajectory moves. Fig. 2-b) shows a scenario where the same final path is being generated with a set of hypothetical WF subtrajectories resembling the top scenario. Instead of five full trajectories, only four short subtrajectories and one full trajectory are needed to establish a new full path (path 5) from the old one (path 0).

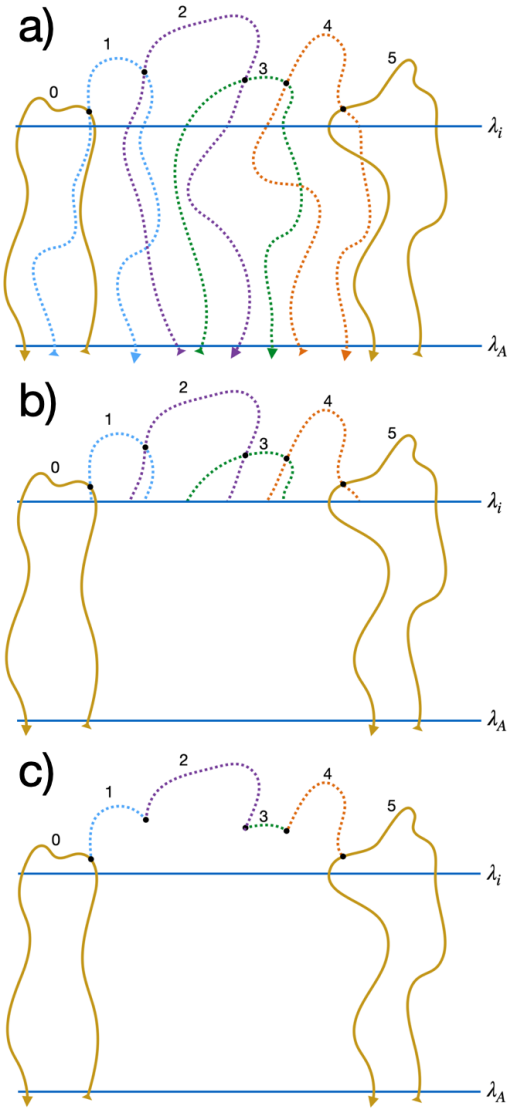


FIG. 2. Illustration of wasted MD steps in shooting and WF. Panel a) shows six consecutive paths being generated by the shooting move where only the solid golden paths, with index 0 and 5, are being saved. Panel b) gives an equivalent scenario in the WF algorithm showing that considerably fewer MD steps are needed to obtain the same paths 0 and 5 via $N_s = 5$ subtrajectories. Still, WF is not as thrifty as SS and WT since only parts of the subtrajectories, shown in panel c), actually contribute to the sampling progress to get from path 0 to 5. The additional steps in panel b) are seemingly "wasted" but still needed for the superdetailed balance relation. SS and WT do not generate wasted MD steps, but rely on a one-step crossing condition as discussed in the main text.

Based on this principle alone, the relative efficiency gain η of subtrajectory moves compared to standard shooting is expected to be

$$\eta(N_s) = \frac{N_s L_p}{L_p + (N_s - 1)L_s} \tag{24}$$

where L_p and L_s are, respectively, the average length of a full path and a subpath. Still, if we purely focus on the MD steps that are required to allow for the progression from path 0 to path 5, even fewer MD steps are needed as shown in Fig. 2-c). Yet, the "extra" (wasted) MD steps in panel b) are required for the superdetailed balance as discussed in Sec. III. Wasted MD steps are avoided in SS and WT where the shooting always happens at an interface (see Figs. 1-a) and b)). The price to be paid for this is the additional complication with regard to the one-step crossing condition (see Sec. VI). But even with a slightly higher MD waste, the WF move requires considerably fewer MD steps than standard shooting.

Eq. 24 levels off to a constant L_p/L_s for increasing N_s. Likewise, Eqs. 23 and 22 show that the trend $\mathcal{N}(N_s) = \mathcal{N}(1)/N_s$ is not sustained for increasing N_s as $\mathcal{N}$ ultimately levels off to 1. It is henceforth assumed that while efficiency initially increases quite rapidly as function of N_s, it can not surpass L_p/L_s and ultimately even decreases when $\mathcal{N}(N_s)$ levels off. Clearly, for the $[0^-]$ and $[0^+]$ ensemble where $L_p = L_s$ no gain is expected and one could set $N_s = 1$ if data storage latency would not be an issue. Therefore, as a rule of thumb, N_s can be set approximately equal to L_p/L_s such that for $L_p > L_s$ the cost of the MC move is less than doubled, while Eq. 24 reaches more than 50% of its anyways unattainable maximum of L_p/L_s.

Although the essence of the above analysis is correct, there is however a caveat: rejections leave a much heavier mark on the subtrajectory move than on standard shooting. If, for instance, the extension of the fifth and last subpath in Fig. 2-b) is rejected, it would imply a complete reset to the latest accepted full path (path 0) since subpath 4 is not a valid trajectory and extending subpath 4 after the rejection would violate detailed balance. As a result, all MD steps of subpaths 1 to 5 are trashed as the next move starts from path 0 again. Instead, the MC chain will only fall back to path 4 (assuming path 4 was accepted) in standard shooting. It is therefore clear that rejections in the subtrajectory move approach should be avoided even more than in the shooting method. This can be achieved with the high-acceptance procedure that is discussed in the next section.

V. HIGH-ACCEPTANCE PROCEDURE

As discussed in the previous section, a rejection in the subtrajectory moves implies a large amount of wasted MD steps. An early rejection scheme, like the one used in TIS and RETIS with standard shooting (see Sec. III), is also not so helpful as a rejection cannot be made until the generation of the last subtrajectory has been initiated. It is, therefore, preferable to combine the subtrajectory

moves with the *high-acceptance* scheme.[17] The approach uses the following two tricks. First, if the final subtrajectory has a backward extension ending in state B, the MC move is not directly rejected. Instead, the extension forward in time is completed and, if it ends in state A, the path is time-reversed providing an $A \to B$ path. The consequence is that the time-direction selection probability $P_{\text{sel}}(\text{td})$ in Eq. 4 is no longer 0.5 for all paths as an $A \to B$ path can be generated in two ways: either by choosing the correct time-direction immediately, or in reverse. This implies an extra factor two in the generation probabilities P_{gen}, in Eqs. 1 and 10, of the $A \to B$ paths compared to $A \to A$ paths. We henceforth write:

$$\frac{P_{\text{gen}}(\text{path}^{(n)} \to \text{path}^{(o)} \text{ via } \overline{\chi})}{P_{\text{gen}}(\text{path}^{(o)} \to \text{path}^{(n)} \text{ via } \chi)} = \frac{P(\text{path}^{(o)})q(\text{path}^{(o)})/M^{(n)}}{P(\text{path}^{(n)})q(\text{path}^{(n)})/M^{(o)}} \tag{25}$$

where

$$q(\text{path}) = \begin{cases} 1 & \text{if path} \in \{A \to A\} \\ 2 & \text{if path} \in \{A \to B\} \end{cases} \tag{26}$$

The second trick is to slightly change the sampling distribution. Instead of sampling the correct physical path distribution, $P(\text{path})$, restrained to the path ensemble's requirements, an alternative path distribution $\tilde{P}(\text{path})$ is sampled. From Eqs. 1 and 25, the acceptance probability thus becomes

$$P_{\text{acc}} = \min\left[1, \frac{\tilde{P}(\text{path}^{(n)})P(\text{path}^{(o)})q(\text{path}^{(o)})M^{(o)}}{\tilde{P}(\text{path}^{(o)})P(\text{path}^{(n)})q(\text{path}^{(n)})M^{(n)}}\right] \tag{27}$$

and to maximize the acceptance, we choose the sampling distribution in ensemble $[i^+]$ as

$$\begin{aligned} \tilde{P}(\text{path}) &= P(\text{path})w_i(\text{path})\mathbf{1}_{[i^+]}(\text{path}) \text{ with} \\ w_i(\text{path}) &= q(\text{path})M_{\lambda_i}(\text{path}) \end{aligned} \tag{28}$$

where $\mathbf{1}_C(x)$ is the indicator function that equals 1 if x is part of set C and 0 otherwise. A subscript λ_i was added to the last term M, as the number of equal probable possibilities for a first shooting, generally depends on the interface λ_i. Substituting Eq. 28 in Eq. 27 implies that with high-acceptance

$$P_{\text{acc}} = \mathbf{1}_{[i^+]}(\text{path}^{(n)}) \tag{29}$$

In other words, the new path will always be accepted unless the MC move led to a path not obeying the ensemble's definition: starting at λ_A, ending at λ_A or λ_B, and having at least one crossing with λ_i. By construction, the crossing of λ_i is always achieved in the subtrajectory moves if the starting condition at λ_A is met. Hence, the only necessary rejection is when the extension of the final successful subtrajectory ends at λ_B in both time-directions.

If no successful subtrajectories were generated after N_s attempts, s^0 could be extended. However, since this would regenerate the old trajectory in deterministic dynamics and otherwise a trajectory that is highly correlated with the old one, it is preferable to reject the move. Other potential reasons for rejections could be due to non-convergence of the atomistic forces in AIMD level calculations. Another potential issue is jumpy order parameters,[42] such that M_{λ_i} can be zero even if the path is actually valid. This issue is further discussed in Sec. VII.

Exact natural averages can still be obtained by weighting each sample j with the inverse of $w_i(j)$. For instance, the estimated local crossing probability, previously defined by Eq. 13, can now be expressed as

$$p(m) = \frac{\sum_{j=0}^{m-1} w_i(j)^{-1} h_j}{\sum_{j=0}^{m-1} w_i(j)^{-1}} \approx P_A(\lambda_{i+1}|\lambda_i) \tag{30}$$

The effect of the weighting implies that different samples have different contribution. If a sample j' has a much lower than average w_i^{-1} factor, the sample could essentially be removed from Eq. 30 without significantly affecting the estimate $p(m)$. Yet, thanks to this sample not being rejected, sample $j'+1$ is more different than $j'-1$ than it would be in the case that j' was rejected. This shows the power of the high-acceptance approach.

The improved acceptance in the subtrajectory move will slightly reduce the acceptance in the replica exchange move. For instance, if a path j from ensemble $[i^+]$ will be exchanged with a path k from ensemble $[(i+1)^+]$, the acceptance becomes:[17]

$$P_{\text{acc}} = \mathbf{1}_{[(i+1)^+]}(j) \times \min\left[1, \frac{w_i(k)w_{(i+1)}(j)}{w_i(j)w_{(i+1)}(k)}\right] \tag{31}$$

Without high-acceptance, the factor in Eq. 31 after the multiplication sign equals 1. This means that whenever j, the path originating from $[i^+]$ is valid for $[(i+1)^+]$, the swap will be accepted. Note that any path in $[(i+1)^+]$ is also valid in $[i^+]$. This lower acceptance is not dramatic since replica exchange moves do not require any MD steps. Therefore, replica exchange moves have negligible CPU cost. The only exception is the $[0^-] \leftrightarrow [0^+]$ swap in which two new paths are generated. Without high-acceptance, this move is always accepted. For SS and WT the acceptance remains 100%, but this is not the case for WF. We can solve this problem for WF in RETIS by sampling the $[0^-]$ and $[0^+]$ ensembles with the standard shooting method without high acceptance. Due to this w_{0^+} and w_{0^-} equal 1 irrespective to the paths and swapping between these two ensembles will always be accepted. The absence of high-acceptance is partly compensated by early rejection (see Sec. III). Moreover, in these ensembles there is no difference between the average path length of a subpath and a full path, making the subtrajectory moves anyways not so effective for these ensembles.

The high-acceptance protocol eliminates the more serious drawbacks of rejections in the subtrajectory moves compared to shooting. In the next section we discuss how the one-step crossing condition can be met.

VI. ONE-STEP CROSSING CONDITION

As discussed above, SS and WT are very thrifty algorithms with respect to the number of generated MD steps. Yet, the one-step crossing condition puts a challenge to the implementation. One can eliminate the one-step crossing condition via the new but less thrifty WF algorithm that is further discussed in Sec. VII. In this section we discuss a few algorithmic solutions to overcome the one-step crossing condition in SS and WT. These two approaches assume that one time step in (RE)TIS is effectively also one MD step.

The one-step crossing can be achieved in different ways. The most straightforward way is to generate velocities from a Maxwell-Boltzmann distribution, execute an MD step, calculate the new order parameter, and if the crossing is established, then the two frames comprising the crossing are extended at the side above λ_i to create a new subpath. The problem with this approach is that, after each velocity generation, an MD step, and therefore a force calculation is required. Especially if λ_i is at a steep slope of the potential energy surface, the two trajectory frames forming the crossing of a given interface might be rather far apart in λ-space. In such cases, if one of the two frames is located in the very proximity of the interface, it might be extremely unlikely to re-generate a new one-step crossing from the configuration furthest to the λ_i interface given a random approach to generate velocities.

There are essentially two strategies to reduce the cost for fulfilling the one-step criterion: i) generate atom velocities from a Maxwell-Boltzmann distribution and predict the next step's order parameter without performing an actual MD step, and ii) generate velocities in a way such that the crossing is likely achieved after very few attempts. Strategy i) assumes that generating new velocities is rather computationally inexpensive and the expense of the one-step crossing condition is mostly provided by the force calculation. This is the case for AIMD level simulations as these typically consist of just a few (hundreds of) atoms, while requiring a high CPU demand for the force calculation. In large classical MD systems with a significant number of atoms, the velocity generation might actually be equally expensive as a force calculation. In that case, strategy ii) might be preferable.

i) Prediction strategy

The velocity-Verlet[29] MD integrator propagates a phase point $x(t) = (r(t), v(t))$ deterministically to a next phase point $x(t+\Delta t) = (r(t+\Delta t), v(t+\Delta t))$. The integrator is most conveniently expressed via "intermediate velocities" at $t + \Delta t/2$:

$$\begin{aligned} v(t + \Delta t/2) &= v(t) + f(t)\Delta t/(2m) \\ r(t + \Delta t) &= r(t) + v(t + \Delta t/2)\Delta t \\ v(t + \Delta t) &= v(t + \Delta t/2) + f(t + \Delta t)\Delta t/(2m) \end{aligned} \quad (32)$$

where m is the mass and f are the forces. We used a simplified notation here, but one should realize that for an N particle system both r, v and f are 3N-dimensional vectors and m is actual a $3N \times 3N$ diagonal mass matrix. Further, the forces are determined from the positions: $f(t) = f(r(t))$.

Eq. 32 suggests that one MD step requires two force evaluations, but this is not the case when the steps of Eq. 32 are called repeatedly in a loop. After the force calculation at the third step, required to determine $v(t + \Delta t)$, the forces are stored such that these can be used at the first step of the next cycle. With the same reasoning, if the forces are known already at time t from its previous step, a new force evaluation is only needed to determine $v(t+\Delta t)$, but not $r(t+\Delta t)$. This means that if the order parameter only depends on geometry, $\lambda = \lambda(r)$, its value at $t + \Delta t$ can also be determined without the need of doing an actual force calculation.

When testing the one-step crossing for the selected configuration with randomized velocities, a new (single step) MD trajectory is started with no information available from the previous MD step. However, the selected configuration is also part of the previous subpath, so the corresponding forces could have been known, in principle. When not available, the forces can be reobtained from the trajectory data without further electronic structure calculation in AIMD or from the gradient of force field potential in classical MD. In particular, let $x_1 = (r_1, v_1)$ and $x_2 = (r_2, v_2)$ be two consecutive phase points of the latest subpath that define a crossing. This means that x_2 follows from x_1 through a single MD step and both points are at opposite sides of the interface. Therefore, both points are viable points for shooting off the next subpath. By inverting Eq. 32 we can derive

$$f_1 = \frac{2\,m\,(r_2 - r_1 - \Delta t\, v_1)}{\Delta t^2}, \quad f_2 = \frac{2\,m\,(r_1 - r_2 + \Delta t\, v_2)}{\Delta t^2} \quad (33)$$

So Eq. 33 directly provides the forces on the two potential shooting points by reading the trajectory data from the subpath. Given that one of these two points is selected as a shooting point and new randomized velocities are generated, the coordinates after one MD step can be determined without any additional force calculation but using just the first two steps of Eq. 32. Hence, the value of the order parameter after one step can be asserted.

If the prediction suggests that a crossing might be achieved, the MD step is completed and then the next subtrajectory is generated. If the velocities do not lead to a crossing, a new velocity randomization is attempted until the crossing condition is met. As in SS, the shooting point selection has to be maintained, the computation of Eq 33 only needs to be done once for the generation of each subpath. Naturally, if the MD step integrator is more complex than velocity-Verlet (due to thermostats, barostats, constraints, stochasticity), then the prediction becomes more difficult. The method also works best if a MD step is computationally expensive while regenerating

velocities is relatively cheap. This method is therefore more suitable for simulations with AIMD level. The approach has been implemented in the PyRETIS software, and it can be directly used with the CP2K[25] external MD engine. Note that the use of the plain velocity-Verlet MD integrator is rather common in path sampling since the generation of paths is already thermostated via the shooting move that allows a change of energy, while the individual paths have NVE dynamics.

ii) Alternative velocity generation

The mathematically simple form of Eq. 10 is due to the many terms conveniently canceling out. For instance, the terms in Eqs. 9, $\rho_r(r^{0,3})$, $\rho_r(r^{4,3})$, and $\rho_r(r^{4,6})$ in $P_{\text{gen}}(\text{path}^{(o)} \rightarrow \text{path}^{(n)} \text{ via } \chi)$ cancel out with, respectively, $\rho_r(r^{3,0})$, $\rho_r(r^{4,3})$, and $\rho_r(r^{6,4})$ in $P_{\text{gen}}(\text{path}^{(n)} \rightarrow \text{path}^{(o)} \text{ via } \overline{\chi})$ because $r^{\alpha,\beta} = r^{\beta,\alpha}$. However, whereas consecutive (accepted) subtrajectories share a common configuration point, they do not necessarily share of common phase point as $v^{\alpha,\beta} \neq v^{\beta,\alpha}$. Here, $v^{\alpha,\beta}$ refers to the velocities of s^β at the configuration point $r^{\alpha,\beta}$, and $v^{\beta,\alpha}$ refers to the velocities of s^α at an identical configuration point. These velocities have typically not the same orientation nor amplitude. Luckily, the $\rho_v(v^{\alpha,\beta})$ terms still cancel out via $P_{\text{gen}}(v^{\alpha,\beta}) = \rho_v(v^{\alpha,\beta})$ and $\rho(x^{\alpha,\beta}) = \rho_r(r^{\alpha,\beta})\rho_v(v^{\alpha,\beta})$ in Eq. 8.

Now, suppose that in a N particle system not all $3N$ velocity components are regenerated from a Maxwell-Boltzmann distribution, but some velocities components are kept and some others are inverted (multiplied with -1). These two velocity groups do not cancel out in Eq. 8 as they are not part of P_{gen} which implies that the final results changes from $P(s^6)/\rho_r(r^{4,6})$ to $P(s^6)/[\rho_r(r^{4,6})\rho_v(u^{4,6})]$ where $u^{4,6}$ are the velocity components that are either unchanged or inverted. Since the equilibrium velocity distribution is symmetric $\rho_v(v) = \rho_v(-v)$, and $u^{4,6}$ is identical to $u^{6,4}$ except for some components having different sign, all the $\rho_v(u^{\alpha,\beta})$ terms cancel in the ratio, Eq. 10, just like the $\rho_r(r^{\alpha,\beta})$ terms.

This allows for different strategies. For instance, if the dynamics is stochastic, all velocities can simply be inverted. This option was used for WT in Ref. 17. Inverting the velocities of specific atoms or molecules whose coordinates determine the order parameter could also be effective. The other velocities could be either kept unchanged, randomized, or a combination. For instance, in protein folding simulations inverting the velocities of all protein atoms while leaving the velocities of the solvent molecules (partly) unchanged would make the sampling less diffusive. Reinspection of Eq. 32 shows that the coordinates of the atoms with the inverted velocities are mapped exactly back after 1 MD step to the previous coordinates regardless of the velocities of the other atoms. As a result, the one-step crossing condition is automatically fulfilled.

This approach requires, however, a single MD step resolution at the interface crossing. In large molecular systems, it is not desirable to save trajectory coordinates every MD step as it could overwhelm hard disk capacity and will result in a loss of effective CPU efficiency due to the time that is spent writing to disk. An adaptive scheme could be adopted when the frequency of order parameter determination and the data retention is intensified whenever the system approaches an interface. Since trajectories can later be swapped in a replica exchange move, this adaptive approach would have to be carried out for all interfaces or, at least, in the proximity of neighboring interfaces. The latter choice might still lead to path ensembles receiving a trajectory missing the right resolution at the relevant interface. That part of the trajectory would then have to be reintegrated by MD. While all these issues can be solved in theory, it puts quite some challenges to the implementation. Moreover, if the integrator is not deterministic, but involves a thermostat or barostat, the one-step crossing might still not be guaranteed. Several velocity generation steps might still be needed. These challenges lead us to derive the WF move that straightforwardly can be implemented in present path sampling codes like OpenPathSampling[18,19] and PyRETIS[20,21] with, potentially, any MD engine.

VII. WIRE FENCING

Compared to the SS and WT moves, the shooting point selection of the WF move is constructed to avoid the one-step crossing issue altogether. Instead of restricting the shooting point to sets of crossing points at an interface, WF allows any phase point between the path ensemble's specific ensemble interface, λ_i, and interface λ_{B} to be picked. To increase the efficiency of the WF move in systems with asymmetric free energy barriers (See Fig. 3), the selection range and the boundaries of the subtrajectories can be changed by replacing λ_{B} with a user-defined *cap*-interface, λ_{cap} with $\lambda_i < \lambda_{\text{cap}} \leq \lambda_{\text{B}}$ value.

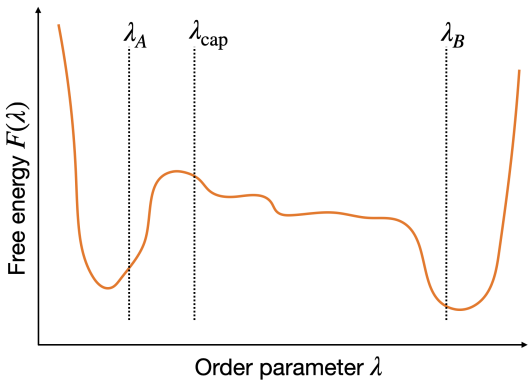


FIG. 3. Illustration of an asymmetric barrier where the placement of a cap-interface, λ_{cap}, in WF can avoid the generation of long subtrajectories and too many shooting points being in the basin of attraction of state B.

The presence of a relatively flat downhill region after the barrier's maximum and before a stable product state implies that transition paths can become very long. If accepted, the paths will have a large fraction of points at the right side of the free energy barrier from which shooting has a very high chance to generate a failed $\lambda_B \to \lambda_B$ trajectory. This problem was also addressed by the spring-shooting method.[43]

In an AIMD level simulation of aqueous silicate condensation,[44] this issue was solved by defining λ_B in the RETIS algorithm at the position of λ_{cap} in the figure. After the simulation was completed, all paths reaching λ_B were extended in a straightforward MD simulation. The introduction of the λ_{cap} interface makes these post-simulation MD extensions redundant.

We will first outline the WF algorithm without a cap-interface (or $\lambda_{\text{cap}} = \lambda_B$) using the high-acceptance protocol. The introduction of the λ_{cap} only requires a few modifications that we discuss afterward.

1. From the old path, count the number of frames $M_{\lambda_i}^{(o)}$ between λ_i and λ_B. If $M_{\lambda_i}^{(o)} = 0$ we immediately reject the full MC move. Otherwise continue with the next step.

2. Subdivide the $M_{\lambda_i}^{(o)}$ points into groups where each group are the points lying on a segment connecting λ_i with λ_B or a segment connecting λ_i with itself.

3. Select one segment as s^0 based on a weighted random selection such that each segment has a chance to be selected proportional to the number of points it has.

4. Set two counters n_s and n_a equal to zero: $n_s = n_a = 0$. Then start the following loop: steps 5 to 12.

5. Select at random one of the configuration points of the last subpath, s^{n_s}, as the new shooting point.

6. Generate random velocities from a Maxwell-Boltzmann distribution.

7. Starting from the configuration point with the new velocities, apply the MD integrator to go backward and forward in time until λ_i or λ_{B} is crossed.

8. Increase the n_s counter by one: $n_s = n_s + 1$.

9. If both time-directions crosses λ_{B}, the trial subpath is rejected. In that case, the previous successful subpath is kept, $s^{n_s} = s^{n_s-1}$. Go to step 12. Otherwise, continue with the next step.

10. Increase the n_a counter by one: $n_a = n_a + 1$.

11. Accept the trial subpath such that it becomes s^{n_s}.

12. If $n_s < N_s$, go to step 5. Otherwise, continue with next step.

13. If no accepted subpaths have been generated, $n_a = 0$, stop and reject the move. Otherwise, continue with the next step.

14. Extend the last subpath s^{N_s} in both time-directions with MD until λ_{A} or λ_{B} is hit. If the path ends at λ_{B} at both time-directions, the whole MC move is rejected. Otherwise, continue to the next step

15. If the path is $\lambda_B \to \lambda_A$, reverse the time-direction of the path.

16. Now a new full path has successfully been established. Let $q^{(n)}$ be 2 if it is a $\lambda_A \to \lambda_B$ path. Otherwise, it is 1. Let $M_{\lambda_i}^{(n)}$ be the number of frames between λ_i and λ_B. The weight-factor of the path is $w^{(n)} = q^{(n)} M_{\lambda_i}^{(n)}$ that is needed for computing proper path ensemble averages, Eq. 30, and for a possible swap move via Eq. 31.

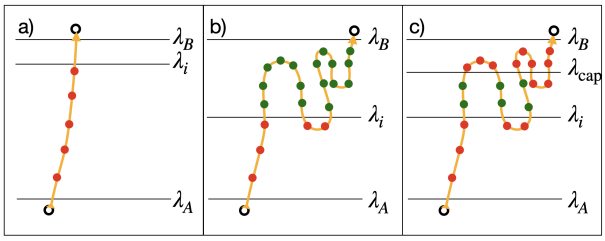


FIG. 4. Illustration of the s^0 selection from the old path. Selectable shooting points are shown in green, end-points by open black circles, and all other points in red. a) shows the "jumpy order parameter" case that leads to an immediate rejection as no selectable points are present. b) and c) show the selectable points without and with cap-interface, respectively.

The scenario of the potential rejection at step 1, is shown if Fig. 4-a) which can occur due to a jumpy character of the order parameter.[42] A typical example is nucleation where the time steps in path sampling is usually chosen to consist of many MD steps[45] for the reason that computing order parameters for nucleation is rather costly. As a result, occasionally the order parameter, defined by the size of the largest cluster, can make sudden jumps such that more than one interface is crossed in a single RETIS time step.

The path shown in Fig. 4-a) is a valid path in $[i^+]$ such that $\mathbf{1}_{[i^+]} = 1$, but $w_i = 0$ since $M_{\lambda_i} = 0$. In a WF move such a path has zero probability to be generated. Yet, its contribution in Eq. 30 to the average, if hypothetically sampled, would be $w_i^{-1} = \infty$ and, therefore, the sampling average becomes ill-defined. This can be solved by not allowing $w = 0$ weights:

$$w_i(\text{path}) = \min\left[1, q(\text{path}) M_{\lambda_i}(\text{path})\right] \qquad (34)$$

Introducing this small modification of Eq. 28 solves the "division by zero" problem and has further no impact of

the implementation nor on the robustness of the algorithm. The existence of jumpy trajectories implies that a pure WF simulation is no longer ergodic. A path like the one in Fig. 4-a) can never be made from a WF move and, vice versa, it can not be destroyed by the WF move if it is fed as the initial path to the algorithm. However, the full sampling remains ergodic due to the replica exchange moves.

Step 2 is further illustrated in Fig. 4-b). We can identify two groups of selectable shooting points (in green), one group of seven points lying on a $\lambda_i \to \lambda_i$ segment and one group of nine points on a $\lambda_i \to \lambda_B$ segment. So these segments are selected as s^0 with a 7/16 and 9/16 probability, respectively. In the next step, the points of the selected segment have an equal probability to be selected for the first shooting.

Despite that all the green points have the same 1/16 probability to be selected for shooting off the first subpath, the two-step selection process is needed to fix s^0. With a single step selection, it could be possible to first obtain a failed trial path t^1 that starts from a point at the first group, followed by a successful subtrajectory that is launched from a point of the second group. This will break the superdetailed balance as it would not be possible to generate t^1 from s^0 in the reverse path (see the example construction paths in Eqs. 2 and 3).

The introduction of the cap-interface changes the initial s^0 selection as is shown in Fig. 4-c) where, for the same path as panel b), there are now three groups of two points that can be chosen. Note that not all the points between λ_i and λ_{cap} are selectable as the points on a $\lambda_{\text{cap}} \to \lambda_{\text{cap}}$ segment should be excluded. The algorithm is further identical as described above with λ_{cap} instead of λ_B in the main loop (steps 5 to 12). Outside the main loop (step 13-15), λ_B is not replaced by λ_{cap} since the final extension always shall reach the A or B states. In the final step (16), $M_{\lambda_i}^{(n)}$ is replaced with the number of frames between λ_i and λ_{cap} excluding those on $\lambda_{\text{cap}} \to \lambda_{\text{cap}}$ segments.

VIII. NUMERICAL RESULTS

We tested the WF algorithm on three model systems: a simple one-dimensional system for which we can perform full RETIS simulations with high convergence, and two challenging complex systems based on classical MD and AIMD, where our analysis is more qualitative based on a single path ensemble simulation. The one-dimensional system describes a single particle in a double-well potential that is moving following the underdamped Langevin equation as previously described in Ref. 33. The purpose of these simulations is to show numerically that the WF method leads indeed to exact results. In addition, due to the high degree of convergence that can be reached, we also draw some conclusions on the efficiency compared to standard shooting. However, it should be taken into account that a larger boost factor is expected for more complex high-dimensional systems.

The other two systems are part of ongoing projects on which we plan to report extensively in later publications. The classical MD system describes the thin film breakage in oil-water mixtures based on the studies Ref. 46–49. The system size of this simulation is over 100,000 atoms making the one-step crossing impracticable as it requires a stop/restart at every MD step. Instead, in our single path ensemble simulation the coordinates were recorded every 50 MD steps. The AIMD system describes the electron transfer between ruthenium ions in a redox reaction taking place in liquid water. To determine the relative position of the moving electron, the Kohn-Sham orbitals are projected on maximally localized Wannier Functions[50] whose centers can be viewed as "electron positions". This implies that in order to compute the order parameter from a configuration point, a full electronic structure calculation is required. A cheap prediction scheme as described in Sec. VI is therefore not suitable. For both systems, we show the usefulness of the cap-interface in practical simulations.

A. Double-well 1D barrier

Despite the model's simplicity, several popular rare event simulation methods, like forward flux sampling (FFS)[51,52] and other splitting based methods,[53–55] have shown that they can easily fall into a kind of sampling trap when applied to this system yielding a too low rate and non time-symmetric transition paths.[33]

The double-well barrier system consists of a one-dimensional particle moving in the following potential[33]

$$V(z) = z^4 - 2z^2 \tag{35}$$

with underdamped Langevin dynamics. In reduced units, the Boltzmann constant and mass are set to unity, $k_B = m = 1$, while the temperature and friction coefficient are set equal to $T = 0.07$ and $\gamma = 0.3$. The equations of motion are propagated using an MD time step equal to $\mathrm{d}t = 0.025$. In a straightforward MD run, the particle will mostly oscillate within one of the potential minima at $z = -1$ and $z = 1$, but also (very) infrequently cross the transition state at z = 0. During the oscillatory movement, the total energy of the particle will fluctuate by the random force of the Langevin dynamics. As a result, the system is effectively two-dimensional in phase space where the velocity can be considered as an orthogonal degree of freedom. The reason that FFS and other splitting type methods underestimate the crossing rate is due to an insufficient sampling of the tail in the velocity distribution.[33] Path sampling methods like RETIS which are based on both forward and backward in time propagation do not have this issue.

We defined eight RETIS interfaces: $\lambda_A = \lambda_0 = -0.99$, $\lambda_1 = -0.8$, $\lambda_2 = -0.7$, $\lambda_3 = -0.6$, $\lambda_4 = -0.5$, $\lambda_5 = -0.4$, $\lambda_6 = -0.3$, and $\lambda_B = \lambda_7 = 1.0$, and ran four RETIS simulations using the PyRETIS code[20,21] consisting of 200,000 cycles. In all simulations (Shooting, WF, WF*,

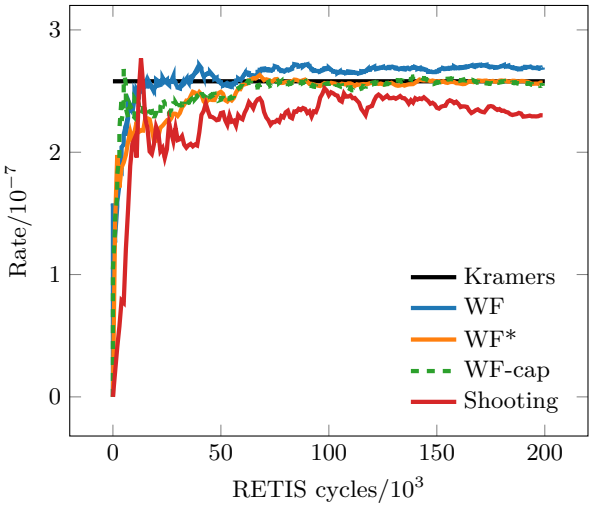


FIG. 5. Total running average of the computed rate as function of RETIS cycles.

WF-cap), each path ensemble either employs only shooting or only WF as the main MC move in addition to replica exchange moves. In the simulation "Shooting" all path ensembles employ the shooting move. In the other simulations the WF move is used for most path ensembles. However, simulation WF* uses normal shooting in the $[0^-]$ ensemble, while simulations WF and WF-cap use the shooting move in both the $[0^-]$ and $[0^+]$ ensemble as was suggested in Sec. III. The WF-cap simulation uses a cap-interface at $\lambda_{\text{cap}} = 0.1$ At each cycle, all path ensembles are updated with an ensemble move (shooting or WF) or with replica exchange moves with a 50%-50% probability. In case that a replica exchange move is selected, another 50%-50% probability determines whether the $[0^-] \leftrightarrow [0^+]$, $[1^+] \leftrightarrow [2^+]$, ..., $[5^+] \leftrightarrow [6^+]$ swaps will be attempted or the $[0^+] \leftrightarrow [1^+]$, $[2^+] \leftrightarrow [3^+]$, ..., $[4^+] \leftrightarrow [5^+]$ swaps. In the latter case, the $[0^-]$ and $[6^+]$ ensembles simply duplicate the previous path (null move). In the WF simulations, the number of subpaths was arbitrarily set equal to $N_s = 6$ for all path ensembles.

The results are shown in Fig. 5 and in table I where they are compared with Kramers' theory[56] which, for this system, can be considered as a nearly exact reference. Fig. 5 shows that the WF based simulations rapidly converge close to the Kramers' value of the rate confirming the exactness of the superdetailed balance relations and the correct implementation in the PyRETIS code. The results based on shooting are further off, but have a significantly lower computational cost per RETIS cycle (see table I).

Based on the relative errors from the block averaging analysis and the cost per cycle, we can compute the CPU efficiency time for each method, shown in the last column. Based on these numbers, we can see that the WF, WF* and WF-cap simulations are 2.5, 2.4 and 2.7 times more efficient than the simulation in which all path ensembles use the standard shooting move as their main MC move. Note that an improvement of more than a factor 2 is rather remarkable given the low dimensionality of the system.

TABLE I. Simulation data for the double-well 1D barrier system. The cost column describes the total number of calculated MD steps. The errors are based on block averaging using single standard deviations. The final column shows the CPU efficiency times[37] corresponding to the number of required MD steps for obtaining a relative error equal to 1. Simulation "Shooting" uses the standard shooting move as the main MC move in all path ensembles. The other simulations use the WF move in all ensembles except for $[0^-]$ (WF, WF*, WF-cap) and $[0^+]$ (WF, WF-cap). WF-cap uses a cap-interface at $\lambda_{\text{cap}} = 0.1$.

Simulation	Rate/10^{-7}	ϵ_r (%)	Cost/10^7	Cost·ϵ_r^2/10^{11}
Shooting	2.30	6.46	5.32	222.0
WF	2.69	2.28	16.98	88.3
WF*	2.58	2.19	19.56	93.9
WF-cap	2.54	2.29	15.72	82.4
Kramers	2.58			

In table II, we further examine the acceptance probabilities of the different moves. It is apparent that in all simulations the main MC move has a nearly 100% acceptance in the path ensembles where the WF move is employed thanks to the high-acceptance protocol. The acceptance is marginally lower at the last path ensembles $[5^+]$ and $[6^+]$ from which there is a higher probability to generate $\lambda_B \to \lambda_B$ paths. The shooting move has a lower acceptance, but has the advantage that all swapping moves with the $[0^-]$ ensemble are accepted if shooting is the main move in both $[0^-]$ and $[0^+]$. Since $[0^-]$ can only swap with $[0^+]$, these are the computationally expensive $[0^-] \leftrightarrow [0^+]$ swaps.

The other swapping moves are inexpensive as they do not require any MD steps. Therefore, an anticipated lower acceptance for these swapping moves in the WF simulations would not be dramatic. However, even this is not always the case. At first sight this appears counter-intuitive. Given a pair of paths in two neighboring ensembles, the standard swap should always have an acceptance probability that is equal to or higher than the acceptance based on Eq. 31. However, this effect can be canceled by the path distributions not being the same. Since the altered path distribution in the high-acceptance scheme, Eq. 28, overrepresents paths with many points between λ_i and λ_B or λ_{cap}, the $[i^+]$ path ensemble is likely to contain a higher fraction of paths crossing λ_{i+1}. From the data of table II, this seems indeed the case in the majority of path ensembles.

B. Thin Film Breakage

A system of 1100 dodecane molecules layered on a slab of 23936 water molecules is studied in the NPT ensemble via full atom TIS simulations using the GROMACS 2020.1

TABLE II. Acceptance ratio (%). Simulation "Shooting" uses the standard shooting move as main MC move in all path ensembles. The other simulations use the WF move in all ensembles except for $[0^-]$ (WF, WF*, WF-cap) and $[0^+]$ (WF, WF-cap).

ens.	shooting		WF		WF*		WF-cap	
	main	swap	main	swap	main	swap	main	swap
$[0^-]$	84.6	100.0	84.3	100.0	84.5	83.9	84.3	100.0
$[0^+]$	84.2	57.8	84.0	55.6	100.0	49.0	84.0	55.8
$[1^+]$	48.8	15.5	100.0	16.3	100.0	17.9	100.0	16.8
$[2^+]$	37.8	13.4	100.0	19.9	100.0	19.7	100.0	20.2
$[3^+]$	32.2	11.5	100.0	18.5	100.0	18.0	100.0	18.4
$[4^+]$	30.1	12.2	100.0	20.6	100.0	20.3	100.0	20.4
$[5^+]$	30.0	14.7	99.8	28.2	99.9	28.3	100.0	26.6
$[6^+]$	29.1	16.7	99.2	33.9	99.2	34.2	100.0	30.8

simulation package[57] as the external engine. The dodecane molecules are simulated according to the OPLS-AA force field[58] and the water molecules with the TIP4p/2005 model.[59] The preparation of the initial equilibrated system is explained in detail by Ref. 46. The temperature is set to 300 K and is controlled with a velocity rescaling method[60] employing a coupling time of 0.1 ps. Pressure is controlled by the Berendsen barostat and its normal component is maintained constant at 1 bar, with a time constant of 1.0 ps and compressibility coefficient of $4.7 \cdot 10^{-5}\text{bar}^{-1}$. The velocity-Verlet algorithm is used to solve the Newton equations of motion with a timestep of 0.002 ps. Periodic boundary conditions are applied in all directions, with the z direction being perpendicular to the 2D film. The box size is set to equal a box size of $15 \times 15 \times 5.1983$ nm.

The order parameter of the system is calculated by discretizing the system into 85×85 tiles in the x and y direction such that the order parameter value becomes the number of empty dodecane tiles that also have empty neighbors in the x and y direction. Such a definition provides a way to measure the presence of low-density regions, in addition to any breakage or "hole" formation that occurs within a trajectory. The sensitivity of the order parameter is determined by the specified discretizing size. In our case, the order parameter values fluctuated between 0 and 5 during an equilibrium run at $T = 300$ K. Based on this, we set $\lambda_A = 5$. We further defined $\lambda_B = 100$ as preliminary analysis showed that from this point on the hole tends to grow further with a negligible chance to close again.

To obtain an initial reactive trajectory, we ran an equilibrium run at $T = 375$ K until the thin film broke down. For our single path ensemble analysis we further defined $\lambda_\text{i} = 10.0$ as the interface that has to be crossed. In addition, we set the cap-interface $\lambda_\text{cap} = 15.0$. We then created 1000 trajectories using standard shooting and WF with $N_s = 10$. Three exemplary trajectories from the WF simulation are shown in Fig. 6-a).

From the sample size of 1000 MC moves, the acceptance in WF was equal to 73.4% and 35.0% for standard shooting. The limited sample size prohibits accurate CPU efficiency analysis, but a qualitative assertion of the sampling effectivity can be obtained by viewing the simulated path lengths as function of the MC step.

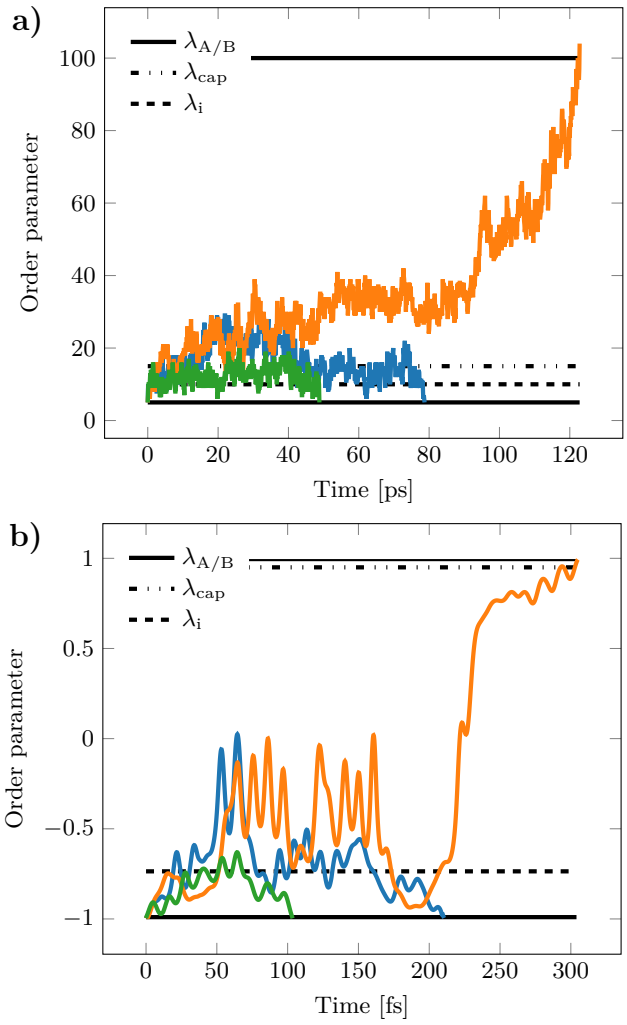


FIG. 6. Exemplary trajectories from the WF algorithm in the $[i^+]$ path ensemble showing the progress of the order parameter versus time. The stable state interfaces λ_A, λ_B, the cap-interface λ_cap, and the ensemble interface λ_i are shown as well. The two different panels represent the a) classical MD level simulation of the thin film breakage and b) the AIMD level simulations of the ruthenium self-exchange reaction.

Fig. 7-a) shows that the WF sampling has much more frequent transitions between long and short paths whereas shooting is mostly stuck in the short path domain. Once the shooting move manages to produce a long path, the path remains in the MC chain due to a long series of rejections (e.g. around step 500 where the same path length remains for a number of steps due to rejections). This indicates that the shooting move is struggling to properly sample path space. Even if the acceptance is not extremely low for the short paths, it fails to make regular switches to the longer paths. Moreover, if a long path is generated, the subsequent moves are likely rejected such that other longer paths are not likely found.

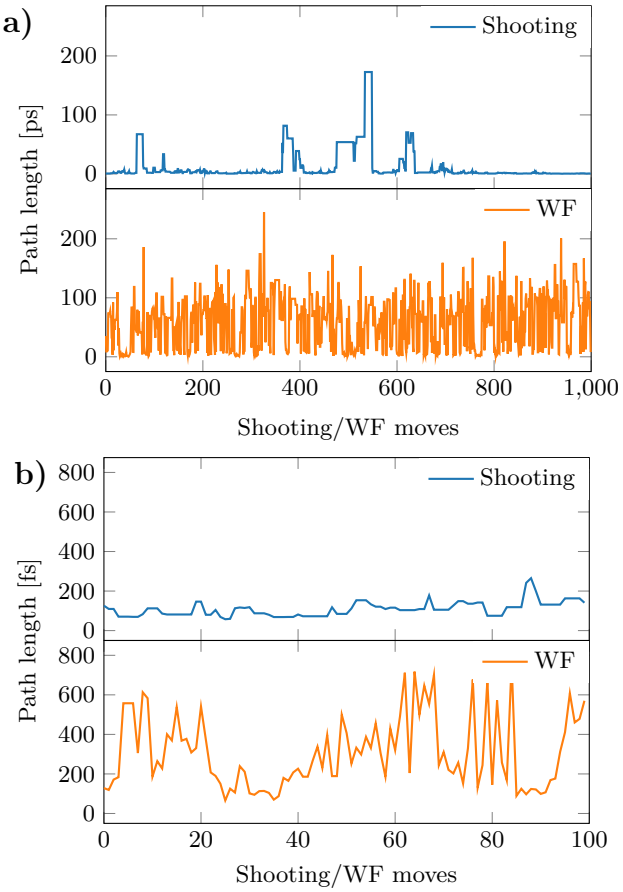


FIG. 7. Path length vs MC move for WF and standard shooting for a) classical MD system of thin film breakage and b) AIMD system of ruthenium self-exchange reaction.

C. Ruthenium-Ruthenium Self-Exchange Reaction

We studied the self-exchange reaction between two ruthenium ions in aqueous solution described by the following chemical reaction

$$Ru^{2+} + Ru^{3+} \rightarrow Ru^{3+} + Ru^{2+} \tag{36}$$

The simulation system consisted of two ruthenium ions, 63 H_2O molecules and one OH^- ion. The dynamics were propagated using NVE velocity-Verlet and the CP2K[25] simulation package. The effect of temperature was introduced via the randomization of velocities from a Maxwell-Boltzmann distribution at a temperature of 300 K. We used a time step of 0.5 fs and periodic boundary conditions were applied to a cubic box with an edge length of 12.4138 Å. Further simulation details on functional and basis sets are explained in Ref. 61.

To monitor the reaction progress, the electron transfer has been "followed" by transforming the occupied Kohn-Sham orbitals[62] into maximally localized Wannier functions (MLWF)[50] and computing the distance between the center of these localized functions (X) describing the moving electron to each of the ruthenium ions. The order parameter of the system is then defined as

$$\lambda = \frac{(d_{\mathrm{Ru}-X} - d_{\mathrm{Ru}'-X})}{d_{\mathrm{Ru}-\mathrm{Ru}'}} \tag{37}$$

where $d_{\mathrm{Ru}-X}$ is the distance between X and the initial ruthenium electron donor, $d_{\mathrm{Ru}'-X}$ is the distance between X and the initial ruthenium electron acceptor and $d_{\mathrm{Ru}-\mathrm{Ru}'}$ is the distance between the two ruthenium ions in the system. In this formulation, $\lambda = -1$ and $\lambda = +1$ define the reactant state and product state, respectively. Ru^{2+}/Ru^{3+} have 5/6 d-electrons and H_2O/OH^- have 8 valence electrons. This means there are a total of 523 MLWFs in the system. The order parameter, Eq. 37, requires the location X of the transferring electron, which is one of the centers of these 523 MLWFs. To identify which is X, each Wannier center is linked to either a ruthenium or oxygen atom that is closest. Then, if one ruthenium ion has 6 associated MLWFs, X is set to be the one that is the farthest away from this ruthenium ion. In the case that both ruthenium ions have 5 associated MLWFs, one of the oxygens has an excess MLWF (9 instead of 8), and the center that is farthest away from this oxygen is set as X.

To qualitatively compare standard shooting and WF for this system we run two single path ensemble simulations representing $[i^+]$ with $\lambda_i = -0.736$, $\lambda_A = -0.99$, and $\lambda_B = +0.99$. The value for $\lambda_i = -0.736$ was chosen from preliminary runs where we aimed for a 20% probability that a path ends up at state B. In the WF simulation, an additional $\lambda_{\mathrm{cap}} = +0.95$ was set to avoid $\lambda_B \rightarrow \lambda_B$ rejections due to the selection of shooting points lying within the basin of attraction of state B. Here, we only applied a rather modest number of subtrajectories $N_s = 2$. Higher performances might be obtained with a larger number of subpaths. Exemplary trajectories of the WF simulation are shown in Fig 6-b).

Due to the relatively low value of N_s, the subpath contribution to the total WF computational cost is only 15%. The acceptance probability increased from the shooting move's 48% to WF's 96%. Similarly to the classical MD system, the WF simulation seems to show a better sample exploration when we look at the path length as function of the MC step (Fig. 7-b)). The standard shooting algorithm seems not to be able to produce any paths larger than 300 fs. The WF algorithm, however, started with a short initial path but was able to quickly move up to the 600 fs range and making regular transitions between the shorter and longer paths. So also here, the sampling quality of the WF algorithm appears substantially superior to the one of standard shooting.

IX. CONCLUDING REMARKS

We reviewed the recently developed subtrajectory moves stone skipping (SS) and web throwing (WT) and added a new member to this group: wire fencing (WF).

These moves are more efficient than the standard shooting move which has been the main MC move for path sampling simulations during the last two decades. The subtrajectory moves proceed from a complete old path to a complete new path via a series of intermediate short paths (subpaths/subtrajectories). While this increases the average cost of a MC step, the correlations between paths are substantially reduced leading to a lower statistical inefficiency. The use of shorter paths resembles approximate path sampling methods like PPTIS or milestoning. However, the subtrajectory moves are still exact like standard shooting as they are based on mathematically rigorous superdetailed balance relations. The approach is preferably combined with a high-acceptance protocol in which the sampling distribution of the paths is adjusted in order to maximize the acceptance of newly generated trajectories. The effect of the biased distribution is undone in the post-simulation analysis using appropriate reweighting. The SS and WT move, however, require a one-step crossing condition which complicates their implementation and we discussed several solutions for this issue. The new WF does not rely on the one-step crossing condition and is therefore the most practical solution to the aforementioned problem even if it is slightly more wasteful than SS and WT. The WF move is in particular useful when the path sampling code uses an external MD engine and/or when the computation of the order parameter is costly. We showed the exactness and the efficiency gain of the WF approach in a RETIS simulation where the transition rate of an underdamped Langevin particle in a double-well potential has been computed and compared with the analytical Kramers' expression. Thereafter, we showed qualitatively how the WF move performs in a classical MD system describing the thin film breaking process and in an AIMD level system describing an electron transfer process between ruthenium ions in aqueous solution. In both cases, the WF move seems to allow a faster sampling through path space than standard shooting, which was concluded from the rapid switches that WF made between the shorter and longer paths.

ACKNOWLEDGMENTS

We acknowledge funding from the Research Council of Norway (toppforsk project Theolight, grant no. 275506) and computational resources from NOTUR (project nn9254k).

AUTHOR DECLARATIONS

The authors have no conflicts to disclose.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request. The algorithmic developments have been implemented in the current pyretis2.dev version (the current release is PyRETIS-2[21]) and will be included in the forthcoming main release (PyRETIS-3). The code and input files for the double-well 1D barrier system is already available at https://gitlab.com/pyretis following the FAIR principle for scientific data and software and data.[63,64]

Appendix A: Analytical expressions for the statistical inefficiency in model systems

Sec. IV introduces a model where at each MC move j there is a chance of π_R that the state of the system remains essentially unchanged and a chance of $\pi_M = 1 - \pi_R$ to "throw a dice". The latter implies that at step j the output value of h_j equals 1 with a probability p and 0 with a probability $1 - p$. Let us consider the conditional probability that $h_j = 0$ given that $h_0 = 0$: $P(h_j = 0|h_0 = 0)$. We can distinguish two scenarios. Scenario 1 relates to the case that all j moves implied a "remain" and therefore $h_j = h_0 = 0$. Scenario 2 is related to the situation that at least once the dice was thrown. In this scenario we have that h_j is either 1 or 0 with respective probabilities p and $1 - p$. The probability of having scenario 1 equals π_R^j and that of scenario 2 equals $1 - \pi_R^j$. Therefore,

$$\begin{aligned} P(h_j = 0|h_0 = 0) &= \pi_R^j + (1 - \pi_R^j)(1 - p) \\ &= (1 - p) + p\pi_R^j \end{aligned} \tag{A1}$$

Likewise, we can derive all the other conditional probabilities:

$$\begin{aligned} &P(h_j = 1|h_0 = 0) = (1 - \pi_R^j)p = p - p\pi_R^j \\ &P(h_j = 0|h_0 = 1) = (1 - \pi_R^j)(1 - p) = (1 - p) + (1 - p)\pi_R^j \\ &P(h_j = 1|h_0 = 1) = \pi_R^j + (1 - \pi_R^j)p = p + (1 - p)\pi_R^j \end{aligned} \tag{A2}$$

Let us call $p_{kl} = P(h_j = k \wedge h_0 = l) = P(h_j = k|h_0 = l)P(h_0 = l)$. From Eqs. A1 and A2 we can derive:

$$\begin{aligned} p_{00} &= (1 - p)^2 + p(1 - p)\pi_R^j \\ p_{10} &= p(1 - p) - p(1 - p)\pi_R^j = p_{01} \\ p_{11} &= p^2 + p(1 - p)\pi_R^j \end{aligned} \tag{A3}$$

and from this we can compute

$$\begin{aligned} \langle (h_0 - p)(h_j - p) \rangle &= p_{00}p^2 - p_{10}(1 - p)p \\ &\quad - p_{01}(1 - p)p + p_{11}(1 - p)^2 \end{aligned} \tag{A4}$$

In the above expression, all the π_R-independent terms cancel. This is expected since we know the result equals 0

if $\pi_R = 0$. The remaining π_R-dependent terms sum up to

$$\begin{aligned} &p(1-p)\pi_R^j \left[p^2 + 2p(1-p) + (1-p)^2\right] \\ &= p(1-p)\pi_R^j \left[p + (1-p)\right]^2 = p(1-p)\pi_R^j \end{aligned} \tag{A5}$$

From Eqs. 16, 17, and A5 we derive that

$$C(j) = \pi_R^j \Rightarrow n_c = \frac{\pi_R}{1-\pi_R} \tag{A6}$$

and via Eq. 15:

$$\mathcal{N} = 1 + 2\frac{\pi_R}{1-\pi_R} = \frac{1+\pi_R}{1-\pi_R} \tag{A7}$$

As $\pi_M = 1 - \pi_R$, A7 is equivalent to Eq. 19 of Sec. IV.

In the second model we assume $\pi_R = 0$, but there are two phases $x = \alpha, \beta$ that have, respectively, probabilities P_α and P_β and local crossing probabilities p_α and p_β. Let $\pi_\rho = 1 - \pi_\mu$ be the probability that the MC maintains the previous phase. The inverse probability π_μ implies throwing the dice to determine the phase x such that the selection probability for x corresponds to P_α and $P_\beta = 1 - P_\alpha$. After the phase x is set, h_j will be set to 1 or 0 with respective probabilities p_x and $(1-p_x)$. Given that the phase of the first sample equals $x_0 = x$, the chance that the j-th sample has the same or opposite phase equals, respectively, $\pi_\rho^j + (1-\pi_\rho^j)P_x$ and $(1-\pi_\rho^j)(1-P_x)$. This leads to the following conditional probabilities akin Eqs. A1 and A2:

$$\begin{aligned} P(h_j = 0|x_0 = x) &= \pi_\rho^j(1-p_x) + (1-\pi_\rho^j)(1-p) \\ &= \pi_\rho^j(p-p_x) + (1-p) \\ &= \pi_\rho^j P_y(p_y - p_x) + (1-p) \\ P(h_j = 1|x_0 = x) &= \pi_\rho^j p_x + (1-\pi_\rho^j)p \\ &= \pi_\rho^j(p_x - p) + p \\ &= \pi_\rho^j P_y(p_x - p_y) + p \end{aligned} \tag{A8}$$

where $y \in (\alpha, \beta)$ and $y \neq x$. Hence, analogous to Eqs. A3

$$\begin{aligned} p_{k0} &= \sum_{x=\alpha,\beta} P_x(1-p_x)P(h_j = k|x_0 = x) \\ p_{k1} &= \sum_{x=\alpha,\beta} P_x p_x P(h_j = k|x_0 = x) \end{aligned} \tag{A9}$$

which leads to

$$\begin{aligned} p_{00} &= (1-p)^2 + \pi_\rho^j \sum_x P_x P_y (1-p_x)(p_y - p_x) \\ &= (1-p)^2 + \pi_\rho^j P_\alpha P_\beta (p_\alpha - p_\beta)^2 \\ p_{10} &= p(1-p) + \pi_\rho^j \sum_x P_x P_y (1-p_x)(p_x - p_y) \\ &= p(1-p) - \pi_\rho^j P_\alpha P_\beta (p_\alpha - p_\beta)^2 = p_{01} \\ p_{11} &= p^2 + \pi_\rho^j \sum_x P_x P_y p_x (p_x - p_y) \\ &= p^2 + \pi_\rho^j P_\alpha P_\beta (p_\alpha - p_\beta)^2 \end{aligned} \tag{A10}$$

Analogous to Eqs. A4 and A5 we find that

$$\langle (h_0 - p)(h_j - p) \rangle = \pi_\rho^j P_\alpha P_\beta (p_\alpha - p_\beta)^2 \tag{A11}$$

and like Eq. A6:

$$\begin{aligned} C(j) &= \frac{P_\alpha P_\beta (p_\alpha - p_\beta)^2}{p(1-p)} \pi_\rho^j = K_s \pi_\rho^j \\ \Rightarrow n_c &= K_s \frac{\pi_\rho}{1-\pi_\rho} = K_s \frac{1-\pi_\mu}{\pi_\mu} \end{aligned} \tag{A12}$$

where we used $\pi_\mu = 1 - \pi_\rho$ and Eq. 21. Substitution of Eq. A12 in Eq. 15 leads to Eq. 20.

[1] K. Lindorff-Larsen, S. Piana, R. O. Dror, and D. E. Shaw, Science **334**, 517 (2011).

[2] D. E. Shaw, P. J. Adams, A. Azaria, J. A. Bank, B. Batson, A. Bell, M. Bergdorf, J. Bhatt, J. A. Butts, T. Correia, R. M. Dirks, R. O. Dror, M. P. Eastwood, B. Edwards, A. Even, P. Feldmann, M. Fenn, C. H. Fenton, A. Forte, J. Gagliardo, G. Gill, M. Gorlatova, B. Greskamp, J. Grossman, J. Gullingsrud, A. Harper, W. Hasenplaugh, M. Heily, B. C. Heshmat, J. Hunt, D. J. Ierardi, L. Iserovich, B. L. Jackson, N. P. Johnson, M. M. Kirk, J. L. Klepeis, J. S. Kuskin, K. M. Mackenzie, R. J. Mader, R. McGowen, A. McLaughlin, M. A. Moraes, M. H. Nasr, L. J. Nociolo, L. O'Donnell, A. Parker, J. L. Peticolas, G. Pocina, C. Predescu, T. Quan, J. K. Salmon, C. Schwink, K. S. Shim, N. Siddique, J. Spengler, T. Szalay, R. Tabladillo, R. Tartler, A. G. Taube, M. Theobald, B. Towles, W. Vick, S. C. Wang, M. Wazlowski, M. J. Weingarten, J. M. Williams, and K. A. Yuh, in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '21 (Association for Computing Machinery, New York, NY, USA, 2021).

[3] M. E. Goldberg, G. V. Semisotnov, B. Friguet, K. Kuwajima, O. B. Ptitsyn, and S. Sugai, FEBS Lett. **263**, 51 (1990).

[4] B. Peters, *Reaction rate theory and rare events* (Elsevier, Amsterdam, Netherlands, 2017).

[5] T. S. van Erp, D. Moroni, and P. G. Bolhuis, J. Chem. Phys. **118**, 7762 (2003).

[6] T. van Erp, Phys. Rev. Lett. **98**, 268301 (2007).

[7] C. Dellago, P. G. Bolhuis, F. S. Csajka, and D. Chandler, J. Chem. Phys. **108**, 1964 (1998).

[8] R. Cabriolu, K. M. S. Refsnes, P. G. Bolhuis, and T. S. van Erp, J. Chem. Phys. **147**, 152722 (2017).

[9] A. Arjun and P. G. Bolhuis, J. Phys. Chem. B **124**, 8099 (2020).

[10] M. Moqadam, A. Lervik, E. Riccardi, V. Venkatraman, B. K. Alsberg, and T. S. van Erp, Proc. Natl. Acad. Sci. USA **115**, E4569 (2018).

[11] M. Eigen and L. de Maeyer, Proc. R. Soc. Lond. A Math. Phys. Sci. **247**, 505 (1958).

[12] W. C. Natzle and C. B. Moore, J. Phys. Chem. **89**, 2605 (1985).

[13] D. Moroni, P. Bolhuis, and T. van Erp, J. Chem. Phys. **120**, 1044 (2003).

[14] A. K. Faradjian and R. Elber, J. Chem. Phys. **120**, 10880 (2004).

[15] S. Roet, D. T. Zhang, and T. S. van Erp, J. Phys. Chem. A **126**, 8878 (2022).

[16] C. Dellago, P. G. Bolhuis, and D. Chandler, The Journal of Chemical Physics **108**, 9236 (1998), https://doi.org/10.1063/1.476378.

[17] E. Riccardi, O. Dahlen, and T. S. van Erp, J. Phys. Chem. Lett. **8**, 4456 (2017).

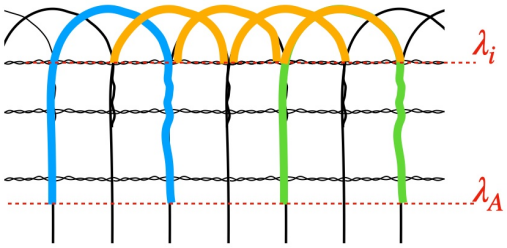
[18] D. W. H. Swenson, J.-H. Prinz, F. Noe, J. D. Chodera, and P. G. Bolhuis, J. Chem. Theory Comput. **15**, 813 (2019).

[19] D. W. H. Swenson, J.-H. Prinz, F. Noe, J. D. Chodera, and P. G. Bolhuis, J. Chem. Theory Comput. **15**, 837 (2019).

[20] A. Lervik, E. Riccardi, and T. S. van Erp, J. Comput. Chem. **38**, 2439 (2017).

[21] E. Riccardi, A. Lervik, S. Roet, O. Aaroen, and T. S. van Erp, J. Comput. Chem. **41**, 370 (2020).

[22]M. J. Abraham, T. Murtola, R. Schulz, S. Páll, J. C. Smith, B. Hess, and E. Lindahl, SoftwareX **1-2**, 19 (2015).
[23]S. Plimpton, J. Comput. Phys. **117**, 1 (1995).
[24]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, and V. S. Pande, J. Chem. Theory Comput. **9**, 461 (2013).
[25]J. Hutter, M. Iannuzzi, F. Schiffmann, and J. VandeVondele, WileyWIREs Comput Mol Sci **4**, 15 (2014).
[26]A. Stukowski, Model. Simul. Mater. Sci. Eng. **20**, 045021 (2012).
[27]S. Winczewski, J. Dziedzic, and J. Rybicki, Comput. Phys. Commun. **198**, 128 (2016).
[28]N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, J. Chem. Phys. **21**, 1087 (1953).
[29]D. Frenkel and B. Smit, *Understanding molecular simulations from algorithms to applications* (Academic press, San Diego, California, U.S.A., 2002).
[30]J. I. Siepmann and D. Frenkel, Mol. Phys. **75**, 59 (1992).
[31]T. Vlugt, R. Krishna, and B. Smit, J. Phys. Chem. B **103**, 1102 (1999).
[32]W. Hastings, Biometrika **57**, 97 (1970).
[33]T. van Erp, Adv. Chem. Phys. **151**, 27 (2012).
[34]A. Ghysels, S. Roet, S. Davoudi, and T. S. van Erp, Phys. Rev. Research **3**, 033068 (2021).
[35]C. Dellago and P. G. Bolhuis, Mol. Simu. **30**, 795 (2004).
[36]T. van Erp and P. Bolhuis, J. Comput. Phys. **205**, 157 (2005).
[37]T. S. van Erp, J. Chem. Phys. **125**, 174106 (2006).
[38]A. Ferrenberg and R. Swendsen, Phys. Rev. Lett. **63**, 1195 (1989).
[39]T. S. van Erp, M. Moqadam, E. Riccardi, and A. Lervik, J. Chem. Theory Comput. **12**, 5398 (2016).
[40]J. Rogal, W. Lechner, J. Juraszek, B. Ensing, and P. G. Bolhuis, J. Comp. Phys. **133**, 174109 (2010).
[41]E. Vanden-Eijnden, M. Venturoli, G. Ciccotti, and R. Elber, J. Comp. Phys. **129**, 174102 (2008).
[42]A. Haji-Akbari, J. Chem. Phys. **149**, 072303 (2018).
[43]Z. F. Brotzakis and P. G. Bolhuis, J. Chem. Phys. **145**, 164112 (2016).
[44]M. Moqadam, E. Riccardi, T. T. Trinh, A. Lervik, and T. S. van Erp, Phys. Chem. Chem. Phys. **19**, 13361 (2017).
[45]D. Moroni, P. R. ten Wolde, and P. G. Bolhuis, Phys. Rev. Lett. **94**, 235703 (2005).
[46]O. Aarøen, E. Riccardi, T. S. v. Erp, and M. Sletmoen, Colloids and Surfaces A: Physicochemical and Engineering Aspects **632**, 127808 (2022).
[47]O. Aarøen, E. Riccardi, and M. Sletmoen, RSC advances **11**, 8730 (2021).
[48]E. Riccardi and T. Tichelkamp, Colloids and Surfaces A: Physicochemical and Engineering Aspects **573**, 246 (2019).
[49]E. Riccardi, K. Kovalchuk, A. Y. Mehandzhiyski, and B. A. Grimes, Journal of dispersion science and technology **35**, 1018 (2014).
[50]N. Marzari, A. A. Mostofi, J. R. Yates, I. Souza, and D. Vanderbilt, Rev. Mod. Phys. **84**, 1419 (2012).
[51]R. J. Allen, C. Valeriani, and P. R. ten Wolde, J. Phys.-Condes. Matter **21**, 463102 (2009).
[52]F. A. Escobedo, E. E. Borrero, and J. C. Araque, J. Phys.-Condes. Matter **21**, 333101 (2009).
[53]T. E. Booth and J. S. Hendricks, Nucl. Techno.-Fus. **5**, 90 (1984).
[54]P. Melnik-Melnikov and E. Dekhtyaruk, Probab. Eng. Eng. Mech. **15**, 125 (2000).
[55]M. Villenaltamirano and J. Villenaltamirano, in *Queueing, Performance and Control in Atm*, North-Holland Studies in Telecommunication, Vol. 15, edited by Cohen, J W and Pack, C D (Elsevier Science Publ B V, Amsterdam, 1991) pp. 71–76, 13th International Teletraffic Congress (ITC-13), Copenhagen, Denmark, Jun 19-26, 1991.
[56]P. Hanggi, P. Talkner, and M. Borkovec, Rev. Mod. Phys. **62**, 251 (1990).
[57]M. J. Abraham, T. Murtola, R. Schulz, S. Páll, J. C. Smith, B. Hess, and E. Lindahl, SoftwareX **1**, 19 (2015).
[58]W. L. Jorgensen, D. S. Maxwell, and J. Tirado-Rives, Journal of the American Chemical Society **118**, 11225 (1996).
[59]J. L. F. Abascal and C. Vega, J. Chem. Phys. **123**, 234505 (2005).
[60]G. Bussi, D. Donadio, and M. Parrinello, J. Chem. Phys. **126**, 014101 (2007).
[61]A. Tiwari and B. Ensing, Faraday Discussions **195**, 291 (2016).
[62]W. Kohn and L. J. Sham, Phys. Rev. **140**, A1133 (1965).
[63]E. Riccardi, S. Pantano, and R. Potestio, Interface Focus **9**, 20190005 (2019).
[64]A.-L. Lamprecht, L. Garcia, M. Kuzak, C. Martinez, R. Arcila, E. Martin Del Pico, V. Dominguez Del Angel, S. Van De Sandt, J. Ison, P. A. Martinez, *et al.*, Data Science **3**, 37 (2020).



TOC