UNIVERSITY OF AMSTERDAM

UvA-DARE (Digital Academic Repository)

Transition path sampling for non-equilibrium dynamics without predefined reaction coordinates

Buijsman, P.; Bolhuis, P.G.

DOI
10.1063/1.5130760

Publication date
2020

Document Version
Final published version

Published in
Journal of Chemical Physics

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

Link to publication

Citation for published version (APA):
Buijsman, P., & Bolhuis, P. G. (2020). Transition path sampling for non-equilibrium dynamics without predefined reaction coordinates. *Journal of Chemical Physics*, *152*(4), Article 044108. https://doi.org/10.1063/1.5130760

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

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

Download date:15 Nov 2026

The Journal of Chemical Physics

Transition path sampling for non-equilibrium dynamics without predefined reaction coordinates

Cite as: J. Chem. Phys. **152**, 044108 (2020); https://doi.org/10.1063/1.5130760
Submitted: 07 October 2019 . Accepted: 08 January 2020 . Published Online: 28 January 2020

P. Buijsman, and P. G. Bolhuis

View Online Export Citation CrossMark

ARTICLES YOU MAY BE INTERESTED IN

Studying rare events using forward-flux sampling: Recent breakthroughs and future outlook
The Journal of Chemical Physics **152**, 060901 (2020); https://doi.org/10.1063/1.5127780

Machine learning for interatomic potential models
The Journal of Chemical Physics **152**, 050902 (2020); https://doi.org/10.1063/1.5126336

Approximating free energy and committor landscapes in standard transition path sampling using virtual interface exchange
The Journal of Chemical Physics **151**, 174111 (2019); https://doi.org/10.1063/1.5119252

© 2020 Author(s).

The Journal of Chemical Physics ARTICLE scitation.org/journal/jcp

Transition path sampling for non-equilibrium dynamics without predefined reaction coordinates

Cite as: J. Chem. Phys. 152, 044108 (2020); doi: 10.1063/1.5130760
Submitted: 7 October 2019 • Accepted: 8 January 2020 •
Published Online: 28 January 2020

View Online Export Citation CrossMark

P. Buijsman and P. G. Bolhuis[a)]

AFFILIATIONS

van 't Hoff Institute for Molecular Sciences and Amsterdam Center for Multiscale Modeling, University of Amsterdam, PO Box 94157, 1090 GD Amsterdam, The Netherlands

[a)]Author to whom correspondence should be addressed: p.g.bolhuis@uva.nl

ABSTRACT

We develop two novel transition path sampling (TPS) algorithms for harvesting ensembles of rare event trajectories using non-equilibrium dynamics. These methods have the advantage that no predefined reaction coordinate is needed. Instead, an instantaneous reaction coordinate is based on the current path. Constituting a Monte Carlo random walk in trajectory space, the algorithms can be viewed as bridging between the original TPS methodology and the Rosenbluth based forward flux sampling methodology. We illustrate the new methods on toy models undergoing equilibrium and non-equilibrium dynamics, including an active Brownian particle system. For the latter, we find that transitions between steady states occur via states that are locally ordered but globally disordered.

Published under license by AIP Publishing. https://doi.org/10.1063/1.5130760

I. INTRODUCTION

Transition path sampling (TPS) is a powerful method to investigate rare events in complex systems[1–5] that harvests an ensemble of reactive trajectories between predefined stable basins. While many variants exist, the most efficient basic type transition path sampling uses the shooting algorithm in which a trial path is created by choosing a random frame on the current transition path and integrating the equation of motion (shooting) backward and forward in time.[2] This trial path can then be accepted or rejected according to a Metropolis criterion. This procedure results in a random walk in trajectory space that converges to the correct natural distribution of transition paths. This path ensemble (PE) can be scrutinized to extract the optimal reaction coordinate.[6] Indeed, the main attractive point of TPS is that it does not impose but rather gives unbiased insight into the reaction coordinate. In addition, one can compute the rate constant by slowly transforming the unconstrained path ensemble in the initial state toward the fully constrained transition path ensemble. This computation can be expensive, and a particularly efficient approach was introduced with the transition interface sampling (TIS) approach.[7,8] In TIS, the rate constant is computed by computing the flux of trajectories through a series of so-called interfaces: high dimensional surfaces characterized by a single order parameter. For each interface, one samples trajectories crossing that particular interface and determines the fraction of trajectories that reach the next interface. The rate constant follows from combining these fractions with the direct effective positive flux through the first interface.

While the standard TPS (and TIS) approach has seen many improvements, extensions, and applications over the years, it is fundamentally limited to situations where microscopic reversibility is satisfied.[3,5,9] This includes most standard molecular dynamics, (overdamped or Brownian) Langevin dynamics, and Monte Carlo dynamics. It also applies to relaxation of metastable systems, e.g., undercooled liquids nucleating into a solid. However, the backward shooting algorithm is not applicable to systems evolving with dynamics that lack microscopic reversibility, e.g., internal driven dynamics, self-propulsion, or some types of coarse grained dynamics.[5,9] Yet, these dynamical systems are of interest and can also show rare event behavior, which could be investigated with transition path sampling. This was recognized already shortly after the invention of TPS, and several algorithms were especially developed to

Published under license by AIP Publishing

solve this problem. One of the earliest ones was the noise history approach by Crooks and Chandler,[10] in which paths were changed by altering the noise terms in the stochastic integrator. However, this approach only works when paths do not change too much with a variation of a noise term. Later on, other path sampling methods were developed that did not suffer from this, notably the forward flux methods (FFS),[11,12] non-equilibrium umbrella sampling (NEUS),[13] steered TPS (STePS),[14,15] and a noise guidance algorithm to sample long correlated stochastic trajectories.[15] Other important methods are the weighted ensemble[16,17] and stochastic process rare event sampling.[18] Furthermore, much research has been done on large deviation theory and cloning algorithms.[19–21]

By far, the most influential TPS related method is FFS.[12] This method was especially developed with the non-equilibrium nature of the dynamics in mind. The lack of microscopic reversibility means that the backward shot is not allowed in a shooting algorithm; hence, one can only shoot forward in time. Based on the framework of TIS,[7,8,22] FFS employs a series of interfaces and computes the probability of trajectories starting from a surface to reach the next interface. The conventional standard algorithm is direct FFS (dFFS). Starting in the initial state, straightforward dynamics leads to sufficient spontaneous crossings of the first interface. These first crossing frames are collected and act as seeds for spawning (shooting) new trajectories, most of which return to the initial state, but some make it to the next, further out, interface. The first crossing frames for this next interface are again collected, and the procedure is repeated until the entire barrier is crossed. The result is a set of transition paths connecting A with B, using only forward shots and, moreover, a measure of the crossing probability and, thus, the rate constant. The FFS methodology also got many extensions, notably the branch growth methods and non-stationary FFS.[12,23]

Comparing TPS/TIS with FFS, we can discern some similarities: both use dynamical trajectories, both use a shooting (spawning) type algorithm to create new paths, both use a history dependent rate expression, and both result in a path ensemble and a crossing probability. However, there are also differences: TPS/TIS creates a Markov chain through path space, while FFS creates a set of trajectories by ratcheting along the interfaces (note that the Rosenbluth–FFS scheme also allows for creating a Markov chain Monte Carlo algorithm[12]). In contrast, the original TPS algorithm (but not TIS) is independent of the choice of a reaction coordinate. Moreover, as already mentioned, FFS purely uses forward shots, while TPS/TIS also uses backward shooting. One of the main effects of the latter is that dFFS is dependent on sufficient sampling of frames in the first interfaces that will eventually lead to the final state. That is, the order parameter used should be sufficiently close to the real reaction coordinate. If not, the first interface will be sparse in frames that eventually will lead to the final state, and the whole subsequent path ensemble will depend on it.[24] In other words, the path ensemble is not allowed to relax during the crossing of the barrier. One of the great advantages of TPS/TIS is that the path ensemble *can* relax as a trial path follows from perturbing an existing pathway slightly. This perturbation can be tuned, in contrast to FFS where new paths are created from scratch. The dependence of the FFS algorithm on the initial interface is partly compensated by the fact that FFS can handle non-equilibrium dynamics, something that TPS cannot, which was indeed the reason for the development of FFS in the first place.[11]

The purpose of this work is to develop two novel methods that combine the strengths of both FFS and TPS methods: (1) being able to handle non-equilibrium dynamics, (2) being able to provide a Monte Carlo walk through phase space so that the path ensemble can relax, and (3) being independent of an *a priori* chosen reaction coordinate. To do so, we return to the original TPS paper,[1] which introduced a Monte Carlo sampling of paths that was based on a guiding field. The idea was to start in the initial state and construct a "space-time polymer" using a configurational bias Monte Carlo (CBMC) scheme that would select trajectories that are closer to the final state. This leads to a Rosenbluth weight for a trial path, which can be used in a Metropolis acceptance criterion. This kind of Rosenbluth scheme later was also implemented in the FFS framework,[12] where the guiding field was replaced with the interface crossing criterion.[25] Indeed, as the guiding field is determined by an order parameter, it is conceptually similar to a set of interfaces. The downside of both the interfaces and guiding fields is that the order parameter has to be predefined and can in fact be wrong. Of course, optimization and iteration is possible, but a method that is not, or at least much less, dependent on this choice should be beneficial. In this paper, we develop two of those methods. In our first approach, the current path is used to define a guiding field that biases the generation of a trial path toward the final region. To determine the progress of the new path along the old path, we use the path variable concept of Branduardi *et al.*[26] In the second approach, the path progress variable is used to define interfaces that help the trial path in navigating toward the final region. Both algorithms construct a full trial path ratcheting along an existing path and compute the Rosenbluth weights of the old and the new path to decide whether or not to accept the trial path. This results in the sampling of a path ensemble that can relax to the most optimal channel, without ever defining an explicit order parameter for interfaces. Because the approach only takes into account forward integration, it is suitable for non-equilibrium dynamics.

The remainder of the paper is organized as follows. In Sec. II, we introduce the novel algorithms. In Sec. III, we show the correctness of the methods using simple models. We end with conclusions.

II. METHODS

We introduce two new path sampling methods based on the configurational bias Monte Carlo (CBMC) scheme outlined in the original TPS paper[1] and on the Rosenbluth based forward flux sampling (RBFFS). See Appendix A for a short overview of these methods. The novel algorithms sample paths in a procedure similar to the RBFFS method: paths are generated by repeatedly shooting off a number of trial path segments and selecting one segment to continue the path generation with. In contrast to the RBFFS method, however, the new methods do not rely on the fixed definition of an order parameter but rather use an existing path to navigate through phase space toward the final region B. The order parameter is thus replaced by a "progress variable" that measures the progress of a trial path along an existing path in the transition path ensemble (TPE) (see Fig. 1). This means that the methods require no (or at least much less) knowledge of the reaction coordinate, although it is required that the stable regions A and B are properly defined. (Note that replacing the order parameter by the progress variable is not

Published under license by AIP Publishing

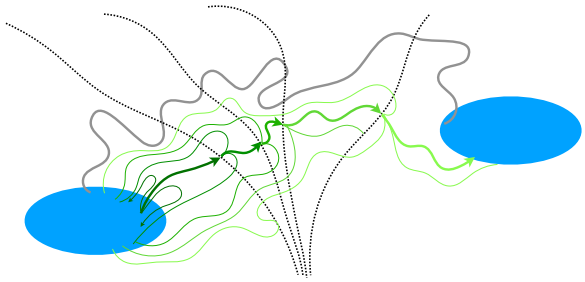

FIG. 1. Illustration of the novel approach in which several segments (thin green lines) are grown, and the one that traverses furthest along the progress variable has the highest probability to be selected to be part of the final trial trajectory (thick green curves). The dotted curves are isosurfaces of the progress variable as calculated with respect to the previous path (solid gray curve). The two algorithms differ in how they construct and select the segments.

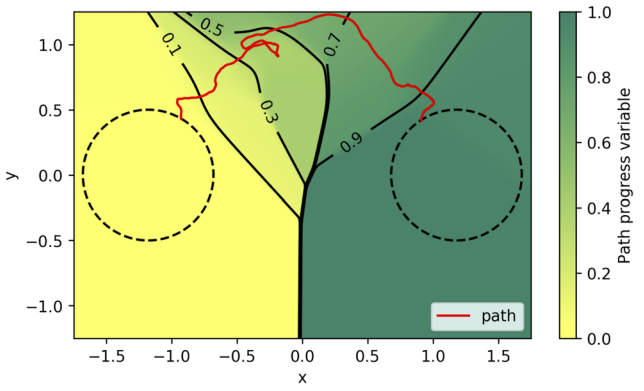

FIG. 2. Illustration of the path progress variable.[26] The transition path is indicated in red and the stable states with dashed circles. Solid black curves denote the interfaces along the progress variable.

completely order parameter free as we still need a metric to measure that progress variable.) Imposing a detailed balance in the Metropolis criterion requires that the progress of the old path is also measured along the new trial path.

A. Measuring path progress

To make things concrete, we first explain how we measure the progress along an existing path. In both novel algorithms, we use the path variable introduced by Branduardi *et al.*[26] to define the guiding field and the interfaces. For this path variable, it is assumed that the transition of interest can be described by a set of collective coordinates $R(x')$ that depend on the system configuration x'. For example, R could be the set of Cartesian coordinates of all particles in a potential energy surface, in which case the number of coordinates can be rather large. In other systems, such as molecular reactions in a solvent, R can be the relative positions of only a subset of all particles in the system (ignoring, for instance, the solvent itself). In general, R can be a function that maps the configuration onto a set of collective variables. A transition path $\mathbf{x} = \{x_0, x_1, \ldots, x_L\}$ gives rise to a sequence $\mathbf{R} = \{R(x_0), R(x_1), \ldots, R(x_L)\}$ that is in turn used to define the path variable,

$$\mathscr{S}(R(x')|\mathbf{x}) = \frac{1}{L}\frac{\sum_{l=0}^{L} l \cdot \exp[-\alpha(R(x') - R(x_l))^2]}{\sum_{l=0}^{L} \exp[-\alpha(R(x') - R(x_l))^2]}, \tag{1}$$

where $\alpha \geq 0$ is a parameter that should be chosen as large as possible. In practice, it is sufficient to take α that is of the order of the inverse of the time interval between subsequent states in a path. The term $(R(x') - R(x_l))^2$ can be calculated as the mean square displacement, but it is also possible to use other types of metrics.[26] Figure 2 illustrates the concept of this path progress variable for an arbitrary (but actual) trajectory in a 2D plane. We stress that both algorithms detailed in Secs. II B and II C are in fact not dependent on this precise definition of the path progress, and other path progress measures could be developed.

B. Path based CBMC

The Path Based CBMC (PBCBMC) is inspired by both the CBMC based TPS algorithm[1] and the RBFFS method.[12] The idea is to define a guiding field using an existing path and to bias the generation of a new path along this guiding field.

The PBCBMC method starts with an MD simulation in the stable region A, which is halted once the system leaves region A and crosses the first interface λ_1. (For consistency with the TIS and FFS methods, it is assumed that the first interface λ_1 is defined and lies close to the boundary of region A, λ_0, although it is possible and often convenient to set $\lambda_1 = \lambda_0$.[9]) The resulting system configuration is then used to initiate a "jet" consisting of K_1 trial path segments that are continued for a maximum number of integration steps L_1 (equivalent to a duration $\Delta t L_1$) or until entering one of the stable states. If a path segment enters state A, it is considered "unsuccessful", and it is given a weight of 0. Otherwise, the segment is considered successful, and it is assigned a weight $w^{(n)}$ that depends on its progress with respect to the old path $\mathbf{x}^{(o)}$. Thus,

$$w^{(n)} = \begin{cases} \mathscr{F}(x_{\text{end}}|\mathbf{x}^{(o)}) & \text{if } x_{\text{end}} \notin A \\ 0 & \text{if } x_{\text{end}} \in A, \end{cases} \tag{2}$$

where x_{end} denotes the endpoint of the trial segment and $\mathscr{F}(x|\mathbf{x}^{(o)})$ denotes the guiding field defined by the old path $\mathbf{x}^{(o)}$. The guiding field $\mathscr{F}$ should be defined in the entire phase space and should be increasing toward the final stable region. For notational convenience, the dependence of $w^{(n)}$ on $\mathbf{x}^{(o)}$ is implicitly assumed in the superscript "(n)".

As there is considerable freedom in choosing $\mathscr{F}(x'|\mathbf{x})$, it is possible to define $\mathscr{F}(x'|\mathbf{x})$ to be a non-linear function of the progress variable defined in Eq. (1). For the PBCBMC method, we exploit this freedom by defining the guiding field as an exponential function of the path progress variable, which increases the bias toward successful trajectories. We set

$$\mathscr{F}_{\text{PBCBMC}}(x'|\mathbf{x}) \equiv \exp[b(\mathscr{S}(R(x')|\mathbf{x}) - 1)], \tag{3}$$

where b is an optional parameter determining the strength of the guiding field. The subtraction of 1 in the exponent of the equation above ensures that the guiding field remains normalized between

Published under license by AIP Publishing

0 and 1. Note that the validity of the algorithm is in general independent of the choice of $\mathscr{F}(x'|\mathbf{x})$.

After generating a jet of K_1 trial segments, it is assigned a weight $W_1^{(\mathrm{n})} = \sum_{k=1}^{K_1} w_k^{(\mathrm{n})}$ equal to the sum of the trial weights. If at least one trial was successful (that is, $W_1^{(\mathrm{n})} > 0$), then one of the trials is selected based on the relative weights $w_k^{(\mathrm{n})}/W_1^{(\mathrm{n})}$. The endpoint of the selected segment is then used as the starting point for the generation of a new jet consisting of K_2 trial path segments that continue for a maximum number of L_2 integration steps. This procedure is repeated until the selected segment ends in region B or when the generation of an intermediate jet J produced no successful segments and $W_J^{(\mathrm{n})} = 0$. In either case, the (normalized) Rosenbluth weight of the resulting trial path is defined as

$$\mathscr{W}^{(\mathrm{n})} = \prod_{j=1}^{J^{(\mathrm{n})}} \frac{W_j^{(\mathrm{n})}}{K_j} = \prod_{j=1}^{J^{(\mathrm{n})}} \frac{\sum_{k=1}^{K_j} w_k^{(\mathrm{n})}}{K_j}. \tag{4}$$

Here, $J^{(\mathrm{n})}$ is the total number of jets that were generated, which is variable because the generation of a trial path terminates once the selected segment ends in B. We stress that the weight $\mathscr{W}^{(\mathrm{n})}$ depends on the old path $\mathbf{x}^{(\mathrm{o})}$ through the biasing function $\mathscr{F}$ that determines the segment weights and, hence, the jet weights $W_j^{(\mathrm{n})}$. In order to maintain detailed balance, the Rosenbluth weight for the old path $\mathscr{W}^{(\mathrm{o})}$ must, therefore, be recalculated with the new trial path $\mathbf{x}^{(\mathrm{n})}$ as reference. Since the PBCBMC method initiates jets always from the same predefined indices 0, L_1, $L_1 + L_2$, ... in a path, it is possible to simply use the jets that were generated when creating the old path in order to calculate $\mathscr{W}^{(\mathrm{o})}$. The weight of each trial path segment is again given by Eq. (2), and the Rosenbluth weight for the old path is calculated with Eq. (4). However, the guiding field $\mathscr{F}(x|\mathbf{x}^{(\mathrm{n})})$ is now defined by the new trial path.

The final acceptance probability for paths generated in the PBCBMC method is given by

$$P_{\mathrm{acc}}[\mathbf{x}^{(\mathrm{o})} \to \mathbf{x}^{(\mathrm{n})}] = \min\left\{1, \frac{\mathscr{W}^{(\mathrm{n})}}{\mathscr{W}^{(\mathrm{o})}} \frac{\prod_{j'=1}^{J^{(\mathrm{o})}} w_{j',*}^{(\mathrm{o})}}{\prod_{j=1}^{J^{(\mathrm{n})}} w_{j,*}^{(\mathrm{n})}}\right\}. \tag{5}$$

Here, the asterisk denotes the index of the selected segment of a particular jet. A subtlety arises when $\mathscr{W}^{(\mathrm{n})} = 0$. In that case, the last jet produced no successful trials, and $w_{J^{(\mathrm{n})},*}^{(\mathrm{n})}$ is undefined since all trial segments have a weight of zero. Therefore, we set $w_{J,*} = 1$ if $w_{J,k} = 0$ for all $1 \le k \le K_J$. The proof that the PBCBMC method samples paths with the correct weights in the TPE is given in Appendix B.

C. Path based Rosenbluth

The Path Based Rosenbluth (PBRB) method is effectively a generalization of the RBFFS method. The main difference between the two methods is how interfaces are defined and, consequently, how the Rosenbluth weights are calculated. Instead of using an order parameter, the interfaces are now defined using a progress variable $\mathscr{F}(x|\mathbf{x}^{(\mathrm{o})})$ along an existing path. The resulting interfaces are then used to generate a trial path in the same way as that in the RBFFS method. An important distinction with the RBFFS method is that interfaces are no longer fixed but vary throughout phase space as paths are replaced with accepted trial paths (see Fig. 1). This also means that the Rosenbluth weights are no longer fixed but now depend on the path that was used to define the interfaces. To indicate that the interfaces depend on a certain path $\mathbf{x}^{(\mathrm{o})}$, the interfaces are denoted as level sets of the function $\mathscr{F}(x|\mathbf{x}^{(\mathrm{o})})$,

$$\lambda_i^{(\mathrm{o})} = \{x|\mathscr{F}(x|\mathbf{x}^{(\mathrm{o})}) = \mathscr{F}_i\}, \tag{6}$$

where $0 < \mathscr{F}_2 < \ldots < \mathscr{F}_{n-1} < 1$ are values that are chosen *a priori* and remain fixed throughout a simulation. It is most convenient to define the interfaces directly along the progress variable of Eq. (1) (see Fig. 2). We thus set

$$\mathscr{F}_{\mathrm{PBRB}}(x'|\mathbf{x}) \equiv \mathscr{S}(R(x')|\mathbf{x}), \tag{7}$$

where $\mathbf{x}$ can refer to both the old existing and the new trial path during the sampling. We stress again that the validity of the PBRB algorithm is in general independent of the choice of $\mathscr{F}(x'|\mathbf{x})$.

The set of interfaces is completed by adding $\lambda_0 \equiv \lambda_A$, λ_1, and $\lambda_n \equiv \lambda_B$. It is important to note that the shape of the intermediate interfaces is not known *a priori*, and it is, therefore, not guaranteed that the interfaces $\lambda_i^{(o,n)}$ do not intersect with the definitions of the stable regions. This is, however, not a problem, and it is possible to skip interfaces in case there is overlap.

The PBRB method starts with an MD simulation in the stable region A that continues until crossing λ_1. If the next interface $\lambda_2^{(\mathrm{o})}$ intersects with λ_1 and the resulting system configuration already crossed $\lambda_2^{(\mathrm{o})}$, the interface λ_1 is ignored and the procedure continues immediately with $\lambda_2^{(\mathrm{o})}$ (see Fig. 3). If the system configuration also already crossed the subsequent interfaces $\lambda_3^{(\mathrm{o})}, \ldots, \lambda_i^{(\mathrm{o})}$ as well, then all but the last one are ignored, and the procedure continues immediately with the interface $\lambda_i^{(\mathrm{o})}$.

The system configuration just across $\lambda_i^{(\mathrm{o})}$ (or λ_1) is used as the starting point to generate a jet of K_i path segments that continue until crossing λ_0 (returning to A), $\lambda_{i+1}^{(\mathrm{o})}$, or λ_n (reaching B). The trial segments that crossed $\lambda_{i+1}^{(\mathrm{o})}$ or λ_n are considered successful and are assigned a weight $w^{(\mathrm{n})}$ equal to 1. The remaining segments entered

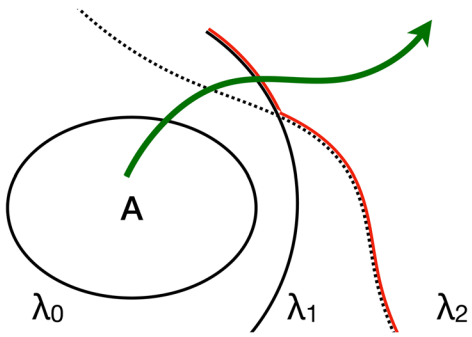

FIG. 3. Illustration of how to deal with interface intersections. When one of the instantaneous interfaces (λ_2) intersects the λ_1 interface, λ_1 is ignored and the procedure continues with λ_2. Effectively, the dotted interface is replaced by the solid red interface. The green trajectory thus crosses λ_1 and λ_2 simultaneously.

Published under license by AIP Publishing

A and are assigned a weight of 0. The weight $w^{(n)}$ of a path segment is thus given by

$$w^{(n)} = \begin{cases} 1 & \text{if } x_{end} \text{ crossed } \lambda_{i+1}^{(o)} \text{ or } \lambda_n \\ 0 & \text{if } x_{end} \in A, \end{cases} \tag{8}$$

where x_{end} denotes the endpoint of the segment. If at least one of the segments is successful, then one of the successful segments is selected at random, and its endpoint becomes the starting point for generating a new jet of trial segments that continue until crossing λ_0, $\lambda_{i+2}^{(o)}$, or λ_n. This procedure is repeated for the subsequent interfaces until the selected segment ends in B or if there were no successful segments produced at an earlier interface. If a selected segment crossed λ_n before crossing an intermediate interface $\lambda_j^{(o)}$, the remaining interfaces $\lambda_j^{(o)}, \ldots, \lambda_{n-1}^{(o)}$ are ignored.

When the generation of a trial path is finished, its Rosenbluth weight is determined by the same expression as in Eq. (4),

$$\mathscr{W}^{(n)} = \prod_{j=1}^{n-1} \frac{W_j^{(n)}}{K_j}, \tag{9}$$

where $W_j^{(n)}$ is now the number of successful trials generated from the jth interface. For the ignored interfaces, we set both $K_j = 1$ and $W_j^{(n)} = 1$. Again, we stress that the weight $\mathscr{W}^{(n)}$ depends on the old path $\mathbf{x}^{(o)}$ through the definition of the interfaces $\lambda_i^{(o)}$, and the Rosenbluth weight for the old path $\mathscr{W}^{(o)}$ must, therefore, be recalculated with the trial path as reference. For this, we define a new set of interfaces $\lambda_2^{(n)}, \ldots, \lambda_{n-1}^{(n)}$ along the trial path using the same $\mathscr{T}_i$ values that were used to define $\lambda_i^{(o)}$. In contrast to the PBCBMC method, it is no longer possible to use the jets that were produced in the old path because the starting point and endpoints of the jets do not match with the new set of interfaces. We, therefore, need to regrow trial segments around the old path and impose the condition of super-detailed balance, which states that detailed balance needs to be satisfied around each set of trial moves.[27] The procedure to regrow jets around the old path starts by finding the first crossing points of the old path with the interfaces $\lambda_1, \lambda_2^{(n)}, \ldots, \lambda_{n-1}^{(n)}$. If there is overlap between λ_1 and intermediate interfaces, and the first crossing point of the old path with λ_1 also crossed $\lambda_2^{(n)}, \ldots, \lambda_i^{(n)}$, then the first interfaces $\lambda_1, \lambda_2^{(n)}, \ldots, \lambda_{i-1}^{(n)}$ are ignored, and the corresponding weights are set to unity. Similarly, if there is overlap of the last interface λ_n with intermediate interfaces, and the endpoint of the path is also the first crossing point of the intermediate interfaces $\lambda_j^{(n)}, \ldots, \lambda_{n-1}^{(n)}$, then the intermediate interfaces $\lambda_j^{(n)}, \ldots, \lambda_{n-1}^{(n)}$ are ignored. The first crossing points of the old path with each considered interface $\lambda_i^{(n)}$ are then used to initiate $K_i - 1$ trial segments that continue until crossing $\lambda_0, \lambda_{i+1}^{(n)}$ or λ_n. The set of trial segments is completed by adding the path segment of the old path between $\lambda_i^{(n)}$ and $\lambda_{i+1}^{(n)}$, and the completed jet is given a weight $W_i^{(o)}$ equal to the number of successful trial segments. Note that $W_i^{(o)}$ is non-zero since the path segment from the old path is always successful. The normalized Rosenbluth weight $\mathscr{W}^{(o)}$ for the old path is given by the same expression as Eq. (9).

The final acceptance probability for paths generated with the PBRB method is the same as the acceptance probability for the PBCBMC method given in Eq. (5). In this case, however, the weight of the selected segments is always equal to 1, and the acceptance probability simplifies to

$$P_{acc}[\mathbf{x}^{(o)} \to \mathbf{x}^{(n)}] = \min\left\{1, \frac{\mathscr{W}^{(n)}}{\mathscr{W}^{(o)}}\right\}. \tag{10}$$

The proof that paths are sampled with the correct weights in the TPE is given in Appendix B.

D. Computing the transition rate constant

To compute the transition rate constant in the PBCBMC and the PBRB method, we factorize it as $k_{AB} = \Phi(\lambda_1|\lambda_A)P(\lambda_B|\lambda_1)$. The flux term $\Phi(\lambda_1|\lambda_0)$ is calculated in the exact same way as in the TIS and RBFFS methods and is simply given by the total number of generated trial paths divided by the total length of the MD simulation in the stable region A. The crossing probability is calculated using the factorization $P(\lambda_B|\lambda_1) = \prod_i P(\lambda_{i+1}|\lambda_i)$. It may seem that the factorization is not defined in the PBCBMC method since this method does not make use of interfaces. However, it is possible to define interfaces as abstract hyper-surfaces in the time dimension. The idea of using time as a substitute for the order parameter was already suggested in Ref. 8. Now, $P(\mathscr{T}_{i+1}, \mathscr{T}_i)$ is the probability that a path coming from λ_1 does not enter region A within a time $\mathscr{T}_{i+1}$, given that the path remained outside of A over a time $\mathscr{T}_i < \mathscr{T}_{i+1}$. This definition is consistent with the meaning of a successful path segment in PBCBMC: a path segment in the jth jet is successful if it did not enter region A. Hence, the factorization $P(\lambda_B|\lambda_1) = \prod_i P(\lambda_{i+1}|\lambda_i)$ is still defined, and the term $P(\lambda_{i+1}|\lambda_i)$ is simply given by the fraction of successful path segments in the ith jet.

In the TIS and the RBFFS method, the term $P(\lambda_{i+1}|\lambda_i)$ is estimated by computing the average $\langle P(\lambda_{i+1}|\lambda_i)\rangle_{\lambda_i}$ in the ensemble of paths that cross interface λ_i. For RBFFS, in particular, each ensemble is sampled separately by applying a Metropolis acceptance/rejection step at every interface.[12] In PBCBMC and PBRB, however, the Rosenbluth weight needs to be recomputed for the old path with the trial path as reference. While theoretically possible, an acceptance/rejection step at each interface makes the practical implementation of the algorithms a tedious and intractable task because each partial trial path needs to be stored. Furthermore, the efficiency of the PBRB method reduces significantly since each partial path comparison requires the generation of new path segments in order to maintain super-detailed balance. Therefore, it is more convenient to calculate $P(\lambda_{i+1}|\lambda_i)$ in the ensemble of paths that cross the first interface λ_1 by taking the weighted average. The ensemble averages are related by[27]

$$\langle P(\lambda_{i+1}|\lambda_i)\rangle_{\lambda_i} = \frac{\langle P(\lambda_{i+1}|\lambda_i)\tilde{\mathscr{W}}_{i-1}\rangle_{\lambda_1}}{\langle \tilde{\mathscr{W}}_{i-1}\rangle_{\lambda_1}}, \tag{11}$$

where $\tilde{\mathscr{W}}_{i-1}$ is the weight of a partial path that connects λ_1 with λ_i. This weight follows from the acceptance probabilities in the Metropolis acceptance criterion by considering only interfaces up to (and including) the interface λ_{i-1}. The weight is given by

Published under license by AIP Publishing

$$\tilde{\mathscr{W}}_{i-1} = \prod_{j<i} \frac{W_j}{K_j w_{j,*}}, \tag{12}$$

where K_j is the number of trials, $w_{j,*}$ is the weight of the selected path segment, and $W_j = \sum_{k=1}^{K_j}$ is the sum of the segment weights at the jth interface. Note that $\tilde{\mathscr{W}}$ differs from the normalized Rosenbluth weight, denoted with $\mathscr{W}$, because the former also includes the factors $w_{j,*}$. For the PBRB method, however, $w_{j,*} = 1$ and $\tilde{\mathscr{W}}$ reduces to $\mathscr{W}$. The total crossing probability is simply given by the product of the individual terms of Eq. (11). Hence,

$$\begin{aligned} P(\lambda_n|\lambda_1) &= \prod_i \langle P(\lambda_{i+1}|\lambda_i)\rangle_{\lambda_i} \\ &= \prod_i \frac{\langle P(\lambda_{i+1}|\lambda_i)\tilde{\mathscr{W}}_{i-1}\rangle_{\lambda_1}}{\langle \tilde{\mathscr{W}}_{i-1}\rangle_{\lambda_1}}. \end{aligned} \tag{13}$$

For the PBRB method, the expression above can be simplified by noting that $P(\lambda_{i+1}|\lambda_i)$ is estimated by the number of successful trials, relative to the total number of trials. In the PBRB method, this ratio is the same as the normalized weight $W_i^{(\mathrm{n})}/K_i$ of the ith jet. Hence,

$$P(\lambda_{i+1}|\lambda_i)\tilde{\mathscr{W}}_{i-1}^{(\mathrm{n})} = \frac{W_i^{(\mathrm{n})}}{K_i}\tilde{\mathscr{W}}_{i-1}^{(\mathrm{n})}. \tag{14}$$

In PBRB, $w_{j,*}^{(\mathrm{n})} = 1$, so we can divide the right-hand side by $w_{j,*}^{(\mathrm{n})}$. This division yields

$$\begin{aligned} P(\lambda_{i+1}|\lambda_i)\tilde{\mathscr{W}}_{i-1}^{(\mathrm{n})} &= \frac{W_i^{(\mathrm{n})}}{K_i w_{i,*}^{(\mathrm{n})}}\tilde{\mathscr{W}}_{i-1}^{(\mathrm{n})} = \frac{W_i^{(\mathrm{n})}}{K_i w_{i,*}^{(\mathrm{n})}} \prod_{j<i} \frac{W_j^{(\mathrm{n})}}{K_j w_{j,*}^{(\mathrm{n})}} \\ &= \prod_{j<i+1} \frac{W_j^{(\mathrm{n})}}{K_j w_{j,*}^{(\mathrm{n})}} = \tilde{\mathscr{W}}_i^{(\mathrm{n})}, \end{aligned} \tag{15}$$

where the second and the last equalities follows from the definition of $\tilde{\mathscr{W}}$ given in Eq. (12). Substituting the result of the equation above into Eq. (13) yields

$$P(\lambda_n|\lambda_1) = \prod_i \frac{\langle \tilde{\mathscr{W}}_i^{(\mathrm{n})}\rangle_{\lambda_1}}{\langle \tilde{\mathscr{W}}_{i-1}^{(\mathrm{n})}\rangle_{\lambda_1}} = \langle \tilde{\mathscr{W}}_n^{(\mathrm{n})}\rangle_{\lambda_1}. \tag{16}$$

The crossing probability is thus given by the average statistical weight of trial paths. The average includes a contribution of 0 for trial paths that did not end in region B. In the numerical verification of the algorithms, the crossing probability is computed according to Eq. (13) for the PBCBMC method and according to Eq. (16) for the PBRB method.

III. RESULTS AND DISCUSSION

The novel algorithms are numerically tested on a simple system consisting of a single particle in a two-dimensional potential energy surface (PES). For the first test case, we revert to the toy model that was used in the original TPS paper, where the force field is given by the negative gradient of the potential. The force is thus conservative, and the system is in equilibrium. For the second test case, the first system is taken out of equilibrium by applying an additional non-conservative force. In the third test case, we consider the Z-potential introduced by Rogal *et al.*,[28] which was designed to have a non-linear reaction coordinate. This makes it hard to come up with a suitable order parameter and is, therefore, a good test case for the path based methods. In these three systems, the results are compared with the Rosenbluth variant of the FFS method.

To show that the methods also work on a more complex test case, we give an application on an active system consisting of multiple self-propelled particles that exhibit collective behavior in a two-dimensional domain. Details of the models and simulation settings are given in Appendix C.

A. Toy model

We first test the novel methods on the potential from Ref. 1, shown in Fig. 4 (see Appendix C for details). To get an indication of the efficiency of the methods, we run each method for the same amount of "cycles". A cycle is defined as the generation of a trial path, which includes paths that did not enter region B and have a weight of 0. The number of cycles used here is 5×10^7 for both test cases. This number is rather large for the systems under consideration but allows us to compare the different methods with greater statistical accuracy. Each simulation is repeated 16 times, and the results are averaged. The standard deviation of the different simulations is used to compute error bars. The results for the conservative and non-conservative force fields will be discussed subsequently.

1. Conservative force field

As a first comparison between the methods, the acceptance probabilities are considered. For the methods used here, the acceptance probability can be decomposed into two factors: $p_{\mathrm{acc}} = p_{\mathrm{succ}} p_{\mathrm{cond}}$. The first factor p_{succ} measures the probability that a transition path is successful and reaches region B. The second factor p_{cond} is then the acceptance probability that a trial path is accepted, conditional on it being successful. The success rate p_{succ} should preferably be large, in which case the bias introduced by the methods is optimal and a large portion of trial paths reach B. In the worst case, p_{succ} is in the same order as the crossing probability itself

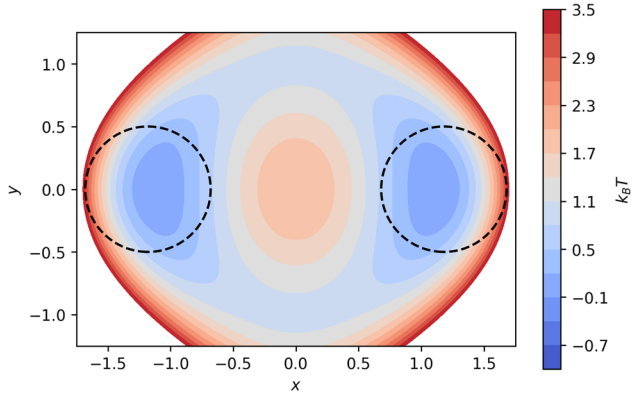

FIG. 4. Illustration of the simple two-dimensional potential energy surface. The boundaries of the stable regions A and B are indicated with black dashed lines.

Published under license by AIP Publishing

(or lower even), meaning that even a straightforward MD simulation would be more efficient to sample transition paths. For p_{cond} and, therefore, p_{acc}, there is no universal optimum that holds for all systems.[27,29] However, if p_{cond} is very low, then there are only a few transition paths that dominate the results. In general, "normal" values for p_{cond} roughly lie in the interval between 0.2 and 0.5.[27,30,31] In Table I, the success probability p_{succ} and conditional acceptance probability p_{cond} are listed for each of the methods. In addition, the average number of accepted paths per hour of simulation time is also shown. The latter gives an idea of how many unique transition paths are generated per central processing unit (CPU) time. A typical simulation required 22 h, 87 h, and 30 h of computation time for the PBCBMC, PBRB, and RBFFS methods, respectively, on the same type of CPU. Clearly, the paths generated with the RBFFS method have the highest probability of being successful. This indicates that for the current system, the bias introduced by the (static) interfaces of the RBFFS method is more optimal than the guiding field approach of the PBCBMC method and the varying interfaces of the PBRB method. Due to the low success probability in the PBCBMC method, the number of accepted trial paths per CPU time is also very low, even though the total simulation time of the PBCBMC method itself was shortest. The low acceptance ratio of the PBCBMC method compared to the other two methods can be explained by the difference in selecting a particular trial segment. The PBRB and RBFFS methods only select a trial if it crossed a certain interface, whereas the PBCBMC method selects any trial that did not enter region A with a non-zero probability. The low p_{succ} in PBCBMC can be improved by increasing the bias factor b, but this will at the same time lead to a lower acceptance due to wildly fluctuating Rosenbluth weights caused by new trial paths that are relatively far away from the old path. To ameliorate this, one needs to bias the trial trajectory also in the perpendicular path distance variable. We leave this for future investigation.

Another notable difference is that a typical simulation for the PBRB method required approximately 3–4 times as many CPU time compared to PBCBMC and RBFFS. The time difference is explained by the fact that the PBRB method requires the evaluation of the progress variable at each time step. This is an expensive operation since it requires the computation of the distance between the current point and all points in the reference path. In addition, the condition of super-detailed balance demands the recalculation of the Rosenbluth weight for the old path with the trial path as reference. This means that an additional set of trial path segments must be generated, which is not needed in the PBCBMC and RBFFS methods.

TABLE I. Acceptance probabilities in the equilibrium test case. The second column contains the average fraction of trial paths that are successful and reach region B, and the third column contains the acceptance probability of trial paths, conditional on them reaching B. The fourth column shows the average number of accepted paths per hour of simulation time.

Method	p_{succ}	p_{cond}	$N_{\mathrm{acc}}/T_{\mathrm{sim}}$
PBCBMC	0.023 60 ± 0.000 02	0.336 05 ± 0.000 59	17 985/h
PBRB	0.128 95 ± 0.000 05	0.406 17 ± 0.000 22	30 240/h
RBFFS	0.367 46 ± 0.000 07	0.302 42 ± 0.000 26	187 722/h

TABLE II. Results for the equilibrium toy model. The second column contains the transition rate constants, and the third column contains the average path length.

Method	$k_{AB}/10^{-5}$	$\langle L\rangle/10^{2}$
PBCBMC	3.925 54 ± 0.004 54	8.920 13 ± 0.010 70
PBRB	3.925 81 ± 0.002 07	8.921 67 ± 0.003 50
RBFFS	3.924 04 ± 0.002 48	8.920 87 ± 0.004 09

In the next comparison, we consider the transition rate constants and average path lengths. The results for these properties are listed in Table II. All results are, within error bar, equal for the three methods, and the transition rate constant also matches the result found by Dellago *et al.*[1] For both properties, the PBRB method has the lowest error, which means that out of the three methods, the PBRB method reached the most accurate estimate with the same amount of cycles. The PBCBMC method, on the other hand, has the lowest accuracy of the three methods, which is due to its relatively low success ratio p_{succ} (see Table I). This means that the correlation between subsequent paths is highest in the PBCBMC method, resulting in a lower accuracy.

To visualize the path ensemble, we plot the path density obtained with RBFFS in Fig. 5. Clearly, transition paths do not go straight over the barrier but, instead, follow one of the two possible channels. The path density is higher in the center of both channels near $x = 0$, where the potential is relatively flat and the dynamics of the particle is more diffusive.

For the RBFFS method, it is obvious that paths in either of the two channels are sampled with equal frequency since the domain (including the interfaces) is perfectly symmetric in $y = 0$. For the PBCBMC and PBRB methods, however, this symmetry is broken by the guiding field and the path interfaces because the path progress variable is not symmetric around $y = 0$. To illustrate this, we plotted the progress variable in Fig. 2 for a single path obtained with the PBRB method. The two stable regions are indicated with dashed black lines, and a path from the TPE is shown in red. The values of the progress variable are indicated with a color gradient going from

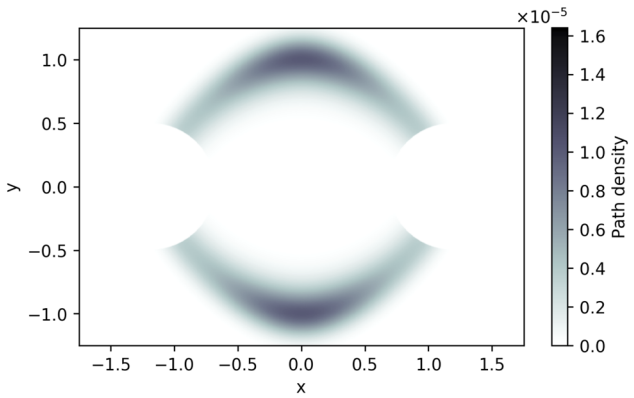

FIG. 5. Path density of trajectories sampled with the RBFFS method in the two-dimensional equilibrium toy model.

Published under license by AIP Publishing

yellow (low progress) to green (high progress). Five interfaces (in this case isolines) defined by the values {0.1, 0.3, 0.5, 0.7, 0.9} are indicated with solid black lines. For paths that go through the same channel as the existing pathway, the progress variable describes the progress toward region B very well. For paths that go through the opposite channel, on the other hand, the progress variable resembles a switching function that is close to 0 for $x < 0$ and close to 1 for $x > 0$. Hence, a trial path that goes through the same channel as the old path has a higher chance to reach B, and swapping channels might be difficult. Thus, it is not clear that PBCBMC and PBRB sample the two different channels with the same frequency.

To compare the path densities sampled with the PBCBMC and PBRB methods against the density sampled with the RBFFS method, the *absolute difference* between the path densities is plotted in Fig. 6. The absolute difference is obtained by subtracting the bin values of the RBFFS method from the bin values of the PBCBMC or PBRB method. Hence, if the absolute difference in a particular bin is negative (indicated with blue), the bin value was larger for the RBFFS method than for the PBCBMC or PBRB method. Conversely, if the absolute difference is positive (indicated with red), then the PBCBMC or PBRB method has a larger value than the RBFFS method. Note the difference in scales between Figs. 6 and 5. For the PBRB [panel (b)], there is no systematic pattern and the noisy pattern is likely caused by random fluctuations due to finite sampling. For the PBCBMC method [panel (a)], the density in the bottom channel is slightly larger than the density in the top channel, but the differences are minor. In the PBCBMC method, 49.851% of the density is contained in the bins for which $y > 0$, which means that the top and bottom channels are sampled with approximately the same frequency. For the RBFFS and PBRB methods, respectively, 49.920% and 49.923% of the density are contained in the bins for which $y > 0$. Thus, the PBCBMC and PBRB methods are not restricted to a single reaction channel and allow sufficient swapping between the two channels. We note that such dependence on swapping (also known as switching) is similar to the original TPS algorithm, which in fact we are trying to mimic.

Finally, we compare the convergence properties of the three methods. In Fig. 7, we plot the rate constant and its standard deviation as a function of the (logarithmic) number of cycles. All

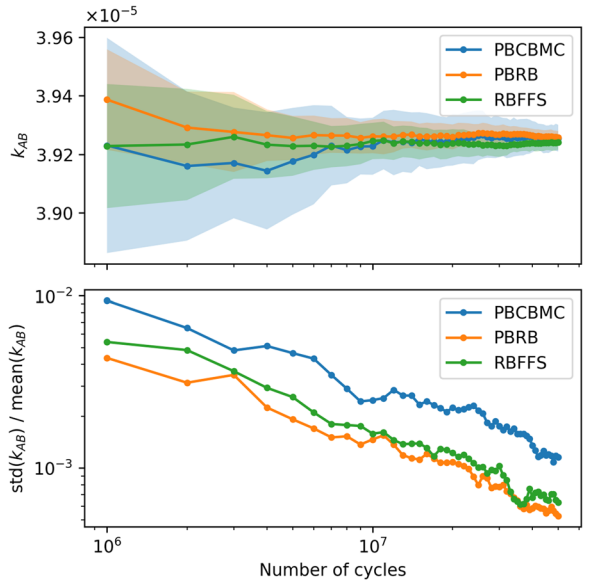

FIG. 7. (Top) Convergence of the rate constant as a function of the number of simulation cycles for the three methods. Shaded areas indicate the error bars. (Bottom) Log–log plot of the standard deviation in the rate constant as a function of the number of cycles. Note the expected power law (square root) behavior.

methods converge to the rate constants given in Table II. The standard deviation scales roughly with a square root of the number of cycles, as expected. The PBCBMC performs less efficiently, possibly because the guiding field is not optimized. While this comparison is important for the performance of the new methods, we stress that this work provides a proof of principle, not the most efficient implementation.

2. Non-conservative force field

For the non-conservative force field (see Appendix C), we only consider the acceptance probabilities, transition rate constant, and average path lengths. As in the previous test case, the number of cycles is set to 5×10^7, and each simulation is repeated 16 times

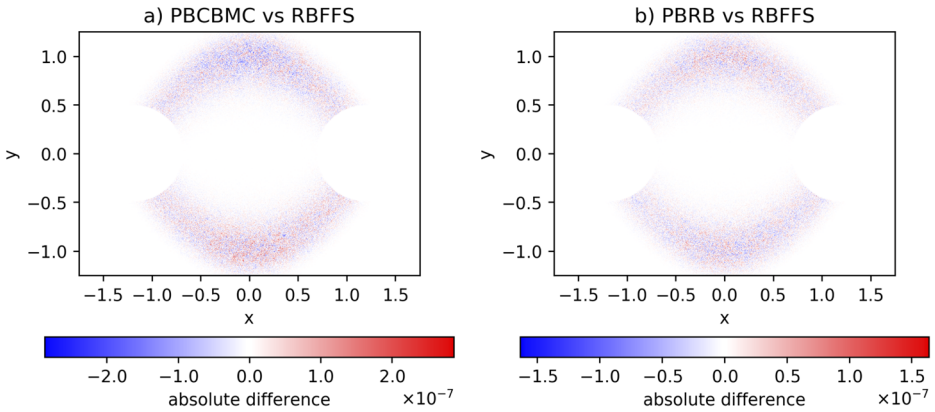

FIG. 6. Absolute difference between the path densities of the PBCBMC and RBFFS methods [panel (a)] and the absolute difference between the path densities of the PBRB and RBFFS methods [panel (b)]. Blue colors indicate that the RBFFS method has a higher value, whereas red indicates that the other method has a higher value.

Published under license by AIP Publishing

TABLE III. Acceptance probabilities for the three methods in the non-equilibrium test case. The second column contains the average fraction of trial paths that are successful and reach region *B*, and the third column contains the acceptance probability of trial paths, conditional on them reaching *B*. The fourth column shows the average number of accepted paths per hour of simulation time.

Method	p_{succ}	p_{cond}	$N_{\text{acc}}/T_{\text{sim}}$
PBCBMC	0.026 49 ± 0.000 02	0.383 50 ± 0.000 49	11 294/h
PBRB	0.215 76 ± 0.000 11	0.439 68 ± 0.000 18	22 172/h
RBFFS	0.095 36 ± 0.000 03	0.643 47 ± 0.000 22	59 556/h

in order to obtain an average and standard deviation. A typical simulation required 45 h, 214 h, and 52 h of simulation time for the PBCBMC, PBRB, and RBFFS methods, respectively. The success probabilities, conditional acceptance probabilities, and average number of accepted paths per hour of simulation time are listed in Table III.

For this system, the success probability is the highest in the PBRB method. The PBCBMC method again has a much lower success probability than the other two methods, despite increasing the bias strength. The RBFFS method still has the highest number of accepted paths per hour of simulation time because of its relatively high success probability and low simulation time.

The resulting transition rate constant and the average path lengths are listed in Table IV. Again, the results for all three methods are equal within error bar, which verifies that the new methods are also applicable to this non-equilibrium system. As in the equilibrium test case, the accuracy is highest in the PBRB method and lowest in the PBCBMC method, although the differences in accuracy are small. Compared to the equilibrium test case, the transition rate constant is slightly lower in the current system, even though the current system includes a stirring force that pushes the particle over the barrier. This is caused by the tenfold lower temperature in the current system compared to the equilibrium case. The lower temperature also causes more diffusive dynamics and, therefore, longer (average) path lengths.

B. Z-potential

For this test case, the PES and the boundaries of the stable regions are illustrated in Fig. 8 (see Appendix C for details). Each method is run for 10^8 cycles, and each simulation is repeated 16 times. A typical simulation takes 23 h, 30 h, and 68 h of computing time for the PBCBMC, PBRB, and RBFFS methods, respectively. The acceptance probabilities for the three methods are listed

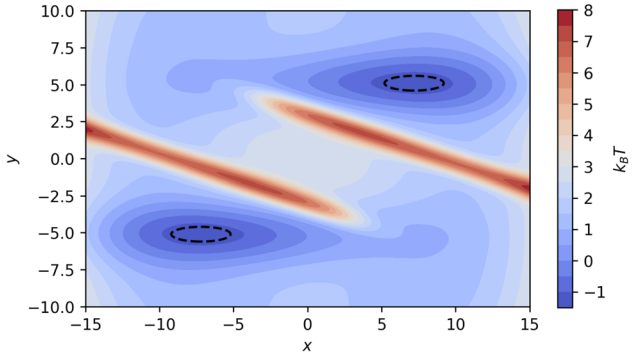

FIG. 8. Illustration of the Z-potential. The boundaries of the stable regions *A* and *B* are indicated with black dashed lines.

in Table V. The success probabilities for the Z-potential are much lower than in the previous two test cases. This is because the crossing probability for this system is much lower (this will be discussed later), which means that it is more difficult to generate successful trajectories. The success probability in the RBFFS method is substantially higher than for the path based methods. This may seem counter-intuitive (as the path based methods were developed to be free of a reaction coordinate), but it is important to note that the interfaces in the RBFFS method were optimized by Rogal *et al.*[28] The interfaces in the PBCBMC and PBRB methods were not optimized in any way, although the placement of the (time based) interfaces in the PBCBMC method was based on the information about the path lengths reported in Ref. 28. Furthermore, the simulations of the RBFFS method required approximately 2–3 times as much simulation time, compared to the PBCBMC and PBRB methods. This clearly illustrates the advantage of the

TABLE V. Acceptance probabilities for the three methods for the Z-potential test case. The second column contains the average fraction of trial paths that are successful and reach region *B*, and the third column contains the acceptance probability of trial paths, conditional on them reaching *B*. The fourth column shows the average number of accepted paths per hour of simulation time.

Method	$p_{\text{succ}}/10^{-4}$	p_{cond}	$N_{\text{acc}}/T_{\text{sim}}$
PBCBMC	0.1084 ± 0.0075	0.165 20 ± 0.0174	8/h
PBRB	1.4986 ± 0.0555	0.231 04 ± 0.0054	113/h
RBFFS	5.3869 ± 0.0161	0.075 72 ± 0.0050	60/h

TABLE IV. Results for the non-equilibrium toy model. The second column contains the transition rate constants, and the third column contains the average path length.

Method	$k_{AB}/10^{-5}$	$\langle L\rangle/10^3$
PBCBMC	1.454 63 ± 0.001 54	1.021 23 ± 0.000 36
PBRB	1.454 80 ± 0.001 12	1.021 49 ± 0.000 17
RBFFS	1.454 53 ± 0.001 21	1.021 31 ± 0.000 26

TABLE VI. Results for the Z-potential. The second column contains the transition rate constants, and the third column contains the average path length in integration steps.

Method	$k_{AB}/10^{-9}$	$\langle L\rangle/10^2$
PBCBMC	1.455 23 ± 0.187 36	9.237 54 ± 0.536 51
PBRB	1.514 17 ± 0.037 63	9.266 00 ± 0.257 18
RBFFS	1.525 68 ± 0.089 49	9.373 32 ± 0.304 54

Published under license by AIP Publishing

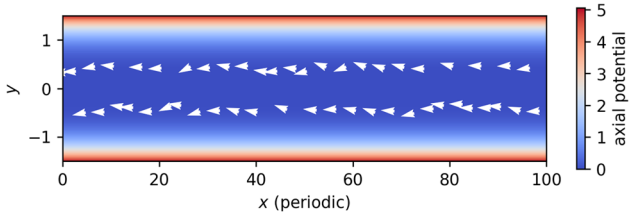

FIG. 9. Illustration of $N = 50$ self-propelled particles on a two-dimensional domain. The particles are denoted with arrows to indicate their orientation.

TABLE VII. Results for the system with self-propelled particles. The second column contains the transition rate constants, and the third column contains the average path length.

Method	$k_{AB}/10^{-6}$	$\langle L \rangle / 10^3$
PBCBMC	8.889 05 ± 2.428 19	1.277 26 ± 0.131 99
PBRB	8.273 70 ± 1.344 33	1.365 18 ± 0.079 46

path based methods that no knowledge of the reaction coordinate is required.

The resulting transition rate constants and average path lengths are listed in Table VI. The results are, within error bar, equal for all three methods, and the transition rate constant also matches the result found by Rogal *et al.*[28] The accuracy of the methods follows the same pattern as reported in Sec. III A: the PBRB method has the smallest error bar, followed by RBFFS and PBCBMC, respectively.

C. Active model

The systems described above consisted of only one particle in an external potential. In this section, we consider an active system consisting of 50 self-propelled particles confined to a two-dimensional domain. The particles' dynamics are given by an extension of the Vicsek model[32] (see Appendix C for details). A snapshot of the model is shown in Fig. 9. Note that the particles form two rows in the confined space, due to excluded volume, and align themselves in either the left or right direction. We investigate the transition from left to right alignment by performing 10^6 cycles using PBCBMC and PBRB and again repeat each simulation 16 times. We did not apply FFS since we have no indication of an order parameter for this transition. A typical simulation takes 39 h and 46 h for the PBCBMC and PBRB methods, respectively. The resulting rate constant and average path lengths are listed in Table VII.

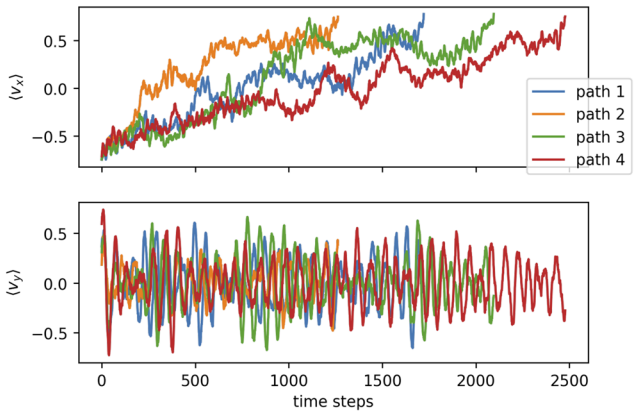

FIG. 10. Time evolution of the x and y components of the average particle velocity for four transition paths obtained in the PBRB method.

As in Secs. III A and III B, the results for the two methods are equal within error bars, and the PBRB method has lower error bars compared to the PBCBMC method even though the simulation times are in the same order. The error bars listed here are relatively large compared to the errors in Secs. III A and III B, which is due to the smaller amount of cycles used for the current system.

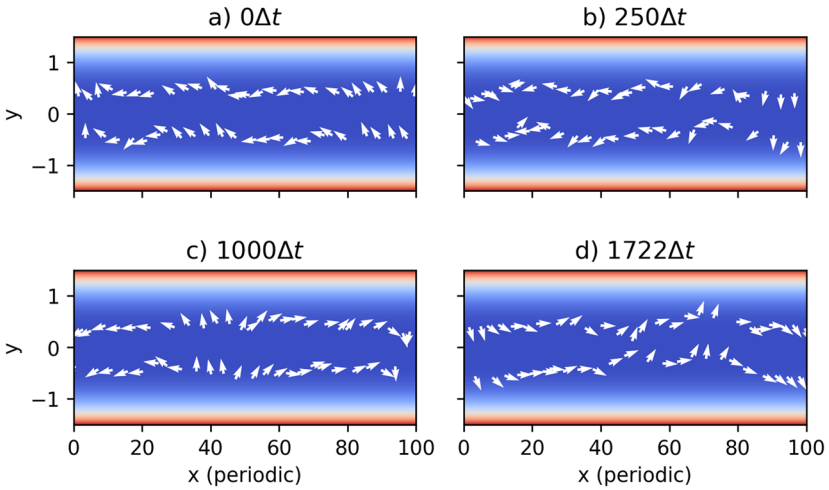

FIG. 11. Four snapshots obtained from a transition path. Panels (a) and (d) contain the first and last points of the path, whereas panels (c) and (d) contain intermediate points in time. Particles are illustrated with arrows that indicate their orientation.

Published under license by AIP Publishing

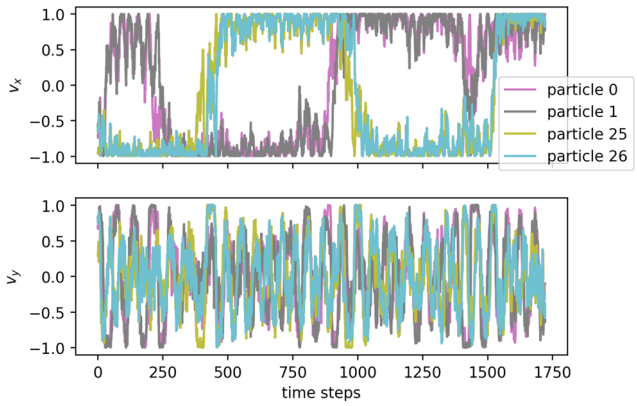

FIG. 12. Time evolution of the x and y components of four individual particles during a transition. Particles 0 and 1 (pink and gray) and particles 25 and 26 (green and blue) are local neighbors.

To get insight into the kinetics of a transition, the x and y components of the average particle velocity are plotted in Fig. 10 for four transition paths, selected from a single PBRB simulation. By the definition of the stable states, the x component of the average velocity (top panel) starts at -0.75 and ends at 0.75 for each path. The x component increases non-monotonically toward region B, with many small and a few larger fluctuations around the diagonal trend line. The fluctuations in the y component (bottom panel) are also clearly visible, where $\langle v_y \rangle$ oscillates around a zero mean value, as expected.

To get a better insight into the behavior of the individual particles, a snapshot of one of the trajectories is shown in Fig. 11 for four points in time. As in Fig. 9, the particles are illustrated with arrows that indicate their orientation. The trajectory from which the snapshots are obtained is the trajectory with a path length of 1722 (blue line) that is labeled with "path 1" in Fig. 10. The snapshots in Fig. 11 show relatively chaotic behavior during a transition from A to B. In the first and last snapshots [panels (a) and (d), respectively], the particles all point in either the negative or positive x direction. During the intermediate snapshots [panels (b) and (c)], the orientation of the particles varies, with some particles pointing in the positive and some in the negative x direction. Due to the alignment force between particles in a local neighborhood, there are (local) groups of particles all pointing into the same direction.

To further illustrate the local alignment of particles, the velocity components of four particles are plotted as a function of time in Fig. 12. The particles are selected by taking two neighboring particles out of two different groups. Clearly, each pair of neighboring particles follows the same pattern. The top panel of Fig. 12 shows that the groups of particles switch multiple times from positive to negative x direction and only align at the start and the end of the path. The bottom panel shows the fluctuations of v_y around a mean of 0, also observed for the mean $\langle v_y \rangle$ in Fig. 10. All four particles oscillate with approximately the same frequency.

The previous results indicate that a transition occurs by breaking the collective movement of the particles and that the particles form groups that switch to the opposite x direction. To further illustrate this, the path density as a function of the *standard deviation* of the particle velocity and the *average* particle velocity in the x direction is plotted in Fig. 13 for both the PBCBMC and the PBRB method, showing the same semicircular path density. Because the acceptance ratio for the PBRB method is higher, the density for the PBRB method [panel (b)] is better distributed than the density for the PBCBMC method [panel (a)]. The semicircular path density indicates that the collective movement of the particles is indeed broken during a transition from A to B. In a collective movement, the particles have the same orientation, and the standard deviation of v_x is small, with 0 being the minimum. For instance, this is what happens near the stable regions A and B, that is, where $\langle v_x \rangle \approx \pm 0.75$. If a transition between A and B would be a collective process, where all particles synchronously change their velocity to the opposite direction, then the standard deviation would remain close to zero during a transition. However, the arc-shape in the path densities of Fig. 13 indicates that the transition occurs via disorder in the particles' orientation. When the average v_x is zero, the standard deviation peaks at approximately 0.8, meaning that approximately half of the particles point in either the positive or negative x direction.

Although the collective movement of the particles is broken during the transition, the particle orientation is not *completely* disordered. As illustrated in Figs 11 and 12, there are still (local) groups of particles that point in the same direction. For a future work, it would be interesting to further quantify the correlation between particles during transitions, by considering, for instance, the correlation length.

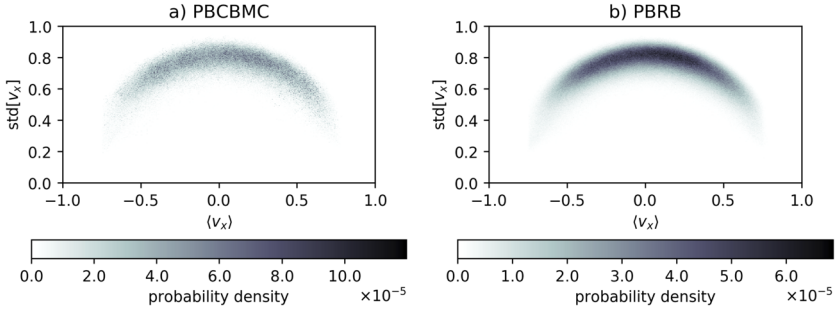

FIG. 13. Path densities of trajectories sampled with the PBCBMC and PBRB methods. Here, the coordinates of the particles are projected onto two coordinates: the average particle velocity in the x direction (horizontal-axis) and the standard deviation of the particle velocities in the x direction (vertical axis).

Published under license by AIP Publishing

IV. CONCLUSIONS

In this work, we have presented two novel algorithms with the aim to provide a transition path sampling technique suitable for non-equilibrium dynamics that is not dependent on predefined collective variables to define a set of interfaces. The methods are based on the configurational bias Monte Carlo technique introduced in the original TPS paper, as well as on the Rosenbluth based FFS method. Yet, both algorithms can be classified as TPS algorithms since they aim at importance sampling of trajectory space. At the heart of both methods is the concept to use the current path to define a progress variable that guides trajectories over the barrier. As this progress variable adapts itself during the sampling, the methods are independent from a predefined reaction coordinate. In this work, we used the path based variable of Branduardi *et al.*[26] for measuring progress, but other effective path based variables could be developed in the future.

While our methodology has similarities to the original steered TPS approach (STePS),[14] we stress that it is slightly different in spirit from this and other existing algorithms for sampling non-equilibrium paths, such as direct FFS, in the sense that there is no predefined collective variable, or constraint function, but rather a continuously adapting pathway which guides new trial paths. Moreover, in many of these methods, there is no Metropolis acceptance rule governing the acceptance of a trial path with respect to the old current path, something that is crucial for our approach, since the new path needs to relax into and between "channels" in path space. However, we note that Gingrich and Geissler have adapted the STePS approach to generate paths that stay in the proximity of reference trajectories, in the context of sampling long, correlated trajectories.[15] This approach is similar to our algorithms, although it differs in details, such as how proximity is measured and how the bias is corrected. Another exception is the RBFFS method, which allows Metropolis sampling, and is what we use to compare with. Note, however, that even RBFFS needs to create a new path along a predefined order parameter, which might not be available.

We tested both the PBCBMC and PBRB algorithms on equilibrium toy models, as well as on non-equilibrium dynamics. Both methods give correct results, and while PBRB seems more efficient overall, optimizing the sampling settings for each method might change this conclusion. For the active system, we find that the transitions occur via states that are locally ordered but globally disordered. This interesting finding can be further studied in future research.

While we presented results on the convergence of the methods as a function of the cycle number, we did not investigate the scaling of the methods as a function of other parameters, such as the number of interfaces or the length of trajectories. As discussed thoroughly in Ref. 15, algorithms that grow trajectories by accumulating weights over time in general suffer from poor convergence when long transition paths are needed as the weights are asymptotically exponentially small for long trajectories. While this is probably also the case for the presented algorithms, it is also the case for, e.g., RBFFS.

In addition to the TPS related methods, there is also literature on large deviation theory and the use of cloning algorithm to sample non-equilibrium path ensembles.[19–21] While this is outside the scope of this work, it would be interesting to investigate in the future how these algorithms relate to the ones introduced here.

We expect that our novel algorithms will be useful for non-equilibrium systems for which there is not (yet) sufficient insight to define FFS interfaces. Examples are active matter undergoing large (dynamical) transformations, biomolecular driven systems, and other systems without microscopic reversibility.

ACKNOWLEDGMENTS

The authors thank Pieter Rein ten Wolde for discussions and a critical reading of the manuscript.

APPENDIX A: THEORETICAL BACKGROUND

1. Transition path sampling

A path is defined as a discrete sequence of $L + 1$ phase space points, or frames, $\mathbf{x} = \{x_0, x_1, x_2, \ldots, x_L\}$. Frames are separated by a small time increment Δt so that the total duration of the path equals $\tau = L\Delta t$. Each frame x_i completely defines the system at a given point in time and typically consists of the positional coordinates and momenta of all particles in the system.[3,33] The Path Ensemble (PE) is defined by the probability $\mathscr{P}[\mathbf{x}]$ of such paths. Under Markovian dynamics, the probability of making a transition between two subsequent frames, $p(x_i \to x_{i+1})$, is independent of any prior state.[34] In this case, the statistical weight of a path can be written as a product of short time step transition probabilities,[3,9]

$$\mathscr{P}[\mathbf{x}] \equiv \rho(x_0) \prod_{i=0}^{L-1} p(x_i \to x_{i+1}), \tag{A1}$$

where $\rho(x_0)$ is the distribution of initial conditions. Transition Path Sampling (TPS) is only concerned with paths that connect two stable states and thus constrains the path ensemble to trajectories that start in one region of phase space, A, and end in another region, B. Most modern path sampling algorithms employ a flexible path length scheme and constrain the path ensemble even further by considering only paths for which only the initial point lies in A and only the final point lies in B. The Transition Path Ensemble (TPE) is defined as the subset of paths in the PE that match these restrictions. To write down a probability for paths in the TPE, it is convenient to define the indicator function,

$$h(\mathbf{x}) = \begin{cases} 1 & \text{if } x_0 \in A, x_L \in B \text{ and } x_i \notin A \cup B \\ & \text{for } 0 < i < L \\ 0, & \text{otherwise.} \end{cases} \tag{A2}$$

The statistical weight for a path in the TPE is given by

$$\mathscr{P}_{AB}[\mathbf{x}] \equiv h(\mathbf{x})\mathscr{P}[\mathbf{x}]/\mathscr{Z}_{AB}, \tag{A3}$$

$$\mathscr{Z}_{AB} \equiv \int_{\mathbf{x}\in \mathrm{PE}} h(\mathbf{x})\mathscr{P}[\mathbf{x}]D\mathbf{x}, \tag{A4}$$

where $\mathscr{Z}_{AB}$ is a normalization factor also known as the partition function for the TPE.[35] The goal of TPS simulations is to sample trajectories with a frequency proportional to the distribution in Eq. (A3). This can be achieved, for instance, with a Monte Carlo procedure. Such a procedure performs a random walk through path

Published under license by AIP Publishing

space by generating a trial trajectory from an existing pathway and accepting or rejecting the trial based on a suitable acceptance criterion. In the shooting move,[36] trial trajectories are generated by selecting a random point from an existing pathway, adding a perturbation (usually in the momenta), and "shooting" forward and backwards by integrating the equations of motion. The exact acceptance probability for the trial trajectory depends on the details of the shooting move but should always be proportional to the indicator function of Eq. (A2).[33] In principle, the acceptance ratio can be tuned by adjusting the perturbation of the shooting point.

2. Transition interface sampling

Transition Interface Sampling (TIS) is an extension to TPS, specifically developed to compute the transition rate constant with greater efficiency.[7] In TIS, phase space is partitioned into a set of $n+1$ non-intersecting interfaces $\lambda_0 < \lambda_1 < \cdots < \lambda_n$, where $\lambda_0 \equiv \lambda_A$ and $\lambda_n \equiv \lambda_B$ denote the boundaries of the stable regions A and B, respectively. Each intermediate interface λ_i is then sampled using a path sampling scheme, where not only reactive pathways but also trajectories that come from A, cross the interface λ_i, and subsequently return to A are considered. The sampling of each interface results in an estimate for the conditional crossing probability $P(\lambda_{i+1}|\lambda_i)$, which measures the probability that a trajectory coming from A crosses λ_{i+1} before returning to A, conditional on crossing λ_i. Combining the conditional crossing probabilities from each interface yields a total crossing probability $P(\lambda_n|\lambda_1) = \prod_{i=1}^{n-1} P(\lambda_{i+1}|\lambda_i)$ that measures the probability of reaching B conditional on crossing the first interface λ_1. The rate constant k_{AB} can then be found by multiplying the total crossing probability with the effective positive flux $\Phi(\lambda_1|\lambda_0)$ for trajectories coming from A,[7]

$$k_{AB} = \Phi(\lambda_1|\lambda_0) \prod_{i=1}^{n-1} P(\lambda_{i+1}|\lambda_i). \tag{A5}$$

The flux factor $\Phi(\lambda_1|\lambda_0)$ considers only crossings in the forward direction toward region B (positive) and does not count recrossings before coming back to A (effective). The flux can be computed from an MD simulation in region A by counting the number of effective crossings with λ_1.[7]

3. Forward flux sampling

Forward Flux Sampling (FFS)[11,12] is built on the same framework as TIS and uses the same partitioning of phase space into interfaces. However, instead of an importance sampling, in FFS, trajectories are generated only in a forward time direction by ratcheting along the interfaces toward region B.[11] Shortly after the introduction of the basic dFFS scheme, two other variants known as the "Branched Growth" and "Rosenbluth" methods were developed in analogy to efficient schemes for sampling polymers.[12] Here, we consider only the Rosenbluth based FFS scheme, from here on referred to as RBFFS, because this variant forms the basis for the novel algorithms that will be introduced in Sec. II. In RBFFS, trial paths are generated in multiple stages. The first stage consists of an MD simulation in stable region A, which is halted once the system leaves A and crosses the first interface λ_1. In the next stage, the resulting system configuration is used to initiate K_1 trial path segments that continue until crossing the next interface λ_2 or until returning to region A. One of the "successful" trials that crossed λ_2 is then selected at random, and its endpoint is used to initiate K_2 trial path segments that continue until crossing λ_3 or until entering region A. This procedure is repeated for the remaining interfaces until reaching B or until there were no successful trials at an intermediate interface. In either case, the resulting trial path is assigned a weight W known as the Rosenbluth factor that is given by

$$W = \prod_{i=1}^{n-1} \frac{N_i}{K_i}, \tag{A6}$$

where N_i denotes the number of successful trials that were generated from the interface λ_i. Similar to TPS, a Metropolis acceptance/rejection criterion can be applied to ensure that paths are sampled with the correct weight. The acceptance probability is given by[12]

$$P_{\mathrm{acc}}[\mathbf{x}^{(\mathrm{o})} \to \mathbf{x}^{(\mathrm{n})}] = \min\left\{1, \frac{W^{(\mathrm{n})}}{W^{(\mathrm{o})}}\right\}, \tag{A7}$$

where the superscripts (o) and (n) denote the (old) existing path and the (new) trial path, respectively. As this is effectively an importance sampling algorithm, this form of FFS could also be classified as being a classical TPS algorithm.[1] Still, it is hard to adjust the acceptance ratio in RBFFS, although the spacing of interfaces gives some limited level of control.

APPENDIX B: JUSTIFICATION OF THE ALGORITHMS

Here, the correctness of both the PBCBMC and the PBRB methods is demonstrated in a single, general framework. For this, we need to re-index the interfaces of the PBRB method. In the PBRB method, it is possible that the first (or last) interfaces are ignored if there is overlap with the intermediate interfaces and λ_1 (or λ_n). For consistency with the PBCBMC method, we relabel the interfaces in such a way that the jth interface that is sampled has index j.

To prove that the acceptance probability of the PBCBMC and PBRB methods leads to correct sampling of paths in the TPE, it is important to note that the probability of generating a path in these methods depends on the generation of the entire jets. Following Allen *et al.*[12] we define a decorated path $\mathbf{x}_{\mathrm{dec}} = \cup_{j,k} \mathbf{s}_{j,k}$ as the bare transition path $\mathbf{x}$, together with the set of unselected path segments from each jet. This includes the path segment $\mathbf{s}_0$ from the flux calculation that connects λ_0 to λ_1. The idea of the proof is to derive the weights with which a decorated path is sampled, under the condition of detailed balance. By subsequently integrating out all possible ways to decorate a transition path, it follows that the bare transition paths are sampled with the correct weights of the TPE. The proof follows a similar (but more general) approach as the proof for the RBFFS method derived by Allen *et al.* in Ref. 12.

Each segment $\mathbf{s}_{j,k}$ has a generation probability equal to the product of the short time step probabilities,

$$p_{\mathrm{gen}}(\mathbf{s}_{j,k}) = \prod_{l=0}^{L_{j,k}} p\left(x_{j,k,l} \to x_{j,k,l+1}\right) / \mathscr{Z}_j, \tag{B1}$$

$$\mathscr{Z}_j \equiv \int d\mathbf{s}_j p_{\mathrm{gen}}(\mathbf{s}_j), \tag{B2}$$

where $\mathscr{Z}_j$ is a normalization constant that takes into account all possible ways to generate a segment in the jth jet.

Published under license by AIP Publishing

After generating K_j segments, one successful segment is selected with a probability $p_{\mathrm{sel}}^{(\mathrm{n})}(\mathbf{s}_{j,*}^{(\mathrm{n})})$, which is given by the relative weight $w_{j,*}^{(\mathrm{n})}/W_j^{(\mathrm{n})}$. The superscript "(n)" used in $p_{\mathrm{sel}}^{(\mathrm{n})}$ follows the same convention as used in the weights. The superscript "(n)" thus indicates that the selection probability holds for the path labeled with "n", which implicitly depends on the path labeled with "o". As in Eq. (5), a subtlety arises when the last jet produced no successful trials, in which case $W_J = 0$ and $p_{\mathrm{sel}}(\mathbf{s}_{J,*})$ is not defined. Since the acceptance probability for an unsuccessful path is zero, we can simply set $p_{\mathrm{sel}}(\mathbf{s}_{J,*}) = 1/K_J$ if $w_{J,k} = 0$ for all $1 \le k \le K_J$.

If there was at least one successful trial and the selected segment did not enter region B, the next jet is generated. The generation probability for a decorated path is thus given by

$$P_{\mathrm{gen}}[\mathbf{x}_{\mathrm{dec}}^{(\mathrm{o})} \to \mathbf{x}_{\mathrm{dec}}^{(\mathrm{n})}] = \rho(x_0^{(\mathrm{n})}) \prod_{j=0}^{J^{(\mathrm{n})}} \Big[p_{\mathrm{gen}}(\mathbf{s}_{j,*}^{(\mathrm{n})}) p_{\mathrm{sel}}^{(\mathrm{n})}(\mathbf{s}_{j,*}^{(\mathrm{n})}) \times h_j(\mathbf{s}_{j,*}^{(\mathrm{n})}; J^{(\mathrm{n})}) \prod_{\substack{1 \le k \le K_j, \\ k \ne *}} p_{\mathrm{gen}}(\mathbf{s}_{j,k}^{(\mathrm{n})}) \Big], \tag{B3}$$

where the last product runs over all unselected trials in the jth jet. Here, we used that $K_0 = 1$ for the initial path segment from the flux calculation that leads up to λ_1. The indicator function h_j is determined by the termination conditions and ensures that the bare transition path starts in A, ends in A or B, and lies in the region between A and B for all intermediate steps. The indicator function is defined by

$$h_j(\mathbf{s}_{j,k}; J) \equiv \begin{cases} 1 & \text{if } (j = 0 \text{ and } \mathbf{s}_{j,k} \cap A = x_{j,k,0}) \text{ or} \\ & (0 < j < J \text{ and } \mathbf{s}_{j,k} \cap (A \cup B) = \emptyset) \text{ or} \\ & (j = J \text{ and } \mathbf{s}_{j,k} \cap (A \cup B) = x_{j,k,L_{j,k}}) \\ 0, & \text{otherwise,} \end{cases} \tag{B4}$$

where the intersection $\mathbf{s}_{j,k} \cap (A \cup B)$ is defined as the subset of states in the segment $\mathbf{s}_{j,k}$ that belong to region A or B. To simplify notation, we define $\tilde{h}(\mathbf{x}) \equiv \prod_{j=0}^{J} h_j(\mathbf{s}_{j,*}; J)$. Note that $\tilde{h}$ is closely related to the indicator function h for the TPE defined in Eq. (A2). The two differ because the former also accepts paths that end in A.

Using the definition of $\tilde{h}$ and combining the factor $p_{\mathrm{gen}}(\mathbf{s}_{j,*}^{(\mathrm{n})})$ with the product over all unselected trials yields

$$P_{\mathrm{gen}}[\mathbf{x}_{\mathrm{dec}}^{(\mathrm{o})} \to \mathbf{x}_{\mathrm{dec}}^{(\mathrm{n})}] = \rho(x_0^{(\mathrm{n})}) \tilde{h}(\mathbf{x}^{(\mathrm{n})}) \prod_{j=0}^{J^{(\mathrm{n})}} \Big[p_{\mathrm{sel}}^{(\mathrm{n})}(\mathbf{s}_{j,*}^{(\mathrm{n})}) \prod_{k=1}^{K_j} p_{\mathrm{gen}}(\mathbf{s}_{j,k}^{(\mathrm{n})}) \Big]. \tag{B5}$$

By imposing the condition of detailed balance in the space of decorated paths, we have

$$\frac{\tilde{\mathscr{P}}_{\mathrm{dec}}[\mathbf{x}_{\mathrm{dec}}^{(\mathrm{n})}]}{\tilde{\mathscr{P}}_{\mathrm{dec}}[\mathbf{x}_{\mathrm{dec}}^{(\mathrm{o})}]} = \frac{P_{\mathrm{gen}}[\mathbf{x}_{\mathrm{dec}}^{(\mathrm{o})} \to \mathbf{x}_{\mathrm{dec}}^{(\mathrm{n})}]}{P_{\mathrm{gen}}[\mathbf{x}_{\mathrm{dec}}^{(\mathrm{n})} \to \mathbf{x}_{\mathrm{dec}}^{(\mathrm{o})}]} \times \frac{P_{\mathrm{acc}}[\mathbf{x}_{\mathrm{dec}}^{(\mathrm{o})} \to \mathbf{x}_{\mathrm{dec}}^{(\mathrm{n})}]}{P_{\mathrm{acc}}[\mathbf{x}_{\mathrm{dec}}^{(\mathrm{n})} \to \mathbf{x}_{\mathrm{dec}}^{(\mathrm{o})}]}, \tag{B6}$$

where $\tilde{\mathscr{P}}_{\mathrm{dec}}[\mathbf{x}_{\mathrm{dec}}^{(\mathrm{n})}]$ is the weight with which a decorated path is sampled. Applying the acceptance probability of Eq. (5) and substituting Eq. (B5) in the equation above, it is clear that many terms cancel in the products $P_{\mathrm{gen}}[\mathbf{x}_{\mathrm{dec}}^{(\mathrm{o})} \to \mathbf{x}_{\mathrm{dec}}^{(\mathrm{n})}] \times P_{\mathrm{acc}}[\mathbf{x}_{\mathrm{dec}}^{(\mathrm{o})} \to \mathbf{x}_{\mathrm{dec}}^{(\mathrm{n})}]$ and $P_{\mathrm{gen}}[\mathbf{x}_{\mathrm{dec}}^{(\mathrm{n})} \to \mathbf{x}_{\mathrm{dec}}^{(\mathrm{o})}] \times P_{\mathrm{acc}}[\mathbf{x}_{\mathrm{dec}}^{(\mathrm{n})} \to \mathbf{x}_{\mathrm{dec}}^{(\mathrm{o})}]$. For these products, there are two possible outcomes. Either the decorated path ends in B and all of the relative weights $w_{j,*}/W_j$ cancel or the decorated path ends in A and the product equals zero. It is, therefore, useful to define a new indicator function h that is equal to $\tilde{h}$ if a path ends in B and 0 if the path ends in A. Note that the indicator function h is now equivalent to Eq. (A2). Using this convention and substituting Eqs. (5) and (B5) into Eq. (B6) yields

$$\frac{\tilde{\mathscr{P}}_{\mathrm{dec}}[\mathbf{x}_{\mathrm{dec}}^{(\mathrm{n})}]}{\tilde{\mathscr{P}}_{\mathrm{dec}}[\mathbf{x}_{\mathrm{dec}}^{(\mathrm{o})}]} = \frac{\rho(x_0^{(\mathrm{n})}) h(\mathbf{x}^{(\mathrm{n})}) \prod_{j=1}^{J^{(\mathrm{n})}} \Big[K_j \prod_{k=1}^{k_j} p_{\mathrm{gen}}(\mathbf{s}_{j,k}^{(\mathrm{n})}) \Big]}{\rho(x_0^{(\mathrm{o})}) h(\mathbf{x}^{(\mathrm{o})}) \prod_{j'=1}^{J^{(\mathrm{o})}} \Big[K_{j'} \prod_{k=1}^{k_{j'}} p_{\mathrm{gen}}(\mathbf{s}_{j',k}^{(\mathrm{o})}) \Big]}. \tag{B7}$$

A particular decorated path $\mathbf{x}_{\mathrm{dec}}$ is thus sampled with weight,

$$\tilde{\mathscr{P}}[\mathbf{x}_{\mathrm{dec}}] = \rho(x_0) h(\mathbf{x}; \tau) \prod_{j=1}^{J} \Big[K_j \prod_{k=1}^{k_j} p_{\mathrm{gen}}(\mathbf{s}_{j,k}) \Big]. \tag{B8}$$

We are, however, interested in the weight of an *undecorated* transition path. For the RBFFS method, this weight is simply equal to the integration over $\tilde{\mathscr{P}}[\mathbf{x}_{\mathrm{dec}}]$, taken over all possible decorations of $\mathbf{x}$.[12] In this case, however, the number of jets J is variable, and the presence of bare transition paths is no longer balanced in the space of decorated paths. For each unselected trial in some jet, the bare transition path gains an additional degree of freedom, so its presence is duplicated in the space of decorated paths. By adding a decoration of $k - 1$ unselected trials, the weight of a bare transition path in the space of decorated paths thus increases by a factor k. Hence, we need to weight the integration by the number of trials in each jet to take into account the additional degrees of freedom.

By the definition of p_{gen} given in Eq. (B1), the integration over all possible ways to generate a particular segment $\mathbf{s}_{j,k'}$ in the jth jet equals unity. That is,

$$\int d\mathbf{s}_{j,k'} p_{\mathrm{gen}}(\mathbf{s}_{j,k'}) = 1. \tag{B9}$$

By repeating the integration for all unselected trials in a jet, and multiplying the results, it thus follows that

$$\prod_{\substack{1 \le k \le K_j, \\ k \ne *}} \int d\mathbf{s}_{j,k} p_{\mathrm{gen}}(\mathbf{s}_{j,k}) = 1. \tag{B10}$$

Since the segments are independent of each other, the integration and the product can be interchanged. Denoting with $\int_{\mathrm{unsel}}^{(j)} \equiv \int d\mathbf{s}_{j,k'} \int d\mathbf{s}_{j,k''} \int \ldots$ the repeated integration over all unselected trials in the jth jet, it follows that

$$\int_{\mathrm{unsel}}^{(j)} \prod_{\substack{1 \le k \le K_j, \\ k \ne *}} p_{\mathrm{gen}}(\mathbf{s}_{j,k}) = 1. \tag{B11}$$

Hence, taking the weighted integration $\frac{1}{K_j} \int_{\mathrm{unsel}}^{(j)}$ over Eq. (B8) for each jet $j = 0, \ldots, J^{(\mathrm{n})}$, it follows that the weight $\tilde{\mathscr{P}}[\mathbf{x}]$ of a bare

Published under license by AIP Publishing

transition path equals

$$\tilde{\mathscr{P}}[\mathbf{x}] = \frac{1}{K_0}\int_{\text{unsel}}^{(0)}\frac{1}{K_1}\int_{\text{unsel}}^{(1)}\cdots\frac{1}{K_J}\int_{\text{unsel}}^{(J)}\left[\tilde{\mathscr{P}}[\mathbf{x}_{\text{dec}}]\right] \tag{B12}$$

$$= \rho(x_0)h(\mathbf{x})\prod_{j=0}^{J}\left[\int_{\text{unsel}}^{(j)}\prod_{k=1}^{K_j}p_{\text{gen}}(\mathbf{s}_{j,k})\right] \tag{B13}$$

$$= \rho(x_0)h(\mathbf{x})\prod_{j=0}^{J}p_{\text{gen}}(\mathbf{s}_{j,*}). \tag{B14}$$

Substituting Eq. (B1) for the generation probabilities of the selected segments proves that the PBCBMC and PBRB methods indeed sample paths proportional to the weights of the TPE defined in Eq. (A3).

APPENDIX C: SIMULATION DETAILS

1. Toy model

The potential used in the original TPS paper is given by[1]

$$V(x,y) = (4x^4 + 10x^2(y^2-1) + 3(y^4 - 2y^2 + 2))/3 \tag{C1}$$

and has two minima located in $\vec{r}_A \equiv (x = -\sqrt{5}/2, y = 0)$ and $\vec{r}_B \equiv (x = +\sqrt{5}/2, y = 0)$. (Note that in this section and Appendixes C 2 and C 3, we refer to x and y as simple Cartesian coordinates). We define the stable regions A and B as circular disks of radius $\sqrt{0.25}$, centered in $\vec{r}_A$ and $\vec{r}_B$, respectively. The PES and the boundaries of the stable regions are illustrated in Fig. 4. In the equilibrium test case, the force field is simply given by

$$\vec{F}_{\text{cons}}(x,y) \equiv -\vec{\nabla}V(x,y). \tag{C2}$$

For the non-equilibrium test case, we add a "stirring force" $\vec{F}_{\text{stir}}(x,y) = (-y,x)$ inspired by Sachs *et al.*[37] The stirring force pushes the particle radially and in anti-clockwise direction around the origin. The total force field $\vec{F}_{\text{nc}}$ is simply given by the sum of the stirring force and the conservative force. Thus,

$$\vec{F}_{\text{nc}}(x,y) = \vec{F}_{\text{cons}}(x,y) + \vec{F}_{\text{stir}}(x,y). \tag{C3}$$

We use the same parameter settings for both test cases, with the exception of the reciprocal temperature β and the bias strength b. The stirring force effectively lowers the free energy barrier for $y < 0$, which results in an increased transition rate constant. We, therefore, use a lower temperature in the non-equilibrium case. More specifically, we set $\beta = 8$ in the equilibrium system and $\beta = 80$ in the non-equilibrium system. The bias strength is set to $b = 1$ for the equilibrium system and increased to $b = 5$ for the non-equilibrium system.

The dynamics are given by the Langevin equation with friction $\gamma = 2.5$ and a time step of $\Delta t = 0.01$. The integration is performed using the BAOAB-integrator.[38] For consistency, we use the same amount of interfaces in all three methods. In the RBFFS method, the interfaces are defined as circles centered in $\vec{r}_A$ and with radii in $\{\sqrt{0.40}, \sqrt{0.75}, \sqrt{1.00}, \sqrt{1.50}\}$. For the PBRB method, the interfaces are defined as the isolines of the progress variable defined determined by the values {0.05, 0.15, 0.25, 0.35}. In the PBCBMC method, we shoot 4 jets where the path segments have a maximum length of $100\Delta t$ and a final jet where the path segments continue until reaching either A or B. To define the path variable of Eq. (1), we consider only the positional coordinates of the particle. The parameter α is set equal to $\alpha = 1/\Delta t$. In each method, we run 10 trials per jet.

2. Z-potential

The Z-potential is given by[28]

$$V(x,y) = \frac{x^4+y^4}{20\,480} - 3e^{-0.01(x+5)^2-0.2(y+5)^2} - 3e^{-0.01(x-5)^2-0.2(y-5)^2} + \frac{5e^{-0.2(x+3(y-3))^2}}{1+e^{-x-3}} + \frac{5e^{-0.2(x+3(y+3))^2}}{1+e^{x-3}} + 3e^{-0.01(x^2+y^2)} \tag{C4}$$

and has two minima located at $\vec{r}_A = (-7.2, -5.1)$ and $\vec{r}_B = (+7.2, +5.1)$. The stable regions A and B are defined as ellipsoid disks around $\vec{r}_A$ and $\vec{r}_B$, respectively. The disks are parameterized by $\{(x,y)|(x-x_m)^2/16 + (y-y_m)^2 < 0.25\}$, where (x_m, y_m) denotes either minima $\vec{r}_A$ or $\vec{r}_B$. The PES and the boundaries of the stable regions are illustrated in Fig. 8. The force field is simply given by the negative gradient of the potential, $\vec{F}(x,y) \equiv -\vec{\nabla}V(x,y)$, and is thus conservative. The dynamics are Langevin with friction $\gamma = 1.0$ and reciprocal temperature $\beta = 4.0$. The time step at the integrator level is set to $\Delta t = 0.05$, and only every 10th integration step is stored. The interfaces for the RBFFS method are defined in the same way as the linear interfaces used by Rogal *et al.*[28] These are parameterized by $y = -0.2x + \lambda$, where λ takes the values {−5.5, −5, −4.5, −4.25, −4, −3.75, −3, −1.5, −1, −0.5, 0.5, 0.75, 1, 1.5}. For the PBRB method, the interfaces are defined by the values {0.05, 0.1, 0.15, 0.25, 0.35, 0.5, 0.75} of the path progress variable along the old path. In the PBCBMC method, we shoot 15 jets of maximum path segment length $250\Delta t$ and a final jet that continues until reaching A or B. The bias parameter is set to $b = 5$. The parameter α for the progress variable is set to $\alpha = 1$, and we run 10 trials for each jet.

3. Active model

The systems described in Appendixes C 1 and C 2 consisted of only a single particle in an external potential. In this section, we consider a larger system that consists of multiple self-propelled particles, still in a two-dimensional domain. The system is modeled using an extension of the Vicsek model.[32] In the Vicsek model, particles move at a fixed absolute speed and align their direction of motion with that of the particles in a local neighborhood. The alignment of the particles is perturbed by adding some random noise. If the noise is relatively small and the density of particles is sufficiently large, the system shows a collective behavior in which the particles all move in the same spontaneously selected direction.[32]

Here, a number of particles N are placed on a two-dimensional domain with a periodic boundary condition in the x direction. In order to prevent particles from escaping toward infinity, an axial potential $V_{\text{ax}}(x, y) \equiv c_{\text{ax}}y^4$ ($c_{\text{ax}} > 0$) is added that pulls the particles toward the y axis. In addition, we add a purely repulsive force between the particles to avoid clustering of particles into groups. Due to the axial potential, the particles tend to have a relatively large velocity component in the x direction compared to the y direction. Due to the alignment and repulsive forces, there is a collective

Published under license by AIP Publishing

movement of the particles in either the positive or the negative x direction. A snapshot of the domain with $N = 50$ particles is shown in Fig. 9. Particles are illustrated with arrows that indicate their orientation.

To model the additional forces in the Vicsek model, we follow Chaté *et al.*, who built on earlier work by Grégoire, Chaté, and Tu and introduced an extended Vicsek model with repulsive (and attractive) forces.[39,40] The difference between their model and the model used here is the addition of the axial potential. Furthermore, we use an exponential cutoff for the definition of the particle neighborhood, rather than a hard cutoff.

The particles are labeled by index i and have a velocity $\vec{v}^{(i)}$ of fixed modulus $v_0 = |\vec{v}^{(i)}|$. At each time step, the velocities of all particles are updated according to Ref. 40,

$$\vec{v}^{(i)}(t + \Delta t) = v_0(\mathscr{R}_\eta \circ \Theta)(\vec{F}^{(i)}), \tag{C5}$$

where $\circ$ denotes the function decomposition, Θ is a normalization operator [that is, $\Theta(\vec{w}) = \vec{w}/|\vec{w}|$], and $\mathscr{R}_\eta$ is an operator that performs a random rotation uniformly distributed around the argument vector inside an arc of amplitude $2\pi\eta$. The total force $\vec{F}^{(i)}$ acting on particle i is a linear combination of the individual forces. Here, we weight each force equally, and the total force acting on particle i is thus given by

$$\vec{F}^{(i)} = \vec{F}^{(i)}_{\text{ax}} + \sum_{j\neq i} \vec{F}^{(i,j)}_{\text{rep}} + \sum_{j} \vec{F}^{(i,j)}_{\text{align}}. \tag{C6}$$

The axial force is simply given by the negative gradient of the axial potential,

$$\vec{F}_{\text{ax}}(x, y) = -\vec{\nabla} V_{\text{ax}}(x, y) = -(4c_{\text{ax}} y^3)\hat{y}, \tag{C7}$$

and, therefore, depends only on the position of particle i. For the repulsive force, we use the same expression as in the work of Chaté *et al.*[40] The repulsive force acting on particle i due to particle j depends on the distance $\vec{r}_{ij} = \vec{r}^{(j)} - \vec{r}^{(i)}$ from i to j and is given by

$$\vec{F}^{(i,j)}_{\text{rep}} = -\frac{\vec{r}_{ij}/|\vec{r}_{ij}|}{1 + e^{|\vec{r}_{ij}|/r_c - 2}}, \tag{C8}$$

where r_c is the typical repulsion range. In the original Vicsek model, the alignment force that particle j exerts on particle i is equal to $\vec{v}^{(j)}$ if particle j is within a certain range of particle i and equals 0 otherwise. Here, we use an exponential rather than a hard cutoff. We define

$$\vec{F}^{(i,j)}_{\text{align}} = 2\frac{\vec{v}^{(j)}/v_0}{1 + e^{|\vec{r}_{ij}|/r_c}}. \tag{C9}$$

The exponential decay for the alignment force is slightly faster than the decay for the repulsive interactions. Note that the total force acting on particle i also includes an alignment force due to particle i itself. Since $\vec{r}_{ii} = 0$, it follows that $\vec{F}^{(i,i)}_{\text{align}} = \vec{v}^{(i)}/v_0$. After updating the particle velocities according to Eq. (C5), the particle positions are updated by the following simple rule:[40]

$$\vec{r}^{(i)}(t + \Delta t) = \vec{r}^{(i)}(t) + \Delta t \cdot \vec{v}^{(i)}(t + \Delta t). \tag{C10}$$

We note that the above described model does not represent any physical process but was developed to demonstrate the novel algorithms in a more challenging system. The model is, therefore, unit-less. In the following, we set $\Delta t = 0.01$ and $r_c = 1$ and express all time and length scales in terms of these units. The length of the periodic x axis is set to 100. There is no limit on the y axis, but particles are bounded by the axial potential. By setting $c_{\text{ax}} = 1$, the particles in the system under consideration are effectively bounded between $y \in [-0.75, 0.75]$. There are $N = 50$ particles in the domain, resulting in an average density of 1 particle per $3r_c^2$. The absolute velocity of particles is set to $v_0 = 1$. Finally, the noise parameter η is set to 0.1. This value is high enough for the particles to switch to the opposite x direction but low enough to make the switching a rare event. Figure 14 shows a typical time evolution (obtained from a dynamical simulation) of the x and y components of the average particle velocity.

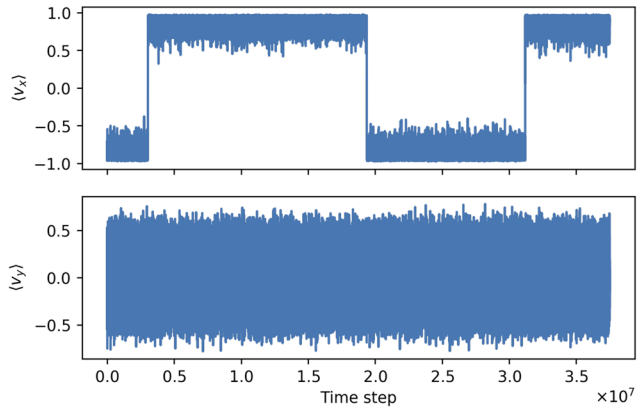

FIG. 14. Typical time evolution of the x (top panel) and y (bottom panel) components of the average particle velocity in the system with self-propelled particles.

We define the stable regions A and B as all system configurations for which $\langle v_x\rangle < -0.75v_0$ and $\langle v_x\rangle > 0.75v_0$, respectively. The first interface for the flux calculation is simply set equal to the boundary of A. The path progress variable is defined using the x and the y components of the velocities of all particles in the system, and we set $\alpha = 1/\Delta t$. In the PBCBMC method, we shoot 15 jets with a maximum path segment length of $50\Delta t$ and a final jet with segments that continue until reaching A or B. The bias strength is set to $b = 5$. For the PBRB method, interfaces are placed at the isolines of the path progress variable determined by the values {0.1, 0.15, 0.2, 0.25, 0.3, 0.4, 0.5}. In both methods, we run 10 trials per jet.

REFERENCES

[1] C. Dellago, P. G. Bolhuis, F. S. Csajka, and D. Chandler, J. Chem. Phys. **108**, 1964 (1998).
[2] P. G. Bolhuis, D. Chandler, C. Dellago, and P. Geissler, Annu. Rev. Phys. Chem. **53**, 291 (2002).
[3] C. Dellago, P. G. Bolhuis, and P. L. Geissler, Adv. Chem. Phys. **123**, 1 (2002).
[4] C. Dellago and P. G. Bolhuis, in *Atomistic Approaches in Modern Biology* (Springer Berlin Heidelberg, 2006), Vol. 268, pp. 291–317.
[5] C. Dellago and P. G. Bolhuis, Adv. Polym. Sci. **221**, 167 (2009).
[6] P. G. Bolhuis, C. Dellago, and D. Chandler, Proc. Natl. Acad. Sci. U. S. A. **97**, 5877 (2000).

Published under license by AIP Publishing

[7]T. S. van Erp, D. Moroni, and P. G. Bolhuis, J. Chem. Phys. **118**, 7762 (2003).
[8]T. S. van Erp and P. G. Bolhuis, J. Comput. Phys. **205**, 157 (2005).
[9]P. G. Bolhuis and C. Dellago, in *Reviews in Computational Chemistry*, edited by K. B. Lipkowitz (Wiley, Hoboken, 2010), Vol. 27, pp. 111–210 (2010).
[10]G. E. Crooks and D. Chandler, Phys. Rev. E **64**, 026109 (2001).
[11]R. J. Allen, P. B. Warren, and P. R. ten Wolde, Phys. Rev. Lett. **94**, 018104 (2005).
[12]R. J. Allen, D. Frenkel, and P. R. ten Wolde, J. Chem. Phys. **124**, 024102 (2006).
[13]A. Dickson, A. Warmflash, and A. R. Dinner, J. Chem. Phys. **131**, 154104 (2009).
[14]N. Guttenberg, A. R. Dinner, and J. Weare, J. Chem. Phys. **136**, 234103 (2012).
[15]T. R. Gingrich and P. L. Geissler, J. Chem. Phys. **142**, 234104 (2015).
[16]G. Huber and S. Kim, Biophys. J. **70**, 97 (1996).
[17]B. W. Zhang, D. Jasnow, and D. M. Zuckerman, J. Chem. Phys. **132**, 054107 (2010).
[18]J. T. Berryman and T. Schilling, J. Chem. Phys. **133**, 244101 (2010).
[19]C. Giardinà, J. Kurchan, and L. Peliti, Phys. Rev. Lett. **96**, 120603 (2006).
[20]T. Nemoto, F. Bouchet, R. L. Jack, and V. Lecomte, Phys. Rev. E **93**, 062123 (2016).
[21]U. Ray, G. K.-L. Chan, and D. T. Limmer, J. Chem. Phys. **148**, 124120 (2018).
[22]R. Cabriolu, K. M. S. Refsnes, P. G. Bolhuis, and T. S. van Erp, J. Chem. Phys. **147**, 152722 (2017).
[23]N. B. Becker, R. J. Allen, and P. R. ten Wolde, J. Chem. Phys. **136**, 174118 (2012).
[24]T. van Erp, Adv. Chem. Phys. **151**, 27 (2012).
[25]Note that in RBFFS, pathways can relax as already pointed out but are created from scratch and are not created with respect to the previous path. As a result, it is hard to control the acceptance ratio.
[26]D. Branduardi, F. L. Gervasio, and M. Parrinello, J. Chem. Phys. **126**, 054103 (2007).
[27]D. Frenkel and B. Smit, *Understanding Molecular Simulation*, 2nd ed. (Elsevier, 2002).
[28]J. Rogal, W. Lechner, J. Juraszek, B. Ensing, and P. G. Bolhuis, J. Chem. Phys. **133**, 174109 (2010).
[29]M. Bédard, Stochastic Processes Their Appl. **118**, 2198 (2008).
[30]C. Dellago, P. G. Bolhuis, and D. Chandler, J. Chem. Phys. **110**, 6617 (1999).
[31]G. O. Roberts, A. Gelman, and W. R. Gilks, Ann. Appl. Probab. **7**, 110 (1997).
[32]T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet, Phys. Rev. Lett. **75**, 1226 (1995).
[33]D. W. H. Swenson, J.-H. Prinz, F. Noe, J. D. Chodera, and P. G. Bolhuis, J. Chem. Theory Comput. **15**, 813 (2018).
[34]G. E. Crooks, J. Stat. Phys. **90**, 1481 (1998).
[35]C. Dellago, P. Bolhuis, and P. Geissler, in *Computer Simulations in Condensed Matter Systems: From Materials to Chemical Biology Volume 1*, edited by M. Ferrario, G. Ciccotti, and K. Binder (Springer Berlin Heidelberg, Berlin, Heidelberg, 2006), Vol. 703, pp. 349–391.
[36]C. Dellago, P. G. Bolhuis, and D. Chandler, J. Chem. Phys. **108**, 9236 (1998).
[37]M. Sachs, B. Leimkuhler, and V. Danos, Entropy **19**, 647 (2017).
[38]B. Leimkuhler and C. Matthews, J. Comput. Phys. **138**, 174102 (2013).
[39]G. Grégoire, H. Chaté, and Y. Tu, Physica D **181**, 157 (2003).
[40]H. Chaté, F. Ginelli, G. Grégoire, and F. Raynaud, Phys. Rev. E **77**, 046113 (2008).

Published under license by AIP Publishing