Exact rate calculations by trajectory parallelization and tilting

Eric Vanden-Eijnden\textsuperscript{a)} and Maddalena Venturoli\textsuperscript{b)}

\textit{Courant Institute of Mathematical Sciences, New York University, New York, New York 10012, USA}

(Received 16 April 2009; accepted 26 June 2009; published online 27 July 2009)

A sampling procedure to compute exactly the rate of activated processes arising in systems at equilibrium or nonequilibrium steady state is presented. The procedure is a generalization of the method proposed in [A. Warmflash \textit{et al.}, J. Chem. Phys. \textbf{127}, 154112 (2007); A. Dickson \textit{et al.}, J. Chem. Phys. \textbf{130}, 074104 (2009)] in which one performs simulations restricted into cells by using a reinjection rule at the boundaries of the cells which is consistent with the exact probability fluxes through these boundaries. Our generalization uses results from transition path theory which indicate how to tilt the dynamics to calculate reaction rates. © 2009 American Institute of Physics. [DOI: 10.1063/1.3180821]

I. INTRODUCTION

The main objective of this work is to revisit and extend in scope the nonequilibrium sampling procedure proposed in Refs. 1 and 2 to compute steady state probability distributions. Specifically, we show how this procedure can be modified to calculate exactly certain dynamical quantities such as the rate of reactions occurring in arbitrary equilibrium or nonequilibrium systems at statistical steady state. This is done by adopting the viewpoint of transition path theory (TPT)\textsuperscript{3–5} which indicates how to tilt the dynamics of a given system to calculate the reaction rate between a given reactant and product state. As we will show below, this tilting procedure is applicable also to equilibrium processes and amounts to placing them out of equilibrium by transforming the reactant state into a source and the product state into a sink. The method proposed in this paper can be seen as a generalization to arbitrary nonequilibrium processes at statistical steady state of the milestoning method proposed in Ref. 6 by building upon the original works in Refs. 7–9. This generalization makes the procedure more expensive computationally, but it permits to relax completely the assumptions made in milestoning—these assumptions were discussed in detail in Ref. 10. As explained below, our method can also be viewed as a generalization of the transition interface sampling (TIS)\textsuperscript{11–13} and forward flux sampling (FFS)\textsuperscript{14,15} methods in which arbitrary sets of interfaces can be used that do not have to be placed in monotone succession.

We will denote by $z$ the location of the system in its state-space $\Omega \subset \mathbb{R}^d$ (e.g., it could be the positions and velocities of all the atoms in a molecular system, in which case $z = (x, v)$ and $d = 6n$ if $n$ is the number of atoms). The specifics of the dynamics of the system are not important except that we assume that (i) its evolution is Markovian and (ii) it is ergodic with respect to a probability density function which we denote by $\rho(z)$. We do not require a detailed balance, i.e., $\rho(z)$ is associated with a nonequilibrium statistical steady state in general.

The remainder of this paper is organized as follows. First in Sec. II we revisit from an original perspective the nonequilibrium sampling procedure of Refs. 1 and 2. Next, in Sec. III, we show how this procedure can be modified to calculate reaction rates exactly using the TPT perspective. In Sec. IV we then compare our procedure with existing methods to compute rates: TIS\textsuperscript{11–13} FFS\textsuperscript{14,15} and the Markovian milestoning method proposed in Ref. 6. Finally in Sec. V we illustrate our procedure on a simple example.

II. SAMPLING BY TRAJECTORY PARALLELIZATION

The methods in Refs. 1, 2, and 6 are based on factorization of the dynamics in which one artificially constrains the system to evolve in a set of cells partitioning state space in a way that (i) does not bias the dynamics inside the cells and (ii) preserves the exact probability fluxes in and out of these cells. In other words, the procedure guarantees that a true unconstrained trajectory of the system can be reconstructed exactly by patching together in some appropriate way the pieces computed in the cells. These pieces can be calculated in parallel in each cell, hence the name trajectory parallelization. The method in Ref. 6 is restricted to equilibrium systems and exploits the time reversibility of the dynamics. The method in Refs. 1 and 2 is more costly but it works also for nonequilibrium systems. To explain how the latter method works, we first recall how to construct the cells using the Voronoi tessellation associated with a given set of generating points or centers.\textsuperscript{6,16} Denoting these centers by $z_n \in \Omega \subset \mathbb{R}^d$ with $n = 1, \ldots, \Lambda$, the Voronoi cell $B_n$ associated with $z_n$ contains all the points that are closer to $z_n$ than to any other center, i.e., (see Fig. 1 for an illustration)

$$B_n = \{ z \in \Omega : \| z - z_n \| < \| z - z_\beta \| \} \text{ for all } \beta \neq n, \quad \text{(1)}$$

where $\| \cdot \|$ is some appropriate norm (e.g., the Euclidean norm in which case $\| z \|^2 = \sum_{i=1}^{d} z_i^2$).

If we were to generate an infinitely long trajectory of the system $z(t)$ with $t > 0$, this trajectory would keep going in...
and out of the cells by crossing the edges between them. Out of this trajectory we could therefore generate an ensemble of exit-entry points, i.e., those points on the edges of the cells at which the trajectory goes from a cell $B_\alpha$ into a neighboring cell $B_\beta$. By the Markovian assumption, these points are all we need to generate an exact sample of trajectories inside each of the cells $B_\alpha$ simply by starting trajectories at the points leading into that cell and running these trajectories forward in time until they exit the cell. The procedure in Refs. 1 and 2 is a way to generate these pieces of trajectories inside the cells without having to compute the entry points beforehand from a long unbiased trajectory but rather by generating them on the fly. To understand how this is done, imagine that we associate an independent copy or replica of the system to each cell $B_\alpha$. Let us denote the instantaneous position of these replicas by $z_\alpha(t) \in B_\alpha$, $\alpha = 1, \ldots , \Lambda$. Even if we start $z_\alpha(t)$ inside $B_\alpha$, sooner or later this trajectory will try to exit $B_\alpha$ and go to another cell. When this happens, we store the exit point on the boundary, put the trajectory on hold, and wait until a trajectory in one of the neighboring cells makes an attempt to exit this cell by crossing the boundary with $B_\alpha$. We then take this crossing point and use it to reinitialize the trajectory in $B_\alpha$. This is illustrated in Fig. 1. At the beginning of the simulation we do not have enough reentry points, so several replicas may be on hold. But as the simulation goes on, we can build databanks of reentry points from any $B_\beta$ into any $B_\alpha$ (assuming that these two cells have a common boundary—otherwise the databank is trivially empty), which contain the last $X$ points by which the trajectory tried to escape from cell $B_\beta$ and enter cell $B_\alpha$. This way, each time a trajectory tries to exit from cell $B_\alpha$, we can immediately pick a reentry point into that cell from the appropriate databank and continue the trajectory from that point without having to put it on hold.

The only remaining issue we need to take care of in order to make complete the procedure outlined above is how to pick the reentry point into $B_\alpha$ among the databanks on the various edges leading into $B_\alpha$. By construction, the reentry points in the databanks on each edge are unbiased samples of reentry points conditional on the trajectory entering by that edge. But there are several edges by which the trajectory can enter a given cell, and to introduce no bias we need to pick an edge with the proper probability of reentrance by that edge. To see how this can be done, imagine that as we run the simulation in each cell $B_\alpha$, we compute an estimate of the effective rate of exit out that cell and into $B_\beta$ via

$$
\nu_{\alpha,\beta} = \frac{N_{\alpha,\beta}}{T_\alpha},
$$

where $N_{\alpha,\beta}$ is the total number of times the trajectory hit the boundary between $B_\alpha$ and $B_\beta$ (which is also the number of times the trajectory in cell $B_\alpha$ had to be reinserted into that cell from a reentry point) and $T_\alpha$ is the total simulation time in cell $B_\alpha$ (i.e., the total time the trajectory in cell $B_\alpha$ has been running). If we then denote by $\pi_\alpha$ the probability to find the unbiased trajectory inside cell $B_\alpha$ at statistical steady state, i.e.,

$$
\pi_\alpha = \int_{B_\alpha} \rho(z) dz,
$$

we see that $\pi_\alpha$ and $\nu_{\alpha,\beta}$ are related by

$$
\sum_{\beta \neq \alpha} \pi_\beta \nu_{\beta,\alpha} = \sum_{\beta \neq \alpha} \pi_\alpha \nu_{\alpha,\beta} \sum_{\alpha=1}^{\Lambda} \pi_\alpha = 1.
$$

The first equation in Eq. (4) simply expresses that at statistical steady state, the total probability flux into $B_\alpha$ [which is the term at the left-hand side of the first equation in Eq. (4)] must be equal to the total flux out of $B_\alpha$ [which is the term at the right-hand side of the first equation in Eq. (4)]. The second equation in Eq. (4) is simply a normalization condition for the probability which follows from $\sum_\alpha \pi_\alpha = \sum_\alpha \int_{B_\alpha} \rho(z) dz = 1$. We stress that Eq. (4) does not require a detailed balance (i.e., $\pi_\beta \nu_{\beta,\alpha} \neq \pi_\alpha \nu_{\alpha,\beta}$ possibly) and so it holds even for nonequilibrium processes provided only that they are at statistical steady state. Having calculated $\nu_{\alpha,\beta}$ from Eq. (2) and $\pi_\alpha$ from Eq. (4), we then have an estimate for the probability flux from $B_\beta$ into $B_\alpha$: $\nu_{\beta,\alpha} \pi_\alpha$. Consistently, the probability that the trajectory enters cell $B_\alpha$ by coming from $B_\beta$ is simply

$$
\eta_{\beta \implies \alpha} = \frac{\pi_\beta \nu_{\beta,\alpha}}{\sum_{\beta' \neq \alpha} \pi_{\beta'} \nu_{\beta',\alpha}}, \quad (\beta \neq \alpha),
$$

if $B_\alpha$ and $B_\beta$ have a common edge and $\eta_{\beta \implies \alpha} = 0$ otherwise. This expression gives us the desired probability to pick the edge $\partial B_\beta \cap \partial B_\alpha$ for reentry into $B_\alpha$.

Summarizing, the algorithm to perform simulations restricted inside the cells for systems at steady state is as follows.

1. Denoting by $z_\alpha(t) \in B_\alpha$ the current state of the replica in cell $B_\alpha$, let $z^*_{\alpha}$ be the state produced from $z_\alpha(t)$ after one time step $\Delta t$ by a standard (i.e., unrestricted) integrator for the system (e.g., velocity Verlet if the system is a molecular dynamics system). If $z^*_{\alpha} \in B_\alpha$, set

2. If $z^*_{\alpha} \notin B_\alpha$, we have to reinitialize the trajectory at this reentry point.

3. The search for the reentry point is performed using a standard procedure as described in Refs. 1 and 2.

4. Given that the trajectory is now at a reentry point, we compute the rate at which the trajectory will exit cell $B_\alpha$, $\nu_{\alpha,\beta}$, by using the aforementioned algorithm.

5. We then use the probability flux from $B_\beta$ into $B_\alpha$, $\eta_{\beta \implies \alpha}$, to select the edge $\partial B_\beta \cap \partial B_\alpha$ for reentry into $B_\alpha$.

6. If we are in the case of nonequilibrium systems, we will have to use the correct probability flux from $B_\beta$ into $B_\alpha$ for reentry into $B_\alpha$.

7. After reinitializing the trajectory at the reentry point, the algorithm proceeds using the new state of the system.
\[ z_a(t + \Delta t) = z_a^*. \]  

(6)

Otherwise, if \( z_a^* \in B_\beta \) with \( \beta \neq \alpha 

(i) Store the point \( z_a^* \) in a databank of entry points from \( B_\alpha \) into \( B_\beta \). In other words, assuming that the databank contains already \( m \) points \( z_{a,\beta}^k \) with \( k = 1, 2, \ldots, m \), set \( z_{a,\beta}^{m+1} = z_a^* \).

(ii) Update \( v_{a,\beta} \) via Eq. (2), \( \pi_a \) via Eq. (4), and \( \Pi_{\partial B_\beta \cap \partial B_\alpha} \) via Eq. (5).

(iii) Select an edge \( \partial B_\beta \cap \partial B_\alpha \) of \( B_\alpha \) with probability proportional to \( \Pi_{\partial B_\beta \cap \partial B_\alpha} \).

(iv) Pick a point \( z_{\beta',\alpha}^* \) with uniform probability from the databank of reentry points on edge \( \partial B_\beta \cap \partial B_\alpha \) and set

\[ z_a(t + \Delta t) = z_{\beta',\alpha}^*. \]  

(7)

(2) Go to step (1) and iterate to collect statistics.

The algorithm above is in essence the same as the one proposed in Refs. 1 and 2, although the two differ in details. For instance, we use a single set of cells instead of two (in Refs. 1 and 2 two staggered sets were used to ensure stability, but we observed no such stability problems with the procedure above). We also compute the \( \pi_a \) differently via the solution of the system in Eq. (4).

Besides a set of trajectories, one of the outputs of the procedure is to give the probability \( \pi_a \) to find the system in cell \( B_\alpha \) at statistical steady state [see Eq. (3)]. Indeed the computation of \( \pi_a \) was the main objective in Refs. 1 and 2. Below we will show how to extract more from the procedure, in particular, reaction rate information, by appropriate modifications. Before getting there, however, let us make a few remarks about the algorithm above. In step (iv) we have assumed that the list of entry points from \( B_\beta ' \) into \( B_\alpha \) is not empty and, as already mentioned above, this may not be true at the beginning of the simulation (there might have been no collision with that edge up to that time). In that case we have to put the simulations in some of cells on hold at the beginning until we get proper reentry points and start building databases on the edges. Also note that the quantities used to evaluate the probability in Eq. (5) have to be computed on the fly and need information from all the cells. This again may lead to problems at the beginning of the simulations when the statistics for \( N_{a,\beta} \) is not accurate enough. To overcome this problem we set \( \pi_a = 1/\Lambda \), \( \forall \alpha = 1, \ldots, \Lambda \) at the beginning when statistics is insufficient to solve Eq. (4). Using more educated guesses is possible too. Also, it is worth noting that \( v_{a,\beta} \) and \( \pi_a \) can be monitored on the fly to assess their convergence as a function of the length of the simulation, and the actual simulations of the replicas in each cell can be performed in parallel; only the reentry events require communication. Finally, we should stress that the procedure above relies on the ability to count the successive points at which a trajectory crosses the edges of the cells. This may lead to difficulties if the dynamics is governed by a stochastic differential equation, in which case these crossing points may form a fractal set. It leads to no difficulty, however, if some components of the trajectory are smooth and the cells are defined accordingly. We will come back to this issue later in Sec. V.

III. RATE CALCULATION BY TILTING

Let us now come to the question of how to compute the reaction rate between a reactant and a product state, which we identify as two disjoint sets in the system’s state space denoted as \( A \subset \Omega \) and \( B \subset \Omega \), respectively. As mentioned earlier, this calculation will be done by tilting the dynamics in some appropriate way consistent with the TPT perspective. We begin by introducing the relevant objects that we will need. Consider again a thought experiment in which we would have at our disposal an infinitely long trajectory \( z(t) \) with \( t > 0 \). This trajectory would go back and forth between \( A \) and \( B \) as time goes on, and we could split this trajectory into two pieces, depending on whether it visited \( A \) or \( B \). This construction is illustrated in Fig. 2, where the trajectory is shown in red if it visited \( A \) last and in black if it visited \( B \) last: If the trajectory visited \( A \) last at time \( t \), we will say that it is assigned to \( A \) at time \( t \); if it visited \( B \) last at time \( t \), we will say that it is assigned to \( B \) at time \( t \). A similar assignment procedure was proposed in Ref. 17. Trajectories assigned to \( A \) are also the ones targeted by the weighted-ensemble Brownian dynamics simulation procedure proposed in Ref. 18 in the context of overdamped dynamics.

Based on this assignment, we can introduce the following quantities. If \( N_{A,B}^T \) denotes the total number of times the trajectory went from being assigned to \( A \) to being assigned to \( B \) during the time interval \( [0, T] \) (i.e., the number of times it switched from red to black in Fig. 2), we set

\[ v_R = \lim_{T \to \infty} \frac{N_{A,B}^T}{T}. \]  

(8)

This quantity gives the average frequency at which the trajectory goes from \( A \) to \( B \), which, because the system is at steady state, is also the same as the average frequency at which the trajectory goes from \( B \) to \( A \), since \( N_{A,B}^T = N_{B,A}^T \) asymptotically. The rate in Eq. (8) is referred to as the rate of the reactive trajectories in TPT, hence the subscript \( R \). The rate of reactive trajectories \( v_R \) should not be confused with the two reaction rates from \( A \) to \( B \) and \( B \) to \( A \) defined, respectively, as

FIG. 2. Schematic representation of a piece of ergodic trajectory visiting the two sets \( A \) and \( B \). The pieces of this trajectory for which the last visited set was \( A \) are depicted in red, and those for which the last visited set was \( B \) are depicted in black.
where \( T_A \) and \( T_B \) are, respectively, the total time during which the trajectory was assigned to \( A \) or \( B \) in the interval \([0, T]\) (i.e., the total times the trajectory is red or black in Fig. 2 and \( T_A+T_B=T \)). If \( A \) and \( B \) are metastable (i.e., if the trajectory commits to each of these sets and loses memory of its past before going back to the other set), \( k_{A,B} \) and \( k_{B,A} \) are the rates that enter the phenomenological mass-action law describing how the populations in \( A \) and \( B \) evolve in time. Notice that \( \nu_R \) is related to \( k_{A,B} \) and \( k_{B,A} \) as

\[
k_{A,B} = \frac{\nu_B}{\rho_A}, \quad k_{B,A} = \frac{\nu_B}{\rho_B},
\]

where \( \rho_A \) is the fraction of time the trajectory is assigned to \( A \) and \( \rho_B \) is the fraction of time it is assigned to \( B \).

By definition \( \rho_A \leq 1, \quad \rho_B \leq 1, \quad \rho_A+\rho_B=1 \). The quantities \( \nu_R, \ k_{A,B}, \ k_{B,A}, \ \rho_A, \ \rho_B \) are the ones we will show how to compute exactly. To see how this can be done, it is useful to give first the TPT expressions for these quantities. These expressions involve the backward committor function \( q_B(z) \), which gives the probability that a trajectory observed at point \( z \) is coming from \( A \) last rather than from \( B \) (i.e., that it is assigned to \( A \) rather than \( B \) using the jargon introduced above). By definition \( q_B(z)=1 \) if \( z \in A \), \( q_B(z)=0 \) if \( z \in B \), and \( 0 \leq q_B(z) \leq 1 \) otherwise. The backward committor function is useful because by the Markovian assumption it follows that at statistical steady state, the probability density to observe a trajectory at point \( z \in \Omega \) at any given time \( t \) and that this trajectory is assigned to \( A \) at that time is simply \( \rho_A(z) = q_B(z)q_A(z) \). Similarly the probability density to observe a trajectory at point \( z \in \Omega \) at time \( t \) and that this trajectory is assigned to \( B \) at that time is \( \rho_B(z) = q_A(z)(1-q_B(z)) \). Since \( \rho_A = \int_{\Omega} q_A(z) dz \) and \( \rho_B = \int_{\Omega} q_B(z) dz \) by definition, this implies that

\[
\rho_A = \int_{\Omega} q_B(z)q_A(z) dz \leq 1, \quad \rho_B = 1 - \rho_A.
\]

Similarly, it is easy to see that \( \nu_R \) is the total probability flux associated with \( q_B(z) = q_B(z)q_A(z) \) going through any dividing surface between \( A \) and \( B \) (such as, e.g., the boundary of \( B \)). The explicit form of this flux depends on the specifics of the dynamics and it is given in Ref. 4. Let us omit to repeat this formula here since it will not be important in what follows.

In practice, we do not know explicitly \( q_B(z) \) (or even \( q_B(z) \) in most nonequilibrium systems) but we can still make use of the observations above to modify the sampling procedure explained before. Suppose that we define the reactant state \( A \) as being the union of a group of cells and the product state \( B \) as the union of another group. Clearly, what we would then like to do is modify the sampling procedure in such a way that only the reentry points associated with the trajectory when it is assigned to \( A \) are put in the databanks (see the illustration in Fig. 3) and keep track of the associated probability fluxes through the boundary of the cells. Indeed, using these reentry points and these fluxes only, we would then simulate in the cells pieces of trajectories that are statistically indistinguishable from the unbiased trajectory when it is assigned to \( A \). We could then compute the total probability in the cells to get an estimate of \( \rho_A \) (and hence \( \rho_B=1-\rho_A \)) as well as the total flux into \( B \) to get \( \nu_R \). We claim that there is a simple procedure to do these operations in practice. The key observation is that trajectories assigned to \( A \) can be propagated like regular trajectories while they are outside of \( A \) and \( B \)—the only constraint imposed on them is via a boundary condition in their past: They need to have come from \( A \) rather than \( B \) last. But we know what this boundary condition entails at least in a statistical sense: By definition the probability flux out of \( A \) of the trajectories assigned to \( A \) is the statistical steady state flux and their probability flux out of \( B \) is identically zero. We also know that the probability density \( q_A(z) \) that a trajectory be at \( z \) and be assigned to \( A \) is equal to the statistical steady state probability density \( q(z) \) for \( z \in A \) while it is equal to 0 for \( z \in B \).

We can easily impose these boundary conditions in practice by modifying our sampling procedure as follows. Suppose that we have performed a sampling as before and computed the steady state (unbiased) \( \pi_a \) and \( \nu_{a,b} \). We can then run another independent sampling where we only consider the cells outside of \( A \) and \( B \). In these cells, we run trajectories as before (although, as we will see, using reentry points that are different from the ones calculated before), store their exit points from the cells to build databanks of reentry points in other cells, and compute

\[
\nu^a_{a,b} = \frac{N^a_{a,b}}{T^a_a},
\]

where \( N^a_{a,b} \) is the total number of times the trajectory hit the boundary between \( B_a \) and \( B_b \) and \( T^a_a \) is the total simulation run.
time in cell $B_a$. Note that $\nu^{A}_{\alpha,\beta}$ is only defined by Eq. (13) if $\alpha$ is the index of a cell $B_a$ not inside $A$ and $B$ (the index $\beta$, on the other hand, runs over all the cells, including those forming $A$ and $B$). Consistent with the bias we need to impose to focus on trajectories assigned to $A$, we supplement this by $\nu^{A}_{\alpha,\beta} = 0$ if $\alpha$ is the index of a cell $B_a$ used to define $A$ (since the effective rate of exit out of $A$ must be the unbiased statistical steady state one) and by $\nu^{A}_{\alpha,\beta} = 0$ if $\alpha$ is the index of a cell $B_a$ used to define $B$ (since the effective rate of exit out of $B$ must be zero). We also set $\pi^0_{\alpha} = \pi_{\alpha}$ if $\alpha$ is the index of a cell $B_a$ used to define $A$, $\pi^0_{\alpha} = 0$ if $\alpha$ is the index of a cell $B_a$ used to define $B$, and in all the other cells we compute $\pi^A_{\alpha}$ via

$$
\sum_{\beta \neq \alpha} \pi^A_{\beta,\alpha} = \sum_{\beta \neq \alpha} \frac{\pi^A_{\beta,\alpha}}{\sum_{\beta' \neq \alpha} \pi^A_{\beta',\alpha}}, \quad (\beta \neq \alpha).
$$

(14)

where the index $\alpha$ runs over all the cells outside of $A$ and $B$ and we use as boundary conditions the values for $\nu^{A}_{\alpha,\beta}$ and $\pi^A_{\alpha}$ set before when the index $\beta$ is that of a cell used to define $A$ or $B$. Finally, we compute the probability of reentry on the edges of a cell $B_a$ not inside $A$ and $B$ as

$$
\pi^A_{\beta,\alpha} = \frac{\sum_{\beta' \neq \alpha} \pi^A_{\beta',\alpha}}{\sum_{\beta' \neq \alpha} \pi^A_{\beta',\alpha}}.
$$

(15)

This procedure automatically guarantees that the focusing on the trajectories assigned to $A$. In particular, by construction we have $\pi^A_{\beta,\alpha} = \int_{B_a} q_{\alpha}(z)dz$, which from Eq. (12) means that

$$
\rho_A = \sum_{\alpha = 1}^{\Lambda} \pi^A_{\alpha} = 1, \quad \rho_B = 1 - \rho_A.
$$

(16)

Similarly we can compute $\nu_B$ from the total flux into $B$.

$$
\nu_B = \sum_{\alpha \text{ such that } B_a \text{ not in } B} \sum_{\beta \text{ such that } B_a \text{ in } \beta} \pi^A_{\alpha,\beta} \nu^{A}_{\alpha,\beta}.
$$

(17)

We can then get $k_{A,B}$ and $k_{B,A}$ from Eq. (10). Finally notice that the average backward committor in each cell $B_a$ (which gives the probability that trajectories in that cell are assigned to $A$ rather than $B$) can be computed as the ratio between $\pi^A_{\alpha}$ and $\pi^A_{\alpha}$. Indeed,

$$
\pi^A_{\alpha} = \int_{B_a} q_{\alpha}(z)dz = \int_{B_a} q(z)q_{\alpha}(z)dz = \frac{\int_{B_a} q(z)q_{\alpha}(z)dz}{\int_{B_a} q(z)dz} = \langle q_{\alpha} \rangle_{B_a} \pi^A_{\alpha},
$$

(18)

from which follows that $\langle q_{\alpha} \rangle_{B_a} = \pi^A_{\alpha} / \pi^A_{\alpha}$.

IV. COMPARISON WITH OTHER METHODS

Let us briefly compare our tilting procedure to existing methods. First, let us point out that it is very similar in spirit to both TIS\textsuperscript{11–13} and FFS.\textsuperscript{14,15} Like TIS and FFS, our method is based on selecting only those trajectories which come from the reactant state $A$. The difference is in the way this selection is achieved. In particular, unlike in TIS and FFS, we do not require that the interfaces be ordered monotonously, i.e., we do not need that a trajectory coming from $A$ crosses all the preceding interfaces before reaching the next one. This allows to run the procedure in parallel and offers more flexibility in the way the interfaces can be chosen. Here we did so using the edges of cells in a Voronoi tessellation because it is convenient, but the formalism above is clearly independent of that choice and can be applied to any type of interfaces.

Regarding the relation with the Markovian milestoning procedure with Voronoi tessellation,\textsuperscript{6} the main advantage of the method proposed in this paper to compute the rate is that it is exact and hence avoids completely the assumptions underlying milestoning.\textsuperscript{10} On the other hand, the new procedure is more costly since it requires not only to build databases of reentry points but also to do the sampling twice—once to get the unbiased statistical steady states quantities and once more to get the reaction rate by tilting the dynamics. Moreover, at variance with the method presented here, Markovian milestoning does not require any communication between the replicas in the cells. For systems at equilibrium, one may therefore prefer to use the Markovian milestoning procedure with Voronoi tessellation,\textsuperscript{6} which is cheaper. For nonequilibrium systems, this procedure is inapplicable (since it relies on the time reversibility of the dynamics), but as a compromise, one could use the nonequilibrium sampling strategy for the unbiased system to compute the relevant quantities in Markovian milestoning and thereby avoid to make the second tilted sampling to compute the rate. Indeed, the formalism developed in Ref. 6 to approximate the dynamics by a continuous-time Markov chain does not rely on the dynamics being at equilibrium. For completeness let us briefly recall the main objects in Markovian milestoning and indicate how to compute them in the present context. If, following Ref. 6, we define the milestones as the common boundaries between any two adjacent Voronoi cells and denote these milestones by $S_i$ with $i = 1, 2, \ldots, N$, the key quantity to approximate the transitions between the milestones by a continuous-time Markov chain is the rate matrix whose off-diagonal elements $d_{ij} \geq 0$ with $i \neq j$ give the rate at which the trajectory crosses milestone $S_i$ after having crossed $S_j$. As shown in Ref. 6, these off-diagonal elements can be estimated as

$$
q_{ij} = N_{ij}/R_i
$$

(19)

where

$$
N_{ij} = T \sum_{a=1}^{\Lambda} \pi^A_{\alpha} \frac{N_{ij}^T}{T_a}, \quad R_i = T \sum_{a=1}^{\Lambda} \pi^A_{\alpha} \frac{R_i^T}{T_a}.
$$

(20)

Here $T_a$ is the total simulation time in cell $B_a$ (as before) and $T = (\sum_{a=1}^{\Lambda} \pi^A_{\alpha})^{-1}$ is added for dimensional consistency (note that is it not needed in the calculation of $q_{ij}$). $N_{ij}^T$ is the total number of times the trajectory went from milestone $S_i$ to milestone $S_j$ and $R_i^T$ is the total amount of time the trajectory is assigned to milestone $S_i$ in cell $B_a$, i.e., the total amount of time this trajectory is such that $S_i$ was the edge of cell $B_a$ it hit last. The quantities can be easily estimated by running the procedure described in Sec. II. For more details on the milestoning procedure, what can be extracted from it,
and the assumptions upon which it relies, we refer the reader to Refs. 6 and 10.

V. ILLUSTRATIVE EXAMPLE

Let us now illustrate our sampling procedure on the example of a system evolving by Langevin dynamics on a two-dimensional potential, i.e., $z(t) = (x(t), v(t))$, $x, v \in \mathbb{R}^2$, and

$$\dot{x}(t) = v(t),$$

$$\dot{v}(t) = -\nabla V(x(t)) - \gamma v(t) + \sqrt{2\beta^{-1}} \eta(t).$$

(21)

Here $V(x)$ is the Mueller potential\(^{19}\) whose contour plot is shown in Fig. 4, $\beta = 1/(k_B T)$ is the inverse temperature, and $\eta(t)$ is a Gaussian white noise with mean zero and covariance $\langle \eta_i(t) \eta_j(t') \rangle = \delta_{ij} \delta(t - t')$. $\gamma$ is the friction coefficient and, for simplicity, we have set the mass tensor to the identity. Below we took $\gamma = 100$ and $\beta^{-1} = 20$ (which is about 20% of the value of the energy barrier between the minimum at the top left corner of the Mueller potential and the saddle point at $(-0.8, 0.6)$]. This example is obviously very simple and serves no other purpose than being a benchmark: The application of our sampling procedure to more interesting examples will be reported elsewhere. To avoid confusions, before presenting our results for Eq. (21) let us note that this system is an equilibrium system with equilibrium density $\bar{Q}(x, v) = Z^{-1} \exp(-\beta H(x, v))$, where $H(x, v) = \frac{1}{2} |v|^2 + V(x)$ is the Hamiltonian and $Z = \int_{\mathbb{R}^2} \exp(-\beta H(x, v)) dx dv$ is the partition function. We are, however, primarily interested in computing the reaction rates between the reactant state $A$ and the product state $B$ shown in Fig. 4, which we will achieve by tilting the dynamics as explained before to focus on trajectories assigned to $A$. In so doing, we automatically put the system out of equilibrium since $A$ becomes a source and $B$ a sink [and $\dot{Q}_A(x, v) \neq \dot{Q}(x, v)$]. This is why we need the non-equilibrium formalism developed above to compute the reaction rates even in the case of an equilibrium system such as Eq. (21).

Shown in Fig. 4 are the 36 cells that we used in our calculations. These cells were defined as

$$B_\alpha = \{ (x, v) \in \mathbb{R}^2 \times \mathbb{R}^2 : |x - x_\alpha| < |x - x_\beta| \}$$

for all $\beta \neq \alpha$, (22)

where $|\cdot|$ denotes the Euclidean norm and $x_\alpha$ with $\alpha = 1, \ldots, 36$ are points chosen randomly in the region where the energy is below a certain threshold value. We identified the reactant and product states with two single cells in the neighborhood of the two deep minima in the potential landscape: $A = B_1$ and $B = B_{36}$ (see Fig. 4). Note that we intentionally took many cells and disposed them in a way that may not be optimal for the reaction to check that our procedure is robust against such choices. A more efficient way to define the cells which remains applicable for high dimensional examples is to take them as discretization points along transition paths computed, e.g., by the string method.\(^ {16,20-23}\) Note also that by defining the cells via a distance check involving only the positions, as we do in Eq. (22), we guarantee that we can count successive crossing of their boundaries. This would not be the case if the boundaries of the cells were bent in velocity space because of the noise term in Eq. (21) that acts on the velocities.

Both steps of the sampling procedure (the one to compute the unbiased equilibrium quantities and the other with the tilted dynamics) were performed, as described above, by running simulations for $10^8$ steps in each cell using the second order integrator of Ref. 24 with a time step $\Delta t = 10^{-4}$. In the first step of the procedure we took $\pi_{\alpha} = 1/36$ in each cell as initial condition; in the second step, we took $\pi_{\alpha}' = \pi_{\alpha}$ as initial condition. We compared the results of our procedure to those obtained by generating a long $(10)^{10}$ unbiased trajectory by brute force simulation and computing $\nu_R$, $k_{A,B}$, and $k_{B,A}$ from finite $T$ approximations of the limits in Eqs. (8) and (9).

The main outputs of our procedure are the rate of reactive trajectories $\nu_R$ and the reaction rates $k_{A,B}$ and $k_{B,A}$, which are reported in Table I. These are within the statistical errors of the corresponding quantities estimated by brute force simulation. Our procedure also produces the equilibrium probabilities $\pi_{\alpha}'$: As shown in Fig. 5 these are within the statistical errors of the corresponding values obtained by the brute force simulation. Also shown in Fig. 5 are the probabilities $\pi_{\alpha}'$ computed by tilting the dynamics. As expected, the closer to $B$, the smaller $\pi_{\alpha}'$ is compared to $\pi_{\alpha}$.

| $\nu_R$ | $k_{A,B}$ | $k_{B,A}$ |
|---------|-----------|-----------|
| $5.9 \times 10^{-3}$ | $7.4 \times 10^{-3}$ | $3.1 \times 10^{-2}$ |

| $5.8 \times 10^{-3}$ | $7.1 \times 10^{-3}$ | $3.2 \times 10^{-2}$ |

FIG. 4. Contour plot of the Mueller potential with the 36 points (shown as gray dots) generating the Voronoi tessellation shown as gray lines. The reactant $A$ and product $B$ states are identified with cells $B_1$ and $B_{36}$, respectively.
VI. CONCLUDING REMARKS

Accelerated sampling strategies for equilibrium systems have a long history and many such methods are available nowadays. For nonequilibrium systems, however, or more generally, when one is interested in dynamical properties such as reaction rates, the situation is quite different and the field remains widely open. In this paper, we showed how the technique recently introduced in Refs. 1 and 2 can be revisited and generalized to permit not only to sample nonequilibrium statistical steady state distributions but also to compute exactly some dynamical quantities such as reaction rates. As we showed, this second objective can be achieved by tilting the dynamics, i.e., by transforming the reactant state into a source and the product state into a sink. The tilting procedure is applicable to systems both at equilibrium and nonequilibrium (in the former case, it artificially places the system out of equilibrium to permit the calculation of the rate), and we showed that in spirit this procedure shares similarities with a method such as TIS or FFS. We also discussed the relation of our sampling with milestoning. A more complete comparison of these different techniques will be the object of a future publication. Other topics that remain to be investigated more thoroughly include a more detailed analysis of the efficiency of the method, especially when used in combination with techniques such as the string method, as well as the investigation of how to place the cells optimally in terms of efficiency.

ACKNOWLEDGMENTS

This work was motivated by a question from Benoît Roux. Partial support by NSF (Grant Nos. DMS-0718172 and DMS-0708140) and ONR (Grant No. N00014-04-1-6046) is also acknowledged.

1 A. Warmflash, P. Bhimalapuram, and A. R. Dinner, J. Chem. Phys. 127, 154112 (2007).
2 A. Dickson, A. Warmflash, and A. R. Dinner, J. Chem. Phys. 130, 074104 (2009).
3 W. E and E. Vanden-Eijnden, J. Stat. Phys. 123, 503 (2006).
4 E. Vanden-Eijnden, in Computer Simulations in Condensed Matter: From Materials to Chemical Biology, edited by M. Ferrario, G. Cicotti, and K. Binder (Springer, New York, 2006), Vol. 1, pp. 439–478.
5 P. Metzner, C. Schütte, and E. Vanden-Eijnden, J. Chem. Phys. 125, 084110 (2006).
6 E. Vanden-Eijnden and M. Venturoli, J. Chem. Phys. 130, 194101 (2009).
7 A. K. Faradjian and R. Elber, J. Chem. Phys. 120, 10880 (2004).
8 A. M. A. West, R. Elber, and D. Shalloway, J. Chem. Phys. 126, 145104 (2007).
9 R. Elber, Biophys. J. 92, L85 (2007).
10 E. Vanden-Eijnden, M. Venturoli, G. Cicotti, and R. Elber, J. Chem. Phys. 129, 174102 (2008).
11 D. Moroni, T. S. van Erp, and P. G. Bolhuis, Physica A 340, 395 (2004).
12 D. Moroni, P. G. Bolhuis, and T. S. van Erp, J. Chem. Phys. 120, 4055 (2004).
13 T. S. van Erp and P. G. Bolhuis, J. Comput. Phys. 205, 157 (2005).
14 R. J. Allen, D. Frenkel, and P. R. ten Wolde, J. Chem. Phys. 124, 024102 (2006).
15 C. Valeriani, R. J. Allen, M. J. Morelli, D. Frenkel, and P. R. ten Wolde, J. Chem. Phys. 127, 114109 (2007).
16 E. Vanden-Eijnden and M. Venturoli, J. Chem. Phys. 130, 194103 (2009).
17 N.-V. Buchete and G. Hummer, J. Phys. Chem. B 112, 6057 (2008).
18 G. A. Huber and S. Kim, Biophys. J. 70, 97 (1996).
19 K. Müller, Angew. Chem. 19, 1 (1980).
20 W. E, W. Ren, and E. Vanden-Eijnden, Phys. Rev. B 66, 052301 (2002).
21 W. E, W. Ren, and E. Vanden-Eijnden, J. Chem. Phys. 126, 164103 (2007).
22 L. Maragliano, A. Fischer, E. Vanden-Eijnden, and G. Cicotti, J. Chem. Phys. 125, 024106 (2006).
23 W. E, W. Ren, and E. Vanden-Eijnden, J. Phys. Chem. B 109, 6688 (2005).
24 E. Vanden-Eijnden and G. Cicotti, Chem. Phys. Lett. 429, 310 (2006).