

## Separating forward and backward pathways in nonequilibrium umbrella sampling

Alex Dickson, Aryeh Warmflash, and Aaron R. Dinner

Citation: *The Journal of Chemical Physics* **131**, 154104 (2009); doi: 10.1063/1.3244561

View online: <http://dx.doi.org/10.1063/1.3244561>

View Table of Contents: <http://scitation.aip.org/content/aip/journal/jcp/131/15?ver=pdfcov>

Published by the [AIP Publishing](#)

---

### Articles you may be interested in

Erratum: "Separating forward and backward pathways in nonequilibrium umbrella sampling" [*J. Chem. Phys.* **131**, 154104 (2009)]

*J. Chem. Phys.* **136**, 239901 (2012); 10.1063/1.4730937

Determination of free energy profiles by repository based adaptive umbrella sampling: Bridging nonequilibrium and quasiequilibrium simulations

*J. Chem. Phys.* **128**, 204106 (2008); 10.1063/1.2920476

Umbrella sampling for nonequilibrium processes

*J. Chem. Phys.* **127**, 154112 (2007); 10.1063/1.2784118

Analysis of the statistical error in umbrella sampling simulations by umbrella integration

*J. Chem. Phys.* **124**, 234106 (2006); 10.1063/1.2206775

Exploring reaction pathways with transition path and umbrella sampling: Application to methyl maltoside

*J. Chem. Phys.* **124**, 114113 (2006); 10.1063/1.2172604

---

COMSOL  
CONFERENCE  
2014 BOSTON

The Multiphysics  
Simulation  
Event of the Year

LEARN MORE >>

COMSOL

# Separating forward and backward pathways in nonequilibrium umbrella sampling

Alex Dickson, Aryeh Warmflash, and Aaron R. Dinner<sup>a)</sup>

*James Franck Institute, The University of Chicago, Chicago, Illinois 60637, USA*

(Received 26 June 2009; accepted 16 September 2009; published online 15 October 2009)

Umbrella sampling enforces uniform sampling of steady-state distributions that are functions of arbitrary numbers of order parameters. The key to applying such methods to nonequilibrium processes is the accumulation of fluxes between regions. A significant difference between microscopically reversible and irreversible systems is that, in the latter case, the transition path ensemble for a reaction can be significantly different for “forward” and “backward” trajectories. Here, we show how to separately treat forward and backward pathways in nonequilibrium umbrella sampling simulations by working in an extended space. In this extended space, the exact rate (for equilibrium or nonequilibrium processes) can be calculated “for free” as a flux in phase space. We compare the efficiency of this rate calculation with forward flux sampling for a two-dimensional potential and show that nonequilibrium umbrella sampling is more efficient when an intermediate is present. We show that this technique can also be used to describe steady-state limit cycles by examining a simulation of circadian oscillations. We obtain the path of the limit cycle in a space of 22 order parameters, as well as the oscillation period. The relation of our method to others is discussed. © 2009 American Institute of Physics. [doi:10.1063/1.3244561]

## I. INTRODUCTION

Systems far from equilibrium are of major interest,<sup>1,2</sup> and there have been significant advances in our ability to describe them recently. In particular, systems with well-defined steady states are now known to be constrained by certain symmetry relations.<sup>3</sup> For such systems, the steady-state distribution is analogous to the free energy in that any quantity of interest can be calculated from it. However, like the free energy, the steady-state distribution can be slow to converge in simulations owing to minima and barriers that hinder exploration of phase space. In such cases, it is useful to divide the space into regions that can be treated independently in parallel (umbrella sampling).<sup>4–6</sup> Some of the features that make irreversible systems interesting also present challenges for umbrella sampling. One must account explicitly for the fact that the flux across any boundary introduced is nonzero because detailed balance is not satisfied. For the same reason, the transition path ensemble that connects two stable states can be significantly different for “forward” and “backward” trajectories. In other words, the physically weighted ensemble of paths from *A* to *B* in general differs from that of paths from *B* to *A*.

Motivated by the string method for equilibrium systems,<sup>7–13</sup> we previously introduced a version of our nonequilibrium umbrella sampling algorithm<sup>5</sup> that focuses sampling on a tube of transition paths that connect two basins in a phase space of arbitrary dimension.<sup>6</sup> The method differs from other interface-based methods such as milestoneing,<sup>14–17</sup> forward flux sampling (FFS),<sup>18–22</sup> and (partial-path) transition interface sampling,<sup>23–25</sup> in that the set of interfaces

evolves over the course of the simulation from an initial guess toward its final configuration. This is important as the choice of reaction coordinate (or equivalently, the set of interfaces orthogonal to it) is often nonobvious and strongly affects the efficiency of a simulation.<sup>26</sup> However, as discussed by Dickson *et al.*,<sup>6</sup> successful application of the stringlike version of nonequilibrium umbrella sampling requires significant overlap between the forward and backward trajectories such that a single tube in the space of order parameters could contain them.

Here we extend this method<sup>6</sup> to treat forward and backward trajectories separately by defining two strings, each with an associated direction. We show how this procedure is equivalent to performing a single nonequilibrium umbrella sampling simulation in an extended phase space. In this extended phase space, we can obtain the forward and backward transition rates “for free” as a flux. We also show that this technique can be used to describe steady-state limit cycles. In Sec. II, we briefly outline the details of the nonequilibrium umbrella sampling method, introduce the dual-string formalism, and show how to obtain transition rates. We also describe a weight-balancing scheme that can accelerate convergence. We apply this method in Sec. III A to a two-dimensional potential with distinct forward and backward transition pathways and calculate rates and steady-state distributions. We compare the efficiency of obtaining these results with that of FFS on the same system. Both algorithms are able to calculate the rate, but in the presence of a long-lived intermediate, the umbrella sampling algorithm converges faster. In Sec. III B we illustrate the ability of our method to describe a limit cycle in a multidimensional order parameter space and determine its period by applying it to a model of circadian oscillations.<sup>27,28</sup>

<sup>a)</sup>Electronic mail: dinner@uchicago.edu.

## II. METHODS

The nonequilibrium umbrella sampling method is described in detail elsewhere.<sup>5,6</sup> Essentially, the order parameter space is divided into regions and a separate simulation (a “walker”) is performed in each region. To perform these piecewise simulations accurately in a nonequilibrium system, two quantities are needed for each region: the weight (the probability of an unconstrained walker being in that region) and the entering flux distribution (the probability density of trajectories entering into the region). Both quantities are obtained by monitoring attempted region-to-region boundary crossings of the walkers. When a walker tries to leave its region, it is reinitialized according to the region’s flux distribution. One way to establish these distributions (the technique that is used here) is to approximate them using lists of saved entry points of length  $N_{\text{list}}$  and to continually refresh them by overwriting the oldest entries.<sup>6</sup>

We refer to the set of regions into which the order parameter space is divided as a (regular or irregular) “lattice.” We found that using two lattices that are shifted relative to each other enhances the stability of the algorithm. Simulations on each lattice are used to record points for the fluxes on the other lattice. The weights of the regions are updated according to methods given in Sec. II C and the steady-state distribution is obtained once the weights and fluxes converge self-consistently. There is no computational cost associated with having two lattices rather than one as the results are pooled at the end of the simulation.

### A. Dual-direction scheme

In our original algorithm,<sup>5</sup> the regions used to confine the walkers are static and evenly tile an order parameter space, such that the computational cost of the algorithm increases exponentially with the dimensionality of the space. In Voronoi-based nonequilibrium umbrella sampling,<sup>6</sup> this dependence is removed by instead using Voronoi polyhedra as regions. Each Voronoi polyhedron is defined by a point in phase space (an “image”), such that every point in the polyhedron is closer to that image than to any other. The images together form a “string” that connects two areas of interest in phase space. By defining the regions as above, the computational cost of the simulation is decoupled from the dimensionality of the order parameter space. Initially, the images are placed in an arbitrary position (usually a linear interpolation between initial and final states) and over the course of the simulation the images evolve until the path stabilizes inside the reaction tube. The movement of the images based on the sampling<sup>5,6</sup> is according to the finite-temperature string method.<sup>7–13</sup> Namely, after a specified length of time ( $T_{\text{move}}$ ), the average position of each walker is determined, and the positions of the images ( $z_i$ ) of the first string are moved toward their walker averages:

$$z_i^\dagger = z_i^\circ + \tau(\langle \theta(x_i) \rangle - z_i^\circ), \quad (1)$$

where  $z_i^\circ$  is the original position of image  $i$ ,  $\tau$  is a parameter between 0 and 1 controlling the rate of the image movement,  $\theta(x)$  returns the position in the order parameter space given  $x$  in the full space, and  $\langle \theta(x_i) \rangle$  is the average position of walker

$i$  during the last time interval. The string is then smoothed according to

$$z_i^\ddagger = z_i^\dagger + \kappa(z_{i+1}^\dagger + z_{i-1}^\dagger - 2z_i^\dagger), \quad (2)$$

where  $\kappa$  is a positive parameter that controls the strength of the smoothing [note that, while Eq. (2) was employed by Dickson *et al.*,<sup>6</sup> the first  $z_i^\dagger$  on the right hand side was incorrectly omitted from Eq. (7) of that paper]. Finally, the string undergoes a reparametrization step such that the images are moved from  $z_i^\ddagger$  to positions spaced equally in arc length to ensure even sampling along the path. For our purposes, Eqs. (1) and (2) are used to move the images in the first lattice, and the second lattice images are placed at the midpoints of the images of the first. More information about this procedure can be found in Vanden-Eijnden and Venturoli<sup>13</sup> and Dickson *et al.*<sup>6</sup>

In the present study, we perform the umbrella sampling algorithm as above, while explicitly defining separate strings for the forward and backward transition path ensembles. This is done by defining two basins,  $A$  and  $B$ , that are independent of the Voronoi sampling regions, and separating trajectories that last visited  $A$  from those that last visited  $B$ . Let  $\mathbb{R}^m$  be the normal  $m$ -dimensional order parameter space, and let the trajectory  $\{\mathbf{x}(t)\}$  have the additional property that  $x_{m+1}(t)=0$  if the trajectory last visited  $A$ , and  $x_{m+1}(t)=1$  if the trajectory last visited  $B$ . Trajectories can now be defined in an extended phase space  $\mathbb{R}^{m+} = \mathbb{R}^m \cup \{0, 1\}$ . Figure 1 shows an example of a set of images spanning  $\mathbb{R}^{m+}$ . To employ the Voronoi-based procedure, we define the distance between two points  $\mathbf{x}$  and  $\mathbf{y}$  in  $\mathbb{R}^{m+}$  as follows:

$$\|\mathbf{x} - \mathbf{y}\| = \begin{cases} \infty & \text{if } x_{m+1} \neq y_{m+1}, \\ \sqrt{\sum_{i=1}^m (x_i - y_i)^2} & \text{otherwise,} \end{cases} \quad (3)$$

such that points that originated from different basins are infinitely far apart. This way, all Voronoi regions are either wholly in  $\mathbb{R}^m \cup 0$  or  $\mathbb{R}^m \cup 1$ . We denote the set of Voronoi regions in  $\mathbb{R}^m \cup 0(\mathbb{R}^m \cup 1)$  by  $\mathcal{S}_A(\mathcal{S}_B)$ . Using the theory above, two forward and backward pathways in  $\mathbb{R}^m$  can be viewed as a single pathway in  $\mathbb{R}^{m+}$ , and since boundary crossings occur in the same way as before, no other modifications to the algorithm are needed.

### B. Obtaining rates

As previously noted in Warmflash *et al.*<sup>5</sup> and Dickson *et al.*,<sup>6</sup> one should be able to obtain kinetic quantities from nonequilibrium umbrella sampling simulations because they harvest segments of true trajectories piecewise and thus contain information about dynamics. Vanden-Eijnden and Venturoli<sup>29</sup> recently enunciated a specific procedure based on nonequilibrium umbrella sampling to calculate the rate of a (equilibrium or nonequilibrium) process. Here we present an equivalent but more compact derivation of the key expression for clarity and completeness.

The rate of transition from a basin  $A$  to a basin  $B$  is, by definition, given by the flux of trajectories that originated in  $A$  and entered  $B$ :<sup>18</sup>

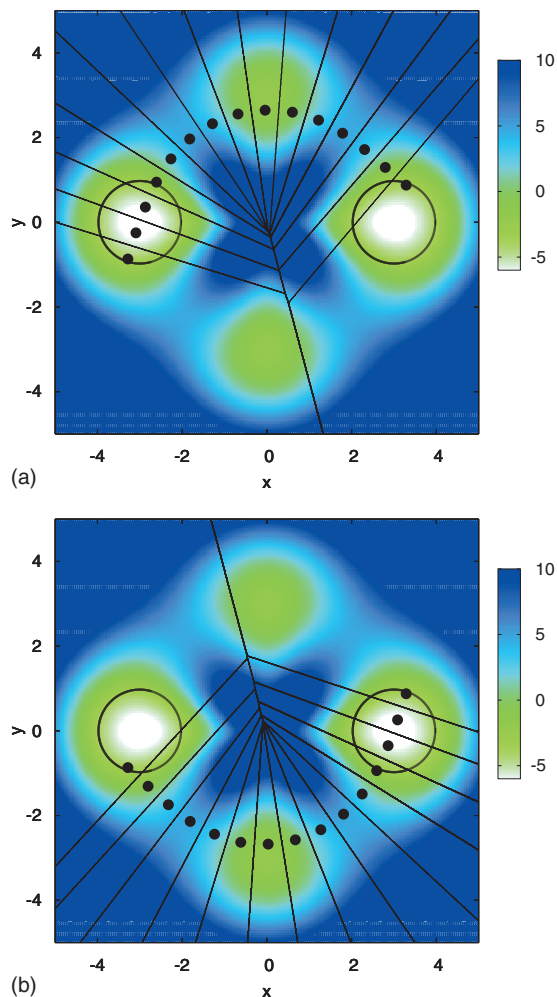


FIG. 1. The two-dimensional potential with two pathways created by a strong force in the clockwise direction. The top and bottom plots show the images and the Voronoi polyhedra for a converged forward and backward string, respectively. The black circles centered at  $(-3,0)$  and  $(3,0)$  are the basins. The scale bar is in units of  $kT$  and the parameters used to generate the map are those given in the text with  $\beta=3.0$ .

$$k_{AB} = \frac{\bar{\Phi}_{B|S_A}}{\bar{h}_A}, \quad (4)$$

where  $\Phi_{B|S_A}$  is the flux into basin  $B$  of trajectories that originated in basin  $A$ , and  $h_A$  is a history-dependent indicator function that is equal to 1 (0) if the system was most recently in basin  $A$  ( $B$ ). Overbars denote time averages. Since the algorithm described above now computes fluxes separately for trajectories that originated in each basin,  $\bar{\Phi}_{B|S_A}$  can be calculated directly.

$\bar{\Phi}_{B|S_A}$  is obtained from the umbrella sampling simulation by measuring the average flux contributions into basin  $B$  from all regions  $j$  in  $S_A$  (denoted  $\bar{\phi}_{jB}$ ). We then have an expression for the rate constant for each lattice  $a$ :

$$k_{AB}^{(a)} = \frac{\sum_{j \in S_A^{(a)}} \bar{\phi}_{jB}}{\sum_{j \in S_A^{(a)}} \bar{W}_j} = \frac{\sum_{j \in S_A^{(a)}} n_{jB} \bar{W}_j / \Delta T_j}{\sum_{j \in S_A^{(a)}} \bar{W}_j}, \quad (5)$$

where  $n_{jB}$  is the number of trajectories from region  $j$  entering basin  $B$  over a given sampling period, and  $\Delta T_j$  is the time

elapsed in region  $j$  during that sampling period.  $\bar{W}_j$  is the time-averaged weight of region  $j$ , and we used the fact that

$$\bar{h}_A = \sum_{j \in S_A^{(a)}} \bar{W}_j. \quad (6)$$

In practice,  $k_{AB}^{(1)}$  and  $k_{AB}^{(2)}$  are calculated separately and then averaged.

### C. Global weight transfer scheme

In our previous work,<sup>5,6</sup> the weights evolved over the course of the simulation by transferring incremental amounts from one region to another whenever walkers attempted to leave their regions:

$$-\Delta W_B = \Delta W_{B'} = s W_B T^* / T_B, \quad (7)$$

where  $B$  labels the region to which the walker is confined,  $B'$  labels the region to which it attempted to go,  $\Delta$  denotes an additive change, and  $s$  is a user-defined parameter chosen to optimize convergence. The factor  $T^*/T_B$  allows the simulations in different regions to be run asynchronously;  $T_B$  is the time elapsed in region  $B$ , and  $T^*$  is an arbitrary standard that in practice is chosen to be the time elapsed in a particular region of the system. We use a time-averaged weight (denoted  $\bar{W}_B$ ) to compute the fluxes:<sup>5,6</sup>

$$\bar{W}_B = \frac{1}{n} \sum_{i=1}^n W_B(i T_{\text{wt}} / n + T_{\text{int}}), \quad (8)$$

where  $W_B(t)$  is the instantaneous weight of that region at time  $t$ .  $T_{\text{wt}}$  is the weight update interval,  $n$  is an arbitrary discretization constant set here to 100, and  $T_{\text{int}}$  denotes the time at the beginning of that sampling interval.

Although this “local” weight balancing scheme is appealing in its simplicity, it is natural in the context of obtaining steady-states to consider a “global” weight balancing scheme in which the weights of all regions are updated simultaneously based on the fluxes. To this end, consider two regions  $j$  and  $k$ . The flux from  $j$  into  $k$  can be measured in two ways: the number of trajectories in region  $j$  that exit into  $k$  can be counted and weighted by  $W_j$ ; additionally, the number of trajectories that are initialized in region  $k$  that originated in  $j$  can be counted and weighted by  $W_k$ . Here we match these two fluxes to get an equality involving the two weights. Note that this is not the same as detailed balance; it only ensures continuity of the trajectory flux across the boundaries.

The physically weighted flux from  $j$  to  $k$ , as measured in region  $k$ , is given by

$$(\text{flux})_{jk}^k = W_k^{(a)} \bar{F}_{kj}^{(a)}, \quad (9)$$

where  $\bar{F}_{kj}^{(a)}$  is the average number of trajectories in region  $k$  being initialized at the  $j:k$  interface per unit time.  $\bar{F}_{kj}^{(a)}$  is calculated using the flux list for region  $k$ :



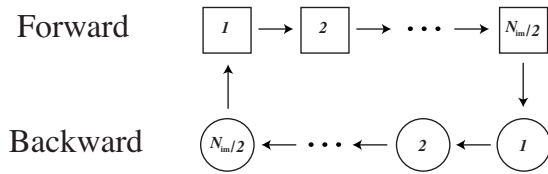


FIG. 2. A schematic of the “principal loop” used for the global weight balancing scheme. An arrow from region  $j$  to region  $k$  indicates that the flux balance Eq. (12) between those two regions is used to form the matrix  $\mathbf{M}$ .

$$\bar{F}_{kj}^{(a)} = \frac{N_k^{(a)}}{T_k^{(a)}} \left( \frac{\sum_{l \in \mathbb{F}_k^{(a)}} W_{b_k(l)}^{(b)} h_{kj}^{(a)}(l)}{\sum_{l \in \mathbb{F}_k^{(a)}} W_{b_k(l)}^{(b)}} \right), \quad (10)$$

where the quantity in parentheses is the weighted fraction of the flux list for region  $k$  of trajectories coming from region  $j$ .  $N_k^{(a)}$  is the total number of trajectory segments that have been initialized in region  $k$  during the time  $T_k^{(a)}$ ,  $b_k(l)$  is a function that returns the region in which crossing point  $l$  was recorded,  $h_{kj}^{(a)}(l)$  is an indicator function equal to 1 if crossing point  $l$  came from region  $j$  and 0 otherwise, and  $a, b = 1, 2$  or  $2, 1$ .  $\mathbb{F}_k^{(a)}$  is the flux list for region  $k$  on lattice  $a$ .

Since the weights and fluxes evolve from an inaccurate initial guess to their steady-state quantities, all quantities that are dependent on the weights and fluxes during this evolution are also inaccurate. The motivation behind this weight transfer scheme is to use quantities that are independent of the weights and fluxes, so they can be accurately determined early in a simulation. Consider the conditional probability  $p_{ijk}^{(a)}$ : the probability (measured in region  $j$ ) of a trajectory exiting via the  $j:k$  interface, given that it originated on the  $i:j$  interface (all regions are on lattice  $a$ ). This quantity is relatively insensitive to errors in the weights and fluxes.

Using this probability, the quantity in Eq. (9) as computed in region  $j$  can be expressed as

$$(\text{flux})_{jk}^j = W_j^{(a)} \sum_i p_{ijk}^{(a)} \bar{F}_{ji}^{(a)}, \quad (11)$$

where the probability  $p_{ijk}^{(a)}$  is estimated directly during the simulation. The sum is required in Eq. (11) since trajectories that originated on any point on the boundary of region  $j$  can contribute to the flux into region  $k$ . Equations (9) and (11) yield a constraint

$$W_j \sum_i p_{ijk} \bar{F}_{ji} - W_k \bar{F}_{kj} = 0 \quad (12)$$

between each  $j:k$  pair where transitions occur. Here, we dropped the lattice indices since all quantities are on the same lattice.

Since there are many more available equations than unknowns, we are free to choose the set of equations that involve the most transitions, and hence have the most accurate statistics for the  $p$  and  $\bar{F}$  quantities. We choose transitions between regions around the “principal loop” (Fig. 2), those passing sequentially through the images of the forward and backward strings, and connecting both strings at the end points. We make this restriction to limit large weight fluctuations caused by occasional transitions from high probability regions to low probability regions.

A matrix ( $\mathbf{M}$ ) is then constructed coupling  $N_{\text{im}}$  equations of the form in Eq. (12) as follows:

$$M_{ji} = \begin{cases} \sum_l p_{ljk} \bar{F}_{jl} & \text{if } i = j, \\ -\bar{F}_{kj} & \text{if } i = k, \\ 0 & \text{otherwise,} \end{cases} \quad (13)$$

where  $k$  is the region following  $j$  in the principal loop, and  $N_{\text{im}}$  is the total number of regions on the lattice. To insert the normalization condition (and to relieve the overdetermination) a row ( $i$ ) is chosen and its elements are replaced with 1 to form the matrix  $\mathbf{M}_{\text{norm}}$ . The weights are then determined by

$$\hat{W}^{(a)} = (\mathbf{M}_{\text{norm}}^{(a)}(i))^{-1} \hat{e}_i \quad (14)$$

for each lattice  $a$ , where  $\hat{e}_i$  is the vector with its  $i$ th entry equal to 1, and all others equal to 0.

For each weight update step this process is repeated once for each region on the lattice using all possible values of  $i$ , and then the weights are moved toward the average value obtained:

$$\hat{W}_{\text{final}}^{(a)} = (1 - \tau_W) \hat{W}_{\text{initial}}^{(a)} + \frac{\tau_W}{N_{\text{im}}^{(a)}} \left( \sum_{i=1}^{N_{\text{im}}^{(a)}} (\mathbf{M}_{\text{norm}}^{(a)}(i))^{-1} \hat{e}_i \right), \quad (15)$$

where  $\hat{W}_{\text{initial}}^{(a)}$  is the weight vector before the update. Note that  $N_{\text{im}}^{(2)} = N_{\text{im}}^{(1)} - 2$  since the forward and backward strings have one less image each. Since this process requires that there be transitions from the last region of the forward (backward) string to the first region of the backward (forward) string, we keep the positions of the string end point images equal by moving them to their joint averages, as opposed to their individual averages as shown in Eq. (1). In other words,

$$z_{1f}^\dagger = z_{(N_{\text{im}}/2)b}^\dagger = z_{1f}^\circ + \tau \left[ \frac{\langle \theta(x_{1f}) \rangle + \langle \theta(x_{(N_{\text{im}}/2)b}) \rangle}{2} - z_{1f}^\circ \right], \quad (16)$$

where  $z_{1f}$  denotes the first image in the forward string. This is done in the same fashion for the  $(z_{(N_{\text{im}}/2)f}, z_{1b})$  pair of end points. In practice, we found that the local weight transfer scheme achieves higher accuracies, but that the global scheme is more efficient early in a simulation; both procedures are used in this work. As noted in Sec. IV, Vanden-Eijnden and Venturoli introduced an alternative global weight transfer scheme.<sup>29</sup>

## D. Forward flux sampling

Here we state the important operational features of FFS in the notation used above, as well as some aspects specific to our implementation. Further information can be found in previous work.<sup>18–21</sup> FFS computes the rate using Eq. (4), but instead of calculating  $\bar{\Phi}_{B|S_A}$  directly, the space between the basins  $A$  and  $B$  is divided by a series of nonintersecting interfaces ( $\lambda_i$ ), and the flux is computed using the product of conditional probabilities:

$$\bar{\Phi}_{B|S_A} = \bar{\Phi}_{0|S_A} P(\lambda_B|\lambda_0) = \bar{\Phi}_{0|S_A} \prod_{i=0}^{n-1} P(\lambda_{i+1}|\lambda_i), \quad (17)$$

where  $\bar{\Phi}_{0|S_A}$  is the time-averaged flux reaching interface  $\lambda_0$  measured in  $S_A$ , and  $P(\lambda_{i+1}|\lambda_i)$  is the probability of reaching interface  $\lambda_{i+1}$  from interface  $\lambda_i$  before returning to the first interface,  $\lambda_0$  (the boundary of basin A). Here,  $\lambda_n = \lambda_B$  is the boundary of basin B.

It has been shown previously by Valeriani *et al.*<sup>21</sup> that steady-state distributions can be obtained by combining the results from separate forward and backward FFS simulations. The probability distribution (over an order parameter  $q$ ) between the interfaces can be resolved by accumulating the density at  $q$  of trajectories leaving interface  $\lambda_i$  in the forward set of simulations (denoted by “+”):

$$\pi_+(q|\lambda_i) = \frac{N_q}{\Delta q M_i}. \quad (18)$$

Here,  $N_q$  is the number of times the order parameter has a value between  $q$  and  $\Delta q$  over all the trial runs, and  $M_i$  is the number of trial runs. The probability distribution conditional on starting at  $\lambda_0$  is then

$$\tau_+(q|\lambda_0) = \pi_+(q|\lambda_0) + \sum_{i=1}^{n-1} \pi_+(q|\lambda_i) \prod_{j=0}^{i-1} P(\lambda_{j+1}|\lambda_j) \quad (19)$$

(the variable  $\tau_+$  is chosen to be congruent with the original notation in Valeriani *et al.*<sup>21</sup> and is not related to the movement fractions  $\tau$  or  $\tau_W$  mentioned earlier). Once these quantities are obtained in the backward simulation as well, the full steady-state probability distribution can be obtained,

$$p(q) = p_A \Phi_{0|S_A} \tau_+(q|\lambda_0) + p_B \Phi_{n|S_B} \tau_-(q|\lambda_n), \quad (20)$$

where  $p_A$  and  $p_B$  are the steady-state probabilities of basins A and B, respectively. This method yields the steady-state distribution between  $\lambda_0$  and  $\lambda_n$ . The contributions from the basins can be computed in separate physically weighted simulations and incorporated using a least-squares fitting procedure.<sup>21</sup> Note that Eq. (20) assumes that there are no intermediates in the system (i.e.,  $p_A + p_B = 1$ ). To compare algorithms in the present study, we generalize Eq. (20) by replacing  $p_A$  and  $p_B$  with  $\bar{h}_A$  and  $\bar{h}_B$ , respectively. The latter pair of quantities can be obtained in a FFS simulation using  $\bar{h}_A k_{AB} = \bar{h}_B k_{BA}$  together with  $\bar{h}_A + \bar{h}_B = 1$ , both of which hold in general.<sup>30</sup>

In previous applications of FFS, the number of trial simulations was set manually for each interface to ensure good statistics and efficient calculation.<sup>18,19,21,22</sup> A goal of the present study is to examine the accuracies of our algorithm and FFS as functions of simulation time. Because simulation time can be allocated in different ways in FFS, a fair assessment requires running many FFS simulations of controlled length. We take the approach of Sear<sup>31</sup> and fix the number of simulations that reach the next interface ( $N_p$ ) rather than the total number of trajectories launched from each interface. We then run FFS simulations with different values of  $N_p$  and determine their accuracy, while keeping track of the total number of time steps needed to obtain the required number

of accepted trajectories. The efficiency is obtained by examining the accuracy as a function of the total number of time steps. A procedure has been introduced to optimize the number of trial trajectories for each interface with respect to the computational cost of the algorithm;<sup>32</sup> we expect that this does provide a boost in efficiency, but it is not implemented here.

### III. EXAMPLES

#### A. Two-dimensional system with two pathways

We first examine a two-dimensional system with two pathways. The potential surface is defined by

$$V = \alpha(r - \gamma)^2 - \chi_1 \cos(2\theta) - \chi_2 \cos(4\theta), \quad (21)$$

where  $r = (x^2 + y^2)^{1/2}$ , and  $\theta$  is the angle in radians measured counterclockwise from the  $x$  axis. The potential surface has a ring-shaped valley of radius  $\gamma$  with basins at  $\theta = \pi/2, \pi, 3\pi/2$ , and  $2\pi$ ;  $\alpha$  controls the depth of the ring, and  $\chi_1$  and  $\chi_2$  control the depth of the basins. We drive the system out of equilibrium using a constant external force in the  $-\theta$  (clockwise) direction ( $\mathbf{F}_{\text{ext}} = -F\hat{\theta}$ ). The force creates distinct forward ( $\pi \rightarrow 0$ ) and backward ( $0 \rightarrow \pi$ ) pathways (Fig. 1).

The system evolves according to the equation of motion,

$$\mathbf{X}(t + \delta t) = \mathbf{X}(t) - \frac{\delta t}{m\xi} (\nabla_{\mathbf{X}} V - \mathbf{F}_{\text{ext}}) + \delta \mathbf{X}^G, \quad (22)$$

where  $\delta t$  is the numerical integration time step,  $\delta \mathbf{X}^G$  is a random noise term with components chosen randomly from a Gaussian distribution with zero mean and variance  $2D\delta t$ , and  $\mathbf{X} = (x, y)$ .  $D$  is the diffusion coefficient, which is related to the friction coefficient  $\xi$  and the inverse temperature  $\beta$  by  $D = (m\beta\xi)^{-1}$ .

The system parameters used here are  $\alpha = 3.0$ ,  $\gamma = 3.0$ ,  $\chi_1 = 2.25$ ,  $\chi_2 = 4.5$ ,  $\xi = 1.5$ ,  $F = 7.2$ , and  $\delta t = 0.005$ . The inverse temperatures used ranged from  $\beta = 1.0$  to  $\beta = 3.0$ . Basins A and B are defined as circles of radius  $R = 1.0$  centered at  $(-\gamma, 0)$  and  $(\gamma, 0)$ , respectively. There are a number of different parameters associated with the umbrella sampling algorithm:  $T$  is the total time of the simulation for each region,  $T_{\text{move}}$  is the string update period,  $N_{\text{list}}$  is the size of the flux list,  $N_{\text{im}}$  is the number of images used for all lattice 1 images (there are  $2N_{\text{im}} - 2$  images total),  $T_{\text{wt}}$  is the weight update period,  $s$  is the weight transfer parameter given in Eq. (7),  $\tau$  and  $\kappa$  are the string parameters given in Eqs. (1) and (2), and  $\tau_W$  is the weight update parameter defined in Eq. (15). The values of these parameters are given in Table I for the different simulation phases and temperatures.

For the FFS simulations, the interfaces  $\lambda_0$  and  $\lambda_n$  were chosen as the basin boundaries for A and B, respectively, to easily compare the results with those obtained from umbrella sampling. Interfaces 1 through  $n-1$  were vertical lines spaced evenly from  $-\gamma + R + \delta$  to  $\gamma - R - \delta$ , where  $\delta = 0.01$  to prevent the circular and linear interfaces from intersecting. In the FFS simulations, a fixed number ( $N_p$ ) of crossing points were obtained on each interface. Ten FFS simulations were run for each of nine different values of  $N_p = 100, 200, 500, 1000, 2000, 5000, 10\,000, 20\,000, 50\,000$ . For each value of

TABLE I. Parameters used in the umbrella sampling algorithm for the two-dimensional potential. In phase I, the images are free to move; their positions are fixed for phases II and III. Phase II is run with the global weight transfer algorithm until the rate converges, and phase III uses the local weight transfer algorithm to determine the rates and steady-state distributions more accurately.

	$\beta=1.5$			$\beta=3.0$		
	Phase I	Phase II	Phase III	Phase I	Phase II	Phase III
Wt. Alg.	Global	Global	Local	Global	Global	Local
$T/\delta t$	$4 \times 10^5$	$1 \times 10^6$	$2 \times 10^7$	$4 \times 10^5$	$2 \times 10^7$	$2 \times 10^8$
$T_{\text{move}}/\delta t$	10 000	$\infty$	$\infty$	10 000	$\infty$	$\infty$
$N_{\text{list}}$	100	500	500	100	1000	500
$N_{\text{im}}$	15	15	15	30	30	30
$T_{\text{wt}}/\delta t$	100 000	20 000	10 000	100 000	100 000	10 000
$s$	...	...	$5 \times 10^{-3}$	...	...	$1 \times 10^{-2}$
$\tau$	0.1	...	...	0.1	...	...
$\tau_W$	1.0	1.0	...	1.0	1.0	...
$\kappa$	0.1	...	...	0.3	...	...

$N_p$  the total number of time steps taken in each simulation is averaged, and this average is used to estimate the error as a function of simulation time.

The projections of the steady-state distribution onto the  $\theta$  coordinate are obtained using conventional (CONV), non-equilibrium umbrella sampling (UMB), and FFS simulations for the high-temperature value ( $\beta=1.5$ ), and are shown in Fig. 3. Since the FFS distributions do not include the basin contributions, we compare them with the others by normalizing over the range of  $\theta$  that is external to the basins. The projections all agree, and they clearly show the locations of the basins, intermediates, and transition states.

We examine the forward and backward rates, as calculated from Eq. (4). The error in a given simulation is taken to be

$$\text{error}(t) = |\log(k(t)) - \log(k^*)|, \quad (23)$$

where  $k^*$  is the target rate constant. This choice for the error identifies both overestimates and underestimates. For  $\beta=1.5$  the target rate constant is obtained by averaging the rate obtained from ten long, independent conventional simulations. The target rate for  $\beta=3.0$  was obtained by fitting the rate as

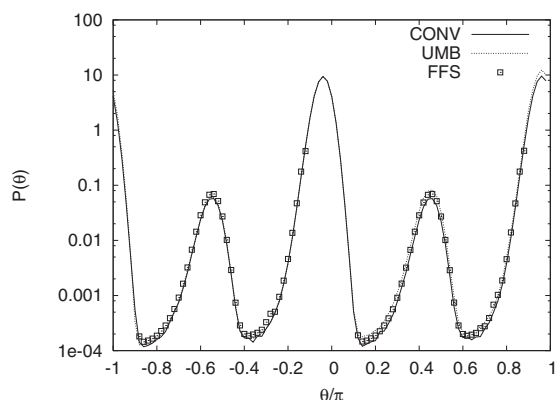


FIG. 3. Projections of the steady-state distribution onto  $\theta$  for  $\beta=1.5$  obtained using conventional simulation (CONV), umbrella sampling (UMB), and FFS. The two highest peaks are due to the basins, located at  $\theta=0$  and  $\pi$ , and the two smaller peaks are due to the intermediates, located at  $\theta=-\pi/2$  and  $\pi/2$ .

a function of temperature to an Arrhenius form and extrapolating (Fig. 4). For each method, the errors plotted in Fig. 5 are the root mean squared averages of errors obtained from ten simulations. For the umbrella sampling algorithm, the results are a rolling average over the last five points in the curve. This both excludes early inaccurate results from the latter parts of the curve and lessens the effects of noise.

Figure 5 compares the efficiency of FFS, umbrella sampling and conventional sampling for  $\beta=1.5$  and  $\beta=3.0$ . For  $\beta=1.5$ , transitions from left to right are common, and conventional sampling is most efficient. The umbrella sampling algorithm is less efficient than conventional sampling for this inverse temperature because the time it takes for the weights and fluxes to converge is long compared to the mean first passage time. FFS does not outperform conventional sampling in this case due to the presence of the intermediates.

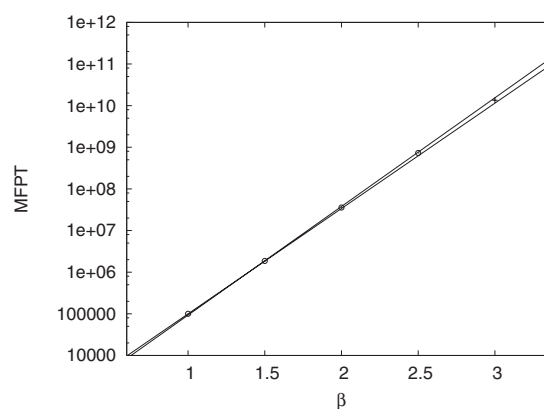


FIG. 4. The target mean first passage time (MFPT) of  $1.35 \pm 0.20 \times 10^{11}$  (right-most point) is obtained by extrapolation of a linear fit. The logarithms of MFPTs obtained from straightforward simulations for  $\beta=1.0, 1.5, 2.0, 2.5$  (circles) are fit linearly assuming an Arrhenius dependence of the rate on the temperature. The error bars are equal to, or smaller than, the symbols for all points. The uncertainty in the target value was estimated using the fits shown in the figure, which were chosen to maximize and minimize the slope of the linear fit. This was done using the mean minus (plus) one standard deviation for  $\beta=1.0, 1.5$  and the mean plus (minus) one standard deviation for  $\beta=2.0, 2.5$  for the first (second) line. In this way, we estimate that the  $\beta=3.0$  target is accurate up to  $\pm 2.0 \times 10^{10}$ , and that the efficiency curves in the bottom of Fig. 5 are accurate above an error of 0.07.

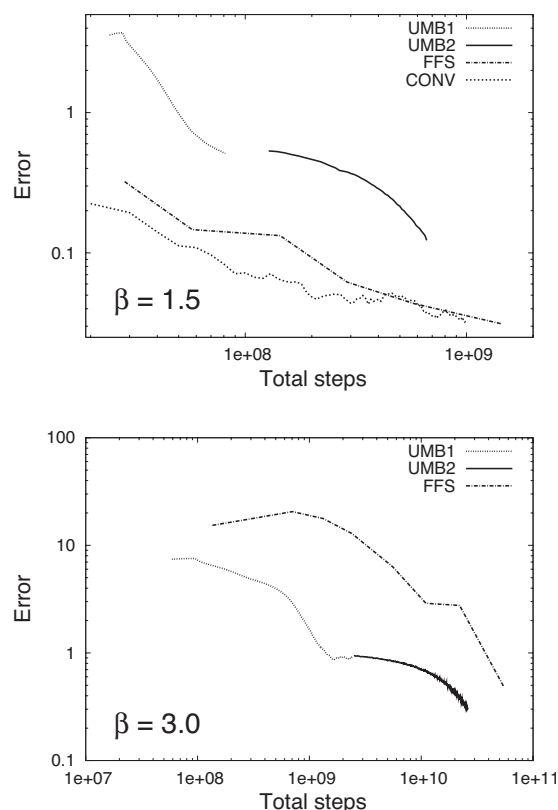


FIG. 5. Convergence of the rate constant for the two-dimensional potential with intermediates. (top) Umbrella sampling (UMB), conventional sampling (CONV), and FFS for  $\beta=1.5$ . At this temperature, transitions are common and conventional sampling is shown to be the most efficient. (bottom) Umbrella sampling and FFS for  $\beta=3.0$ . At this temperature, a single transition in each direction requires about  $2.7 \times 10^{11}$  simulation steps in conventional simulations, limiting the comparison to umbrella sampling and FFS. In both figures, UMB1 shows the global weight balancing step and UMB2 shows the local weight balancing step. For the FFS efficiency curves, each point is a separate group of simulations with a different value of  $N_p$ . The gap between the UMB1 and UMB2 curves in the top figure is the result of the omission (for clarity) of three points that were uncharacteristically low as the flux list was being rebuilt.

The bottom of Fig. 5 compares FFS and umbrella sampling for  $\beta=3.0$ . FFS takes from 3 to 30 times longer than umbrella sampling to converge to a given accuracy. For this inverse temperature, it was unfeasible to determine the rate accurately with conventional simulations.

A summary of the performance of FFS and umbrella sampling for different values of  $\beta$  is given in Fig. 6. The quantity on the y axis is  $T_1/\text{MFPT}$ , where  $T_1$  is the number of sampling steps required to reach an error of 1, which corresponds to an order-of-magnitude estimate of the rate constant. We divide  $T_1$  by the mean first passage time (in numbers of steps) to compare with conventional simulations. The line  $T_1/\text{MFPT}=1$  is plotted as it marks the point where obtaining an order-of-magnitude approximation of the rate requires about the same number of steps as it would take in a conventional simulation.

Figure 6 shows the main result of our work: for high temperatures (low  $\beta$ ) transitions are frequent and both enhanced sampling methods are less efficient than conventional sampling. As the temperature decreases, the efficiency of the umbrella sampling algorithm improves dramatically, while

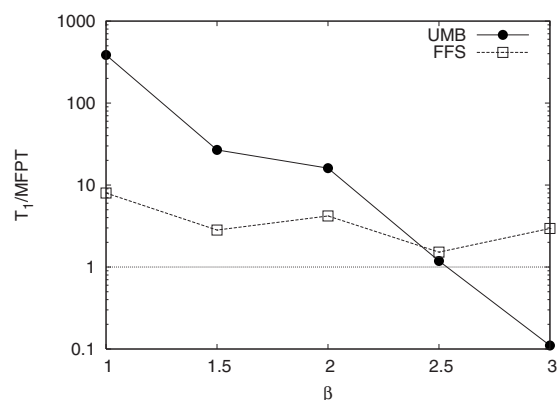


FIG. 6. A summary of the performance of umbrella sampling (UMB) and FFS for a range of inverse temperatures ( $\beta$ ). The y axis shows  $T_1$ , the number of time steps required to achieve an error of 1 (roughly an order-of-magnitude estimate of the rate constant) divided by the mean first passage time.

FFS is hindered by the presence of the intermediate. Similar studies were done on a system without an intermediate and FFS was found to be more efficient than both umbrella and conventional sampling (not shown); this is the case even when the umbrella sampling interfaces are chosen to be the same as those for FFS owing to the simulation time required for the weights to converge.

## B. Simulation of circadian oscillations

Circadian rhythms are sustained oscillations generated by most living organisms to adapt to the natural periodicity of their terrestrial environment. These oscillations are generated autonomously through negative autoregulatory feedback of gene expression, and occur with a period close to 24 h even in conditions of constant darkness.<sup>33</sup> The mechanism of these oscillations can be broken down into linear chemical equations and simulated using the Gillespie algorithm.<sup>34</sup> Here, we apply the Voronoi-based nonequilibrium umbrella sampling procedure to a model for circadian oscillations developed by Gonze *et al.*<sup>27,28</sup> to demonstrate the ability of the algorithm to find a periodic path in a space of order parameters and determine the associated oscillation frequency.

The model used here employs five main variables and is based on the negative feedback exerted by a protein ( $P$ ) on the expression of its gene ( $G$ ). The gene is expressed in the nucleus and then transcribed into messenger RNA ( $M_P$ ).  $M_P$  is then transported to the cytosol where it is degraded and translated into the protein  $P_0$ . The protein undergoes catalyzed phosphorylation and dephosphorylation in multiple steps:  $P_0 \rightarrow P_1 \rightarrow P_2$ .  $P_2$  is then either marked for degradation or reversibly transported into the nucleus ( $P_2 \rightarrow P_N$ ). The nuclear form of the protein represses the transcription of the gene, completing the cycle. The kinetic expressions describing the above system contain nonlinear terms: the enzyme reaction rates are described by Michaelis–Menten expressions, and the successive binding of repressors to the gene promoter is described by Hill type expressions. These have been converted into a larger system of 30 elementary reactions involving 22 species that are summarized in Table II of



TABLE II. Parameters used in the umbrella sampling algorithm for the circadian oscillations.  $T$ ,  $T_{\text{move}}$ , and  $T_{\text{wt}}$  are given in numbers of simulation steps.

	Phase I	Phase II
$T$	$5 \times 10^7$	$3 \times 10^8$
$T_{\text{move}}$	200 000	$\infty$
$N_{\text{list}}$	100	1000
$N_{\text{im}}$	5	5
$T_{\text{wt}}$	10 000	1 000 000
$s$	$5 \times 10^{-3}$	$1 \times 10^{-3}$
$\tau$	0.1	...
$\kappa$	0	...

Gonze *et al.*<sup>27</sup> A 22-dimensional order-parameter space is constructed using the copy number of each species.

We simulate this system using both conventional Gillespie simulations and the umbrella sampling algorithm. For the umbrella sampling algorithm, basins are defined using the number count of  $M_P$ :  $M_P < 25$  is basin A, and  $M_P > 120$  is basin B. These numbers are chosen such that an unrestricted trajectory would go through both basins in each oscillation. No comparison of the rate of convergence was made with conventional simulations or FFS since there are no significant bottlenecks in phase space along the circadian cycle. Rather, we examine both the projection of the reaction path onto three different phase planes (Fig. 7) as well as the

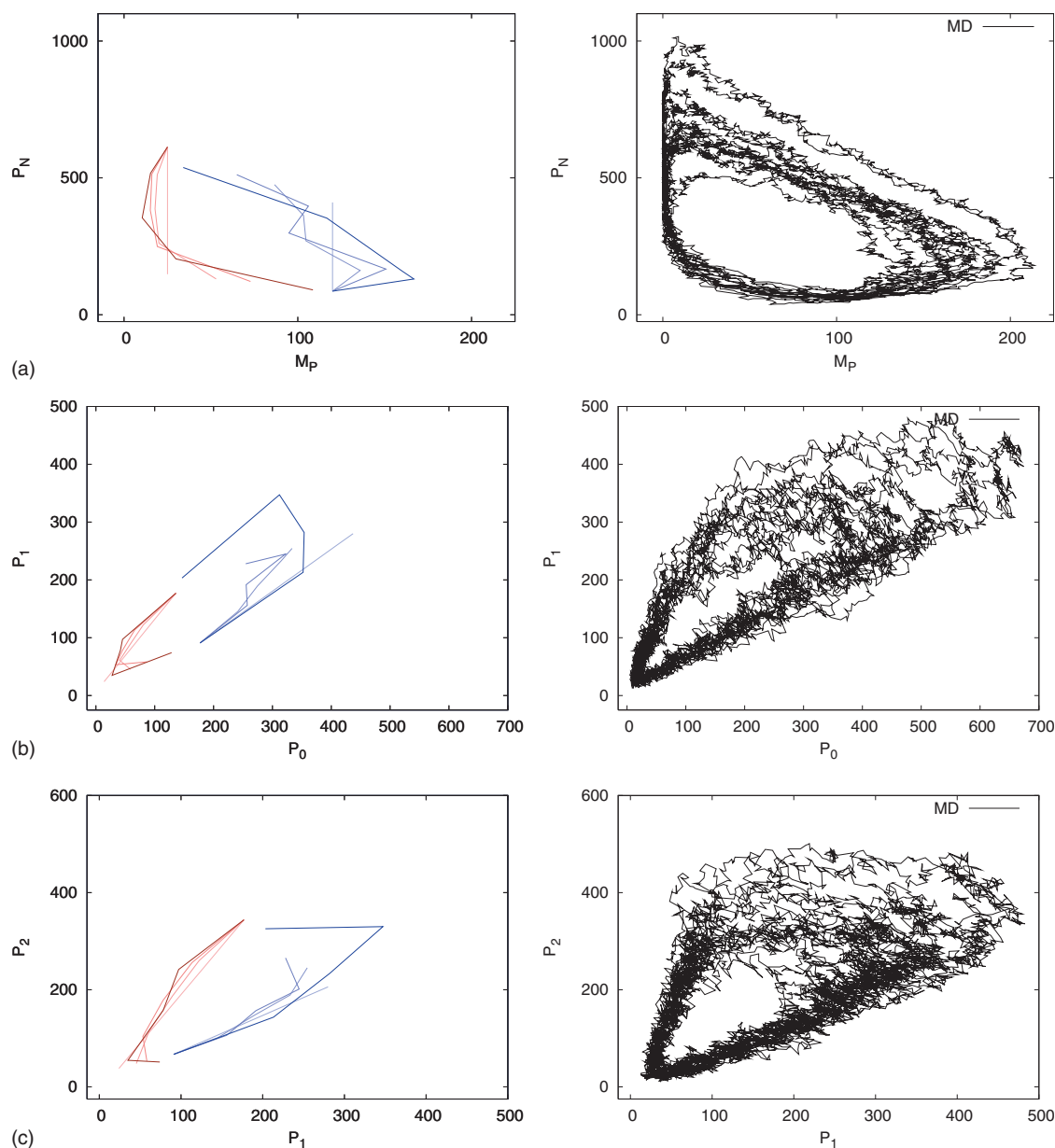


FIG. 7. Projections of circadian oscillations onto system variables. The left figures show the progression of the string used by the Voronoi-based umbrella sampling method, where the progression goes from light to dark, and the forward paths are shown in red and the backward paths are shown in blue. These snapshots were taken at  $T=0$ , 1, 2, and  $50 \times 10^6$  simulation steps. The right figures show a typical straightforward Gillespie trajectory which was run for  $T=8 \times 10^6$ , or about 14 revolutions. The top figures show the projection of the oscillation onto the  $P_N M_P$  phase plane, and they agree with previous results (Ref. 27). The middle and bottom figures show the projection onto the  $P_0 P_1$  and  $P_1 P_2$  phase planes, respectively. Good agreement between the configuration of the string and the straightforward trajectories is obtained for all projections onto pairs of the 22 degrees of freedom in the system.

TABLE III. Mean first passage times (MFPTs) for the forward and backward halves of the limit cycle for the circadian oscillatory system obtained by conventional sampling (CONV) and umbrella sampling (UMB). Basin  $A$  is defined as all points with mRNA number counts less than 25, and basin  $B$  is defined as all points with mRNA number counts greater than 120.  $T_f$  is the MFPT from  $A$  to  $B$ , and  $T_b$  is the MFPT from  $B$  to  $A$ . The MFPTs are given in hours.

	$T_f$	$T_b$	$T_{\text{tot}}$
CONV	$13.4 \pm 0.4$	$11.3 \pm 0.2$	$24.7 \pm 0.6$
UMB	$13.4 \pm 0.8$	$12.5 \pm 0.6$	$25.9 \pm 1.4$

accuracy of the mean period of oscillation. Figure 7 shows the evolution of the forward and backward strings from the initial guesses toward the reaction tube obtained by conventional simulations shown on the right. The end points of the initial forward and backward paths were obtained using four points from a short conventional simulation. Good agreement is shown between the converged string used by the umbrella sampling algorithm and the conventional trajectory in all projections, which demonstrates the ability of the umbrella sampling algorithm to describe an oscillatory path in a high-dimensional order-parameter space. The rate is obtained using only the local weight-balancing algorithm; the parameters used are given in Table II. A comparison of the mean first passage time for both parts of the loop and the total period obtained by conventional sampling and umbrella sampling is given in Table III.

#### IV. DISCUSSION

Here we demonstrated the application of the nonequilibrium umbrella sampling algorithm<sup>5,6,29</sup> to an extended phase space that separates “forward” and “backward” trajectories with respect to two arbitrarily defined basins in phase space. In this extended phase space, rates can be easily computed as the flux into the corresponding product basin. The dual-string formalism introduced here also enables straightforward calculation of statistics for periodic systems, such as the circadian clock model.<sup>27,28</sup> For systems that instead reach limit cycles in response to periodically varying external forces, the nonequilibrium umbrella sampling algorithm could be applied by associating with each copy of the system a “clock” variable that reports the time within the period of the perturbation. This variable would be saved with the phase space point during boundary crossings and used to reset the external environment when a walker is reintroduced to a region following an attempt to leave it. In other words, the reinitialized copy would proceed from the point in the period at which the phase space point used to restart the simulation was saved, so the “clock” runs continuously for each trajectory. This would avoid mismatches between phase space points and the external perturbation, which could lead the system away from the limit cycle. The output of the algorithm (a steady-state probability distribution in space time) would be equivalent to a time-ordered sequence of the probabilities of the spatial order parameters along the limit cycle.

We showed for a two-dimensional system that the convergence of the umbrella sampling algorithm was faster than FFS<sup>18–21</sup> when intermediates were present, as is common in

complex systems. In general, FFS is sensitive to local minima along the reaction path. For a trajectory to be terminated in the standard version of FFS, it must either reach the next interface in the sequence, or return to the first interface. For systems with long-lived intermediates such as the two-dimensional system examined above, this results in the integration of long trajectory segments during which the system is trapped in a local minimum. To alleviate this problem, a pruning procedure was introduced that terminates trajectories on their way back to the first interface (see Appendix C of Allen *et al.*<sup>19</sup>). In this procedure, fewer trajectories that go backward (toward the first interface) are simulated, and the trajectories that are simulated are given a higher weight to compensate for those that were prematurely terminated. It remains an open question as to whether this pruning procedure provides significant boosts to computational efficiency because  $P(\lambda_{i+1}|\lambda_i)$  includes contributions from both trajectories that go straight to  $\lambda_{i+1}$  and ones that first go back to  $\lambda_{i-1}$ , and thus many long trajectory segments that sample the intermediate could still be required to determine  $P(\lambda_{i+1}|\lambda_i)$  accurately.

The method explored here was developed concurrently with the “trajectory parallelization and tilting” scheme recently proposed by Vanden-Eijnden and Venturoli.<sup>29</sup> That method builds on nonequilibrium umbrella sampling to obtain an equivalent calculation of transition rates (for equilibrium or nonequilibrium systems) using fluxes in phase space given a set of regions. The method works by first obtaining the full steady-state distribution (with contributions from both  $S_A$  and  $S_B$ ) and using these results to obtain the conditional distributions in  $S_A$  and  $S_B$ . The rate calculation discussed in Sec. II B differs in that it works for an arbitrary pair of basins that can be defined independent of the Voronoi regions, and it is performed simultaneously with the determination of the probability distribution rather than as a separate step. The extended phase-space formalism is also desirable in that it naturally incorporates the dual-direction scheme into the existing theory.

Vanden-Eijnden and Venturoli also propose a weight update scheme that is built on the condition that the net flux into any region is zero, as opposed to using the continuity of flux across a boundary as was done in Sec. II C. There is then only one equation per region, plus the normalization condition. They rely exclusively on this global weight balancing scheme and use only a single lattice; they showed that their implementation led to accurate results for the Müller potential with a transition rate of about  $5 \times 10^{-3}$  steps<sup>-1</sup>. Although these additional modifications of the nonequilibrium umbrella sampling algorithm are promising, a wider variety of systems must be studied to determine their utilities because we found that, in general, problems make themselves apparent as systems become more challenging to sample.

One can conceive of other (related) methods to estimate rates. It was previously mentioned in the discussion of Dickson *et al.*<sup>6</sup> that rates can be calculated exactly by stitching together individual trajectories obtained piecewise in our method. This approach seems promising at first, since it allows for calculation of all moments of the first-passage time distribution, but it is currently not possible in practice. Points

are deleted from the flux list as it is refreshed, and longer trajectories have a higher probability of being overwritten, so a transition path ensemble generated in this fashion is heavily biased toward short trajectories.

Alternatively, one can define each region as a state, and use state-to-state transitions to build a rate matrix, as in milestoneing.<sup>14–16</sup> This assumes that the states are Markovian, i.e., that the probability of making a transition from state  $i$  to state  $j$  is the same for each time step *everywhere in state  $i$* . For this reason, this approximation becomes better and better as the states get smaller. A natural idea is to instead use the interfaces between regions as the set of states instead of the regions themselves. It has been shown that there exists a set of optimal milestones, for which the Markovian assumption is exact, and that these are the isocommittor surfaces that foliate the space between the two basins.<sup>17</sup> Furthermore, it has been suggested that the isocommittor surfaces can be obtained using the interfaces between Voronoi polyhedra formed by a discretization of the principal curve obtained by the string method.<sup>13,17</sup> However, it should be noted that even if the string converges exactly to the principal curve, the interfaces orthogonal to the string only approximate isocommittor surfaces *at the string*. The isocommittor surfaces are not, in general, planar, and the difference between the interfaces and isocommittors will depend on the width of the transition tube and the radius of curvature of the string.

The rate expression employed in Vanden-Eijnden and Venturoli<sup>29</sup> and the present study derives directly from the fundamental definition of a reaction rate. The advance in the present study is a practical one, concerned with how best to obtain the needed statistics. This work builds on remarkable advances over the last decade or so in reaction rate approaches. Advances such as transition path sampling,<sup>35–37</sup> transition interface sampling,<sup>23–25</sup> and FFS<sup>18–22</sup> go beyond transition state theory (TST)<sup>38–40</sup> in that the rate is calculated using statistics of an ensemble of paths, and the transition state itself does not have to be identified. Although nonequilibrium umbrella sampling is also a form of path sampling, one can view the rate calculation here as a limiting case of TST with the “transition state” moved to the boundary of the product basin. In this case, there is no recrossing by definition, since paths in the  $S_B$  ensemble are initiated the instant that the system enters the  $B$  basin. With a basic computational framework to calculate stationary distributions and rates in equilibrium and nonequilibrium systems now established, it remains to delineate the rules that guide the convergence of such simulations. In addition to exploring larger applications, doing so requires further extending the theoretical foundations. Success in this regard could also better link the practical advances to the emerging principles of nonequilibrium statistical mechanics.<sup>3</sup>

## ACKNOWLEDGMENTS

We wish to thank Albert Pan for a critical reading of the manuscript. This work was supported by the National Science Foundation and Natural Sciences and Engineering Research Council.

- <sup>1</sup>Committee on CMMP 2010, *Condensed-Matter and Materials Physics: The Science of the World Around Us* (National Academies Press, Washington, D.C., 2007).
- <sup>2</sup>G. M. Whitesides, *Chem. Eng. News* **85**, 12 (2007).
- <sup>3</sup>D. J. Evans and G. Morriss, *Statistical Mechanics of Nonequilibrium Liquids*, 2nd ed. (Cambridge University Press, New York, 2008).
- <sup>4</sup>G. M. Torrie and J. P. Valleau, *J. Comput. Phys.* **23**, 187 (1977).
- <sup>5</sup>A. Warmflash, P. Bhimalapuram, and A. R. Dinner, *J. Chem. Phys.* **127**, 154112 (2007).
- <sup>6</sup>A. Dickson, A. Warmflash, and A. R. Dinner, *J. Chem. Phys.* **130**, 074104 (2009).
- <sup>7</sup>W. E. E. Vanden-Eijnden, and W. Ren, *J. Phys. Chem. B* **109**, 6688 (2005).
- <sup>8</sup>W. Ren, E. Vanden-Eijnden, P. Maragakis, and W. E. E., *J. Chem. Phys.* **123**, 134109 (2005).
- <sup>9</sup>W. E. E., W. Ren, and E. Vanden-Eijnden, *Chem. Phys. Lett.* **413**, 242 (2005).
- <sup>10</sup>L. Maragliano, A. Fischer, E. Vanden-Eijnden, and G. Ciccotti, *J. Chem. Phys.* **125**, 024106 (2006).
- <sup>11</sup>T. F. Miller, E. Vanden-Eijnden, and D. Chandler, *Proc. Natl. Acad. Sci. U.S.A.* **104**, 14559 (2007).
- <sup>12</sup>M. Venturoli, E. Vanden-Eijnden, and G. Ciccotti, *J. Math. Chem.* **45**, 188 (2009).
- <sup>13</sup>E. Vanden-Eijnden and M. Venturoli, *J. Chem. Phys.* **130**, 194103 (2009).
- <sup>14</sup>A. K. Faradjian and R. Elber, *J. Chem. Phys.* **120**, 10880 (2004).
- <sup>15</sup>D. Shalloway and A. K. Faradjian, *J. Chem. Phys.* **124**, 054112 (2006).
- <sup>16</sup>A. M. A. West, R. Elber, and D. Shalloway, *J. Chem. Phys.* **126**, 145104 (2007).
- <sup>17</sup>E. Vanden-Eijnden, M. Venturoli, G. Ciccotti, and R. Elber, *J. Chem. Phys.* **129**, 174102 (2008).
- <sup>18</sup>R. J. Allen, P. B. Warren, and P. R. ten Wolde, *Phys. Rev. Lett.* **94**, 018104 (2005).
- <sup>19</sup>R. J. Allen, D. Frenkel, and P. R. ten Wolde, *J. Chem. Phys.* **124**, 024102 (2006).
- <sup>20</sup>R. J. Allen, D. Frenkel, and P. R. ten Wolde, *J. Chem. Phys.* **124**, 194111 (2006).
- <sup>21</sup>C. Valeriani, R. J. Allen, M. J. Morelli, D. Frenkel, and P. R. ten Wolde, *J. Chem. Phys.* **127**, 114109 (2007).
- <sup>22</sup>R. J. Allen, C. Valeriani, S. Tanase-Nicola, P. R. ten Wolde, and D. Frenkel, *J. Chem. Phys.* **129**, 134704 (2008).
- <sup>23</sup>T. S. van Erp, D. Moroni, and P. G. Bolhuis, *J. Chem. Phys.* **118**, 7762 (2003).
- <sup>24</sup>D. Moroni, P. G. Bolhuis, and T. S. van Erp, *J. Chem. Phys.* **120**, 4055 (2004).
- <sup>25</sup>T. S. van Erp and P. G. Bolhuis, *J. Comput. Phys.* **205**, 157 (2005).
- <sup>26</sup>A. Ma and A. R. Dinner, *J. Phys. Chem. B* **109**, 6769 (2005).
- <sup>27</sup>D. Gonze, J. Halloy, and A. Goldbeter, *J. Biol. Phys.* **28**, 637 (2002).
- <sup>28</sup>D. Gonze, J. Halloy, and A. Goldbeter, *Proc. Natl. Acad. Sci. U.S.A.* **99**, 673 (2002).
- <sup>29</sup>E. Vanden-Eijnden and M. Venturoli, *J. Chem. Phys.* **131**, 044120 (2009).
- <sup>30</sup>A. Dickson and A. R. Dinner, “Enhanced sampling of nonequilibrium steady states,” *Annu. Rev. Phys. Chem.* (in press).
- <sup>31</sup>R. P. Sear, *J. Chem. Phys.* **128**, 214513 (2008).
- <sup>32</sup>E. E. Borrero and F. A. Escobedo, *J. Chem. Phys.* **129**, 024115 (2008).
- <sup>33</sup>J. C. Dunlap, *Cell* **96**, 271 (1999).
- <sup>34</sup>D. T. Gillespie, *J. Phys. Chem.* **81**, 2340 (1977).
- <sup>35</sup>C. Dellago, P. G. Bolhuis, F. S. Csajka, and D. Chandler, *J. Chem. Phys.* **108**, 1964 (1998).
- <sup>36</sup>D. Frenkel and B. Smit, *Understanding Molecular Simulation: From Algorithms to Applications* (Academic, London, 2002).
- <sup>37</sup>C. Dellago and P. G. Bolhuis, *Adv. Polym. Sci.* **221**, 167 (2009).
- <sup>38</sup>H. Eyring, *J. Chem. Phys.* **3**, 107 (1935).
- <sup>39</sup>M. G. Evans and M. Polanyi, *Trans. Faraday Soc.* **31**, 875 (1935).
- <sup>40</sup>K. J. Laidler and M. C. King, *J. Phys. Chem.* **87**, 2657 (1983).