Monte Carlo sampling for stochastic weight functions

Daan Frenkel[a,1], K. Julian Schrenk[a], and Stefano Martiniani[a]

[a]Department of Chemistry, University of Cambridge, Cambridge CB2 1EW, United Kingdom

Contributed by Daan Frenkel, May 18, 2017 (sent for review December 13, 2016; reviewed by David Ceperley and Athanassios Z. Panagiotopoulos)

Conventional Monte Carlo simulations are stochastic in the sense that the acceptance of a trial move is decided by comparing a computed acceptance probability with a random number, uniformly distributed between 0 and 1. Here, we consider the case that the weight determining the acceptance probability itself is fluctuating. This situation is common in many numerical studies. We show that it is possible to construct a rigorous Monte Carlo algorithm that visits points in state space with a probability proportional to their average weight. The same approach may have applications for certain classes of high-throughput experiments and the analysis of noisy datasets.

Monte Carlo simulations | transition state | basin volumes | stochastic optimization | free-energy calculation

Monte Carlo simulations aim to sample the states of the system under study such that the frequency with which a given state is visited is proportional to the weight (often "Boltzmann" weight) of that state. The equilibrium distribution of a system, that is, the distribution for which every state occurs with a probability proportional to its (Boltzmann) weight, is invariant under application of a single Monte Carlo step. Algorithms that satisfy this criterion are said to satisfy "balance" (1). Usually, we impose a stronger condition, "detailed balance," which implies that the average rate at which the system makes a transition from an arbitrary "old" state (o) to a "new" state (n) is exactly balanced by the average rate for the reverse rate. The detailed balance condition is a very useful tool to construct valid Markov chain Monte Carlo (MCMC) algorithms. We can write the detailed balance condition as

$$P(\mathbf{x}_o)P_{\mathrm{gen}}(o\to n)P_{\mathrm{acc}}(o\to n)=P(\mathbf{x}_n)P_{\mathrm{gen}}(n\to o)P_{\mathrm{acc}}(n\to o), \quad [1]$$

where $P(\mathbf{x}_i)$ denotes the equilibrium probability that the system is in state i (in this case, i can stand for o or n) characterized by a (usually high-dimensional) coordinate $\mathbf{x}_i$. $P_{\mathrm{gen}}(i\to j)$ denotes the probability to generate a trial move from state i to state j. In the simplest case, $P_{\mathrm{gen}}(i\to j)$ may be the probability to generate a random displacement that will move the system from $\mathbf{x}_i$ to $\mathbf{x}_j$, but in general the probability to generate a trial move may be much more complex (e.g., ref. 2). Finally $P_{\mathrm{acc}}(i\to j)$ denotes the probability that a trial move from state i to state j will be accepted.

Many simple Monte Carlo (MC) algorithms satisfy in addition microscopic reversibility, which means that $P_{\mathrm{gen}}(i\to j) = P_{\mathrm{gen}}(j\to i)$. In that case, detailed balance implies that

$$\frac{P_{\mathrm{acc}}(o\to n)}{P_{\mathrm{acc}}(n\to o)} = \frac{P(\mathbf{x}_n)}{P(\mathbf{x}_o)}. \quad [2]$$

There are many acceptance rules that satisfy this criterion. The most familiar one is the so-called Metropolis rule (3):

$$P_{\mathrm{acc}}(o\to n) = \mathrm{Min}\left\{1, \frac{P(\mathbf{x}_n)}{P(\mathbf{x}_o)}\right\}. \quad [3]$$

The acceptance probability for the reverse move follows by permuting o and n. In the specific case of Boltzmann sampling of configuration space, where the equilibrium distribution is proportional to the Boltzmann factor, $P(\mathbf{x}_i)\sim\exp(-U_i/k_B T)$, where U_i is the potential energy of the system in the state characterized by the coordinate $\mathbf{x}_i$, T is the absolute temperature, and k_B is the Boltzmann constant. In that case, we obtain the familiar result

$$P_{\mathrm{acc}}(o\to n) = \mathrm{Min}\,\{1, \exp[-(U_n - U_o)/k_B T]\}. \quad [4]$$

Monte Carlo Simulations with Noisy Acceptance Rules. There are many situations where conventional MCMC cannot be used because the quantity that determines the weight of a state i is, itself, the average of a fluctuating quantity. Specifically, we consider the case of weight functions fluctuating according to a Bernoulli process, that is, in an intermittent manner, although our approach is not limited to Bernoulli processes. Examples that we consider are "committor" functions, or the outcome of a stochastic minimization procedure.

Note that the problem that we are discussing here is different from the cases considered by Bhanot and Kennedy (4) and by Ceperley and Dewing (5). As we discuss below, these earlier papers consider cases where the weights are nonlinear functions of a fluctuating argument (e.g., an action or an energy), in which case the average of the function is not equal to the function of the average argument. In contrast, we consider the case where the probability to sample a point is given rigorously by the average of the stochastic estimator of the weight function.

To give a specific example, we consider the problem of computing the volume of the basin of attraction of a particular energy minimum i in a high-dimensional energy landscape (6–10). The algorithms developed in refs. 6–9 rely on the fact that, for every point $\mathbf{x}$ in a d-dimensional configuration space, we can determine unambiguously whether this point belongs to the basin of attraction of minimum i: If a (steepest-descent or similar) trajectory that starts at point $\mathbf{x}$ ends in minimum i, the "oracle function" $\mathcal{O}_i(\mathbf{x})=1$, and otherwise it is zero.

Significance

Markov chain Monte Carlo is the method of choice for sampling high-dimensional (parameter) spaces. The method requires knowledge of the weight function (or likelihood function) determining the probability that a state is observed. However, in many numerical applications the weight function itself is fluctuating. Here, we present an approach capable of tackling this class of problems by rigorously sampling states proportionally to the average value of their fluctuating likelihood. We demonstrate that the method is capable of computing the volume of a basin of attraction defined by stochastic dynamics as well as being an efficient method to identify a transition state along a known reaction coordinate. We briefly discuss how the method might be extended to experimental settings.

Author contributions: D.F. and S.M. designed research; D.F., K.J.S., and S.M. performed research; K.J.S. and S.M. analyzed data; and D.F. and S.M. wrote the paper.

Reviewers: D.C., University of Illinois at Urbana–Champaign; and A.Z.P., Princeton University.

The authors declare no conflict of interest.

[1]To whom correspondence should be addressed. Email: df246@cam.ac.uk.

This article contains supporting information online at www.pnas.org/lookup/suppl/doi:10.1073/pnas.1620497114/-/DCSupplemental.

However, many minimizers are not deterministic—and hence the oracle function is probabilistic. (In fact, historical evidence suggests that ancient oracles were probabilistic at best). In that case, if we start a number of minimizations at point $\mathbf{x}$, some will have $\mathcal{O}_i(\mathbf{x}) = 1$ and others have $\mathcal{O}_i(\mathbf{x}) = 0$. We denote with $P_{\mathcal{O}}^{(i)}(\mathbf{x})$ the average value of the Bernoulli process defined by the oracle function $\mathcal{O}_i(\mathbf{x})$. In words, $P_{\mathcal{O}}^{(i)}(\mathbf{x})$ is the probability that the oracle function associated with point $\mathbf{x}$ has a value of one.

We could obtain an estimate for the average weight $P_{\mathcal{O}}^{(i)}(\mathbf{x}) = \langle \mathcal{O}^{(i)}(\mathbf{x})\rangle$ by sampling the same point very many times. However, such an approach would be prohibitively expensive. Below we show that one can construct a rigorous algorithm to sample according to the weight $P_{\mathcal{O}}^{(i)}(\mathbf{x})$, without having to obtain an accurate estimate of $\langle \mathcal{O}^{(i)}(\mathbf{x})\rangle$.

First, however, we generalize the concept of a basin volume v_i as the integral of the probability (the "probability mass") that a stochastic minimization will end up in basin i:

$$v_i \equiv \int d\mathbf{x}\, P_{\mathcal{O}}^{(i)}(\mathbf{x}). \qquad [5]$$

Clearly, for a deterministic process, we recover the original definition of a basin volume. Moreover, we have

$$\sum_{i=1}^{\Omega} v_i = V_{total}, \qquad [6]$$

where Ω is the number of distinct minima. This equation expresses the fact that every trajectory must end up somewhere. If we wish to compute the volume v_i in Eq. **5**, we must be able to sample points with a probability $P_{\mathcal{O}}^{(i)}(\mathbf{x})$, even though we do not know this function a priori.

Naive MC Algorithm. We first describe a naive (and very inefficient) but rigorous MC algorithm to sample stochastic weight functions. After that, we show how the algorithm can be made more efficient.

Our aim is to construct a MC algorithm that will visit points $\mathbf{x}$ with a probability proportional to $P_{\mathcal{O}}(\mathbf{x})$. The normalized configuration-space density $\rho(\mathbf{x})$ is then proportional to $P_{\mathcal{O}}(\mathbf{x})$. If we can sample configuration space with this density $\rho(\mathbf{x})$, the computation of the volume in Eq. **5** becomes a free-energy calculation, for which standard techniques exist (2).

Let us consider two points ($\mathbf{x}$ and $\mathbf{x}'$) between which we can carry out trial moves. The steady-state configuration-space density $\rho(\mathbf{x})$ is determined by our choice for the acceptance probability P_{acc}:

$$\rho(\mathbf{x}) P_{\text{acc}}(\mathbf{x} \to \mathbf{x}') = \rho(\mathbf{x}') P_{\text{acc}}(\mathbf{x}' \to \mathbf{x}). \qquad [7]$$

The average acceptance probability for a very large number of trial moves from point $\mathbf{x}$ to point $\mathbf{x}'$ is $\langle \mathcal{O}(\mathbf{x}')\rangle = P_{\mathcal{O}}(\mathbf{x}')$. If we consider a large number of trial moves in the reverse direction, the acceptance probability is $P_{\mathcal{O}}(\mathbf{x})$. In steady state, the populations should be such that detailed balance holds and hence

$$\rho(\mathbf{x}) P_{\mathcal{O}}(\mathbf{x}') = \rho(\mathbf{x}') P_{\mathcal{O}}(\mathbf{x}) \qquad [8]$$

or

$$\frac{\rho(\mathbf{x})}{\rho(\mathbf{x}')} = \frac{P_{\mathcal{O}}(\mathbf{x})}{P_{\mathcal{O}}(\mathbf{x}')}. \qquad [9]$$

In other words, trial moves that are accepted with a probability equal to the instantaneous value of the oracle function generate the correct distribution of points in configuration space, proportional to $P_{\mathcal{O}}(\mathbf{x})$.

Note that in this naive version of the algorithm, the acceptance rule is not the Metropolis rule that considers the ratio of two weights. Here it is the probability itself. Hence, whenever the probability becomes very low, the acceptance of moves decreases proportionally. We address this problem in what follows.

"Configurational-Bias" Approach. With the naive algorithm described above, the acceptance of moves becomes small when the system moves into a region of configuration space where $P_{\mathcal{O}}(\mathbf{x})$ is low, and hence the "diffusion coefficient" that determines the rate at which configuration space is sampled becomes small. As a consequence, sampling of the wings of the distribution may become prohibitively slow. This problem can be alleviated by basing the Monte Carlo sampling on the average weight of a larger number of trial points. To implement this sampling, we use an approach that resembles configurational-bias MC (CBMC) (11), but is different in some respects. The key point to note is that, if we know all random numbers that determine the value of the oracle function—including the random numbers that control the behavior of the stochastic minimizer—then in the extended space of coordinates plus random numbers, the value of the oracle function is always the same for a given point.

We can then generate a random walk in this extended space, between points that are surrounded by a "cloud" of k points where we compute the oracle function (at this stage k is arbitrary). We denote the central point (i.e., the one to which or from which moves are attempted) by $\mathbf{x}_{\text{B}}$, where "B" stands for "backbone." The reason for calling this point a backbone point is that we will be sampling the k points connected to it, but we will not compute the oracle function at the backbone point. Hence, $\mathbf{x}_{\text{B}}$ may even be located in a region where the oracle function is strictly zero (Fig. 1). We introduce these backbone points because it facilitates generating a random walk that satisfies detailed balance.

The coordinates of the k cloud points around $\mathbf{x}_{\text{B}}$ are given by

$$\mathbf{x}_{\text{B},i} = \mathbf{x}_{\text{B}} + \Delta_i \qquad [10]$$

with $i = \{1, 2, \cdots, k\}$. The vectors Δ are generated by some stochastic protocol; for example, the vectors may be uniformly distributed in a hypersphere with radius R_h. The precise choice

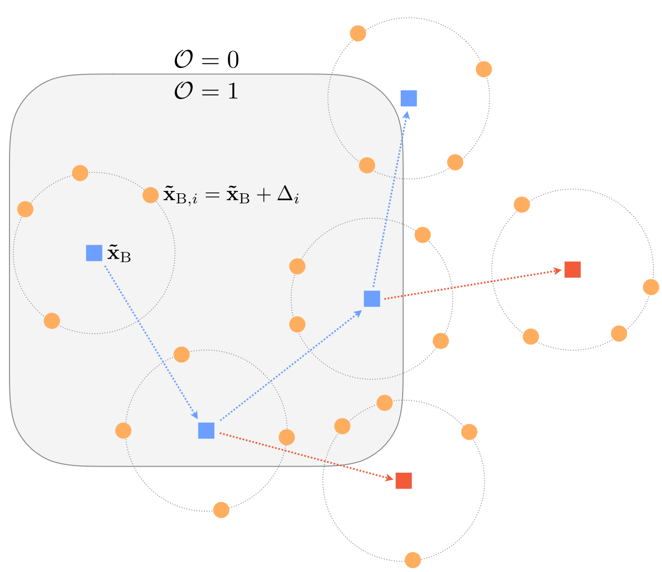

Fig. 1. Cloud sampling. Shown is an illustration of the configurational-bias–like approach for a simple oracle defined by the gray shaded region, such that $\mathcal{O} = 1$ inside the gray boundary and $\mathcal{O} = 0$ outside it. Blue and red squares denote typical accepted and rejected backbone points $\tilde{\mathbf{x}}_{\text{B}}$, respectively. The cloud points $\tilde{\mathbf{x}}_{\text{B},\,i} = \tilde{\mathbf{x}}_{\text{B}} + \Delta_i$ are represented by orange circles. In this example we randomly sample $k = 4$ cloud points from a circle of fixed radius centered on the backbone point (dotted circles). Each cloud is sampled with probability proportional to the Rosenbluth weight defined in Eq. **12**. Note that backbone points (e.g., the one at top right) may fall outside the region where $\mathcal{O} = 1$ because the Rosenbluth weight (Eq. **12**) does not depend on the value of the oracle at the backbone point.

of the protocol does not matter, as long as the rules are not changed during the simulation. For a fixed protocol, the set $\mathbf{x}_{\mathrm{B},i}$ is uniquely determined by a set of random numbers $\mathcal{R}_{\mathrm{B}}$. Finally, we note that the value of the oracle function $\mathcal{O}_i$ for a given point $\mathbf{x}_{\mathrm{B},i}$ is uniquely determined by another set of random numbers $\mathcal{R}_{\mathcal{O}}$ (e.g., the random numbers in a stochastic minimization).

We now define an extended state space

$$\tilde{\mathbf{x}}_{\mathrm{B}} \equiv \{\mathbf{x}_{\mathrm{B}}, \mathcal{R}_{\mathrm{B}}, \mathcal{R}_{\mathcal{O}}\}. \qquad [11]$$

In this space, the oracle functions are no longer fluctuating quantities.

We can now construct a MCMC to visit (but not sample) backbone points. To this end, we compute the "Rosenbluth weight" of point $\tilde{\mathbf{x}}_{\mathrm{B}}$ as

$$W(\tilde{\mathbf{x}}_{\mathrm{B}}) = \sum_{i=1}^{k} \mathcal{O}_i \omega_i, \qquad [12]$$

where $\mathcal{O}_i \equiv \mathcal{O}(\tilde{\mathbf{x}}_{\mathrm{B},i})$ and $\omega_i \equiv \omega(\tilde{\mathbf{x}}_{\mathrm{B},i})$ denotes a (Boltzmann) biasing factor. For unbiased sampling, $\omega_i = 1$, but for biased sampling, as is used for instance in thermodynamic integration (2, 6–9), other choices for ω_i can be used.

We can then construct a MCMC algorithm where the acceptance of a trial move from the old $\tilde{\mathbf{x}}_{\mathrm{B}}^{(o)}$ to the new $\tilde{\mathbf{x}}_{\mathrm{B}}^{(n)}$ is given by

$$P_{\mathrm{acc}}(o \to n) = \mathrm{Min}\left\{1, \frac{W(\tilde{\mathbf{x}}_{\mathrm{B}}^{(n)})}{W(\tilde{\mathbf{x}}_{\mathrm{B}}^{(o)})}\right\}. \qquad [13]$$

As the probabilities to generate the trial directions for forward and backward moves and the generation of random numbers that determine the value of the oracle function are also uniform, the resulting MC algorithm satisfies superdetailed balance (2, 11) and a given backbone point $\tilde{\mathbf{x}}_{\mathrm{B}}$ will be visited with a probability proportional to $W(\tilde{\mathbf{x}}_{\mathrm{B}})$. It is important to note that the acceptance rules for the Markov chain determine transition probabilities between the backbone points, but that these points are never sampled. Below, we show that we sample the values of the observable quantities only for the cloud points.

Note that during a trial move, the state of the old point is not changed, and hence it retains the same trial directions (hence the same set $\{\mathcal{R}_{\mathrm{B}}\}$) and the same set $\{\mathcal{R}_{\mathcal{O}}\}$. If the trial move is rejected, it is this "extended point" that is sampled again. This is different from standard CBMC.

The approach of Eq. **13** can be easily incorporated into more sophisticated sampling schemes such as parallel tempering (PT) (12, 13), as discussed in *Supporting Information* and shown in Fig. 2.

Sampling. We have shown that backbone points will be visited with a probability proportional to its instantaneous Rosenbluth weight $P_B(\tilde{\mathbf{x}}_{\mathrm{B}}) \sim W(\tilde{\mathbf{x}}_{\mathrm{B}})$. However, it is not our aim to sample the backbone points but the points in the cloud around the backbone. Let us consider two such points i_o and i_n that belong to the cloud of the old and new of backbone points. The condition for detailed balance states that the forward and reverse fluxes between points i_o and i_n must balance,

$$\begin{aligned} &P(\tilde{\mathbf{x}}_{\mathrm{B}}^{(o)}) P_{\mathrm{gen}}(\tilde{\mathbf{x}}_{\mathrm{B}}^{(n)}) P_{\mathrm{sel}}(i_n) P_{\mathrm{acc}}(o \to n) \\ &= P(\tilde{\mathbf{x}}_{\mathrm{B}}^{(n)}) P_{\mathrm{gen}}(\tilde{\mathbf{x}}_{\mathrm{B}}^{(o)}) P_{\mathrm{sel}}(i_o) P_{\mathrm{acc}}(n \to o), \end{aligned} \qquad [14]$$

where $P_{\mathrm{sel}}(i_n)$ denotes the probability to select point i_n from among the cloud of points around $\mathbf{x}_{\mathrm{B}}^{(n)}$ [and similarly, for $P_{\mathrm{sel}}(i_o)$]. Note that this detailed balance condition comes on top of the one for transitions between the backbone points, which resulted in the acceptance rule **13** for the acceptance of moves between those backbone points. In contrast, Eq. **14** expresses the detailed balance condition for transitions between cloud points. In what follows, we assume that the probability $P_{\mathrm{gen}}(\tilde{\mathbf{x}})$ to generate cloud points around a given backbone point does not depend on $\tilde{\mathbf{x}}$. As a consequence, the probabilities P_{gen} for forward and backward moves cancel, and we drop P_{gen} from the detailed-balance equation.

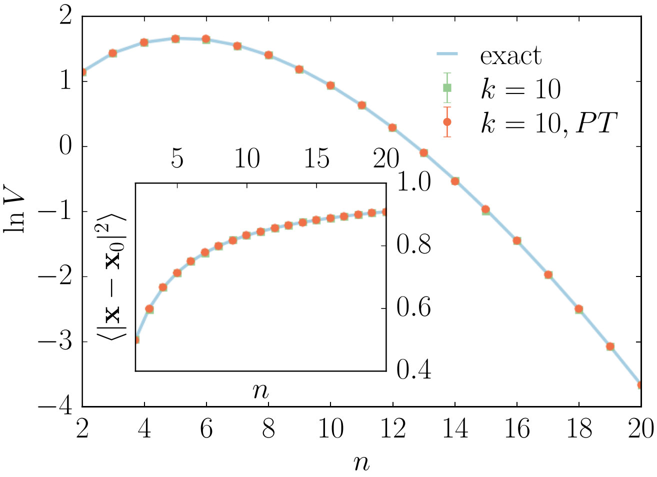

Fig. 2. Deterministic oracle. Shown is volume calculation for an n-dimensional hypersphere with radius $R = 0.5$ and $n \in [2, 20]$. Numerical results (symbols) were obtained by the configurational bias approach of Eq. **13**, with k cloud points and MBAR. *PT* refers to calculations performed by parallel tempering, described in *Supporting Information*. *Inset* shows mean square displacement computed by Eq. **17**. Solid blue curves denote the analytical results. The symbols refer to the numerical results. The error bars, corresponding to twice the SE estimate for the computed volume, are smaller than the size of the symbols.

To achieve the desired sampling of cloud points, we impose that a given cloud point $i \equiv \mathbf{x}_{\mathrm{B},i}$ is selected with a probability

$$P_{\mathrm{sel}}(i) = \frac{\mathcal{O}(i)\omega(i)}{\sum_{j=1}^{k} \mathcal{O}(j)\omega(j)} = \frac{\mathcal{O}(i)\omega(i)}{W(\tilde{\mathbf{x}}_{\mathrm{B}})}. \qquad [15]$$

If we now make use of the fact that the probability to visit a given backbone point at $\tilde{\mathbf{x}}_{\mathrm{B}}$ is proportional to $W(\tilde{\mathbf{x}}_{\mathrm{B}})$, it follows that the overall probability $P(i; \tilde{\mathbf{x}}_{\mathrm{B}})$ that a cloud point i will be sampled is proportional to the desired weight:

$$P(i; \tilde{\mathbf{x}}_{\mathrm{B}}) \sim W(\tilde{\mathbf{x}}_{\mathrm{B}}) \frac{\mathcal{O}(i)\omega(i)}{W(\tilde{\mathbf{x}}_{\mathrm{B}})} = \mathcal{O}(i)\omega(i).$$

But note that $\mathcal{O}(i)$ has not yet been averaged. If we perform the average over the oracle function, we obtain

$$P(i) \sim \langle \mathcal{O}(i) \rangle \, \omega(i).$$

Hence, by combining our rule for visiting backbone points with a Rosenbluth-style selection of the point to be sampled, we ensure that we sample with the correct weight.

The approach that we describe here is better than the naive algorithm because it achieves faster "diffusion" through parts of configuration space where $\langle \mathcal{O} \rangle \omega$ is small.

However, even though Rosenbluth-style sampling ensures that all points in space are sampled with the correct frequency, it is not an efficient algorithm. The reason is obvious: To compute the weights W, the oracle function must be computed for k points, and yet in naive Rosenbluth sampling, only one point would be sampled.

Fortunately, this drawback can be overcome. Rather than sampling one point at a time, we take steps between backbone points sampled according to Eq. **13** and compute the quantity to be sampled for all k cloud points belonging to the current backbone point, as described below. An illustration of the method is given in Fig. 1.

For every backbone point $\tilde{\mathbf{x}}_B$ visited, we can compute the observable (say A) of the set of k cloud points as follows:

$$A_{\text{sampled}} = \frac{\sum_{i=1}^{k} \mathcal{O}_i \omega_i A_i}{\sum_{i=1}^{k} \mathcal{O}_i \omega_i}. \quad [16]$$

The average of A during a MCMC simulation of L steps is

$$\frac{1}{L} \sum_{j=1}^{L} \left(\frac{\sum_{i=1}^{k} \mathcal{O}_i \omega_i A_i}{\sum_{i=1}^{k} \mathcal{O}_i \omega_i} \right)_j, \quad [17]$$

where the index j labels the different backbone states visited.

Combine with "Waste-Recycling" MC. Efficiency can be further improved by using the approach underlying "waste-recycling" MC (14). This approach allows us to sample all trial cloud points in the sampling, even if the actual trial backbone move is rejected. The approach of ref. 14 allows us to combine the information of the accepted and the rejected states in our sampling. Specifically, we denote the probability to accept a move from an old state o to a new state n by $P_{\text{acc}}(o \to n)$; then, normally we would sample $A_{\text{sampled}}(n)$ if the move is accepted and $A_{\text{sampled}}(o)$ otherwise. However, we can do better by combining the information and sample

$$A_{\text{wr}} = P'_{\text{acc}}(o \to n) A_{\text{sampled}}(n) + [1 - P'_{\text{acc}}(o \to n)] A_{\text{sampled}}(o), \quad [18]$$

where P'_{acc} denotes the acceptance probability for any valid MCMC algorithm (not just Metropolis). In fact, it is convenient to use the symmetric (Barker) rule (15) to compute P'_{acc}. In that case, we would sample

$$A_{wr} = \frac{\left(\sum_{i=1}^{k} \mathcal{O}_i \omega_i A_i\right)_{old} + \left(\sum_{i=1}^{k} \mathcal{O}_i \omega_i A_i\right)_{new}}{\left(\sum_{i=1}^{k} \mathcal{O}_i \omega_i\right)_{old} + \left(\sum_{i=1}^{k} \mathcal{O}_i \omega_i\right)_{new}}. \quad [19]$$

Hence, all $2k$ points that have been considered are included in the sampling.

Numerical Results

Basin Volume Calculations. To test the proposed algorithm we compute the basin volume (probability mass) for a stochastic oracle function as defined in Eq. **5**. We choose a few simple oracle functions, for which the integral in Eq. **5** can be solved analytically. The volume calculations were performed using the multistate-Bennett acceptance ratio (MBAR) method (16) described in ref. 9. As described in ref. 9, a high-dimensional volume calculation is in essence a free-energy calculation, where minus the log of the volume plays the role of the free energy.

We compute the dimensionless free-energy difference between a region of known volume $\widehat{f}_{\text{ref}} = -\ln V_{\text{ref}} + c$ and the equilibrium distribution of points sampled uniformly within the basin $\widehat{f}_{\text{tot}} = -\ln V_{\text{tot}} + c$, estimated by MBAR up to an additive constant c. Because $f_{\text{ref}} = -\ln V_{\text{ref}}$ is known, we obtain the basin volume as $f_{\text{tot}} = f_{\text{ref}} + (\widehat{f}_{\text{tot}} - \widehat{f}_{\text{ref}})$. We use 15 replicas with positive coupling constants for all examples discussed herein; see ref. 9 for details of the method.

We first tested the method for a deterministic oracle, namely a simple n-dimensional hypersphere of known volume $V_{n\text{-ball}} = \pi^{n/2} R^n / \Gamma(n/2+1)$ with radius $R = 0.5$ and $n \in [2, 20]$. As shown in Fig. 2 we correctly recover the volume and the mean-square displacement, using the acceptance rule defined in Eq. **13** for $k = 10$ cloud points. Fig. 2 suggests that the algorithm is sampling the correct equilibrium distributions.

Next, we tested the method for a stochastic oracle function defined such that

$$P_{\mathcal{O}}(\mathbf{x}) \sim \begin{cases} 1 & \text{if } |\mathbf{x}| < R \\ \exp[-(|\mathbf{x}| - R)/\lambda] & \text{if } |\mathbf{x}| \geq R \end{cases} \quad [20]$$

with volume

$$V = 2(R^n/n + \lambda^n \exp(R/\lambda)\Gamma(n, R/\lambda))\pi^{n/2} R^n / \Gamma(n/2),$$

where $\Gamma(a, x)$ is the incomplete gamma function. Results for dimensions $n \in [2, 20]$, $R = 0.5$, and $\lambda = 0.1$ are shown in Fig. 3. Note that, despite the volume being finite, the basin is unbounded in the sense that the average value of the oracle tends to zero only as $|\mathbf{x}| \to \infty$. As the dimensionality of the basin increases, all of the volume will concentrate away from the center of mass in regions of space where the oracle has a high probability of returning 0. Hence, it becomes more difficult for a random walker to diffuse efficiently as the dimensionality of space increases. Evidence that the sampling efficiency goes down for small numbers of cloud points is shown in Fig. 3. For $n < 6$ results seem to be independent of the number of cloud points. However, growing deviations are observed for increasing n and accuracy increases significantly for a growing number of cloud points k. For large n, the largest contribution to the integral comes from values of $|x|$ where the average value of the oracle function is very small [$\mathcal{O}(10^{-9})$ for $n = 20$]. We carried out our simulations with at most 100 cloud points. In that case, inefficient sampling could be expected when the average oracle function is significantly less than 0.01. As Fig. 3 shows, for the case of $k = 100$ systematic deviations from the analytical result show up for $n \geq 11$, where the dominant contributions come from points where the average oracle function is $\mathcal{O}(10^{-5})$.

Transition-State Finding. The algorithm that we described above has wider applicability than the specific examples that we discussed. As an illustration of a very different application, we show that our approach can be used to efficiently identify the transition state along a known reaction coordinate.

Note that points in the transition-state ensemble (in the one-dimensional case: just one point) are characterized by the property that the committor has an average value of 0.5. However, any individual trajectory will either be crossing ("1") or non-crossing ("0"). Hence, the "signal" is stochastic. As an illustra-

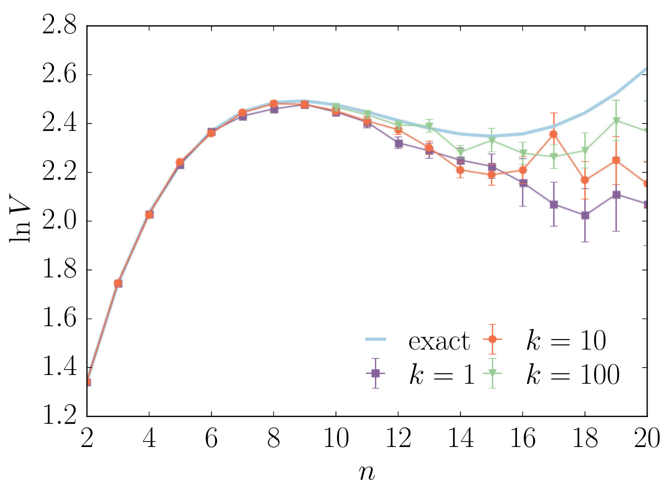

Fig. 3. Stochastic oracle. Shown is volume calculation for the oracle defined in Eq. **20** with radius $R = 0.5$, $\lambda = 0.1$, and dimensions $n \in [2, 20]$. Symbols (lines are guide to the eye) are numerical results obtained by the configurational-bias approach of Eq. **13** with k cloud points and MBAR. The light blue curve denotes the analytical results and error bars refer to twice the SE as estimated by MBAR. At large n accuracy increases by increasing k as the random walker diffuses more efficiently through regions of space where $\langle \mathcal{O} \rangle \ll 1$. However, if the integral is dominated by points where the average value of the oracle function is (much) less than the inverse of the number of cloud points, slow convergence leads to systematic errors in the sampling. Analogous results obtained by parallel tempering can be seen in Fig. S1.

APPLIED MATHEMATICS

tion, we consider the (trivial) one-dimensional case of a particle with kinetic energy K sampled according to the one-dimensional Maxwell–Boltzmann distribution, crossing a Gaussian barrier with height $U_{\text{tr}} = 30kT$ and variance $\sigma^2 = 1$. (We choose as our unit of length σ, and hence in our reduced units $kT = \sigma^2$.) We define the oracle symmetrically such as

$$\mathcal{O}(x) = \begin{cases} 1 & \text{if } K > U_{\text{tr}} - U(x) \\ 0 & \text{if } K \leq U_{\text{tr}} - U(x) \end{cases} \quad \textbf{[21]}$$

and constrain the walk to reject moves for which the potential energy is below that of the initial position, such that $\mathcal{O} = 0$ if $U(x) < U(x_0)$; we choose $x_0 = 2\sigma$. By thus constraining the sampling, we are excluding the "reactant" and "product" states from our sampling. In Fig. 4 we show results for backbone step-size 0.25σ, cloud radius 0.25σ, and varying number of cloud points k. One can clearly see that as the number of cloud points increases the system diffuses faster toward the transition state whereas for the traditional single-point sampling the walker does not move at all from the initial position.

Relation to Earlier Work

The problem of Monte Carlo sampling in the presence of noise has been discussed by Bhanot and Kennedy (4) and Ceperley and Dewing (5).

Bhanot and Kennedy (4) considered how to construct an unbiased estimator of an exponential function (e.g., a ratio of Boltzmann weights) with a fluctuating argument. This method involves constructing an estimator on the basis of a number of independent samples. The method is subject to certain limitations (it is not guaranteed to generate acceptance probabilities between 0 and 1) and, crucially, it addresses the problem that the average of an exponential function with a fluctuating argument is not equal to the function of the average argument. In this respect, the work of ref. 4 is similar to that of Ceperley and Dewing (5) who considered the problem of performing Boltzmann MCMC sampling in cases where the energy function is noisy. As in the case of ref. 4, the Boltzmann weight is a nonlinear function of the energy and therefore the Boltzmann factor corresponding to the average energy is not the same as the average of the Boltzmann factor obtained by sampling over energy fluctuations. Specifically, Ceperley and Dewing (5) consider the case where the calculation of the energy function is subject to statistical errors with zero mean. In that case, we cannot use the conventional Metropolis rule $P_{\text{acc}} = \text{Min}\{1, \exp(-\beta\Delta u)\}$, where u is the instantaneous value of the energy difference, because what is needed to compute the correct acceptance probability is $\exp(-\beta\langle\Delta u\rangle)$, but what is sampled is $\langle\exp(-\beta\Delta u)\rangle \neq \exp(-\beta\langle\Delta u\rangle)$. Ceperley and Dewing (5) showed that if the fluctuations in Δu are normally distributed, with constant variance σ, then we can still get an algorithm that samples the correct Boltzmann distribution, if we use as the acceptance rule

$$P_{\text{acc}} = \text{Min}\{1, \exp[-\beta\Delta u - (\beta\sigma)^2/2]\}. \quad \textbf{[22]}$$

Note that the situation considered in refs. 4 and 5 is very different from the case that we consider here, as we focus on the situations where the average of the (fluctuating) oracle functions is precisely the weight function that we wish to sample. However, the present approach allows us to rederive the Ceperley and Dewing (5) result. We note that, as before, we can consider extended states characterized by the spatial coordinates of the system and by the random variables that characterize the noise in the energy function. To discuss the approach of Ceperley and Dewing (5) in the present language, it is easiest to consider the case that the variance in the energy of the individual states is normally distributed, with constant variance σ_s. The average Boltzmann factor of extended state i is then

$$\langle P_i\rangle = \exp[-\beta\langle u\rangle_i]\exp[+(\beta\sigma_s)^2/2] \quad \textbf{[23]}$$

and therefore

$$\frac{\langle P_n\rangle}{\langle P_o\rangle} = \exp[-\beta\langle\Delta u\rangle]. \quad \textbf{[24]}$$

Hence, the average Boltzmann factor of any state i is still proportional to the correct Boltzmann weight. However, an MCMC algorithm using the instantaneous Boltzmann weights would not lead to correct sampling as superdetailed balance yields

$$\frac{P_n(\mathbf{x}_n)}{P_o(\mathbf{x}_o)} = \exp[-\beta\Delta u] \quad \textbf{[25]}$$

and hence

$$\left\langle\frac{P_n}{P_o}\right\rangle = \exp[-\beta\langle\Delta u\rangle + (\beta\sigma)^2/2], \quad \textbf{[26]}$$

which is not equal to

$$\frac{\langle P_n\rangle}{\langle P_o\rangle} = \exp[-\beta\langle\Delta u\rangle]. \quad \textbf{[27]}$$

If, however, we use the Ceperley–Dewing acceptance rule, we would get

$$\begin{aligned}\left\langle\frac{P_n}{P_o}\right\rangle &= \exp[-\beta\langle\Delta u\rangle + (\beta\sigma)^2/2] \times \exp[-(\beta\sigma)^2/2] \\ &= \exp[-\beta\langle\Delta u\rangle] = \frac{\langle P_n\rangle}{\langle P_o\rangle}.\end{aligned} \quad \textbf{[28]}$$

Hence, with this rule the states would (on average) be visited with the correct probability. Note that, as the noise enters nonlinearly in the acceptance rule, the Ceperley–Dewing algorithm is very different from the one that we derived above. Note also that the present derivation makes it clear that the Ceperley–Dewing algorithm can be easily generalized to cases where the noise in the energy is not normally distributed, as long as the distribution of the noise is state independent.

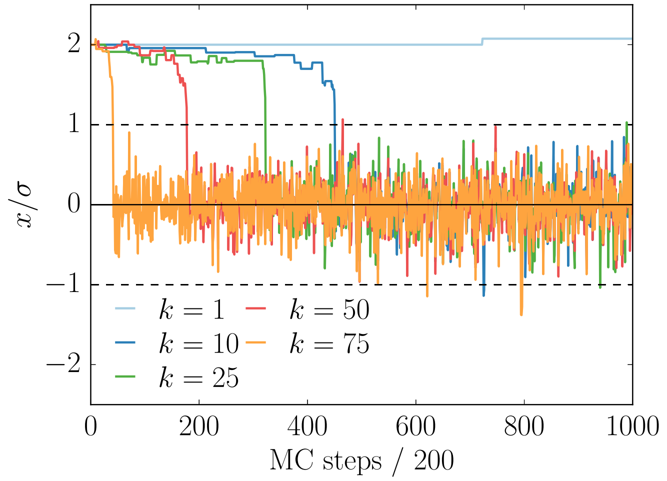

Fig. 4. Transition-state finding. The simple case of one-dimensional barrier crossing is defined (symmetrically) by the stochastic oracle in Eq. **21**. A series of random walks are performed according to Eq. **13** with different numbers of cloud points k. The walkers are constrained to reject moves for which the energy is below that of the initial position, thus excluding reactants and products from the sampling. Shown is the position of the walker backbone along the reaction coordinate as a function of the number of MCMC steps. For increasing k the random walkers diffuse more efficiently and therefore converge faster to the transition state. Traditional single-point sampling does not move at all from the initial condition.

Conclusions and Outlook

Thus far, the algorithm described above was presented as a method to perform MC sampling in cases where the weight function itself is fluctuating.

However, we suggest that the method is not limited to numerical sampling: It could be used to steer sampling of experimental control parameters in experiments that study stochastic events (e.g., crystal nucleation, cell death, or even the effect of advertising). Often, the occurrence of the desired event depends on a large number of variables (temperature, pressure, pH, concentration of various components) and we want to select the optimal combination. However, as the desired event itself is stochastic, individual measurements provide little guidance. One might aim to optimize the conditions by accumulating sufficient statistics for individual state points. However, such an approach is expensive. The procedure described in the preceding sections suggests that it may be better to perform experiments in a cloud of state points around a backbone point. We could then accept or reject the trial move to a new backbone state, using the same rule as in Eq. **13**. In this way, the experiment could be made to evolve toward "interesting" regions of parameter space.

ACKNOWLEDGMENTS. D.F. acknowledges support by Engineering and Physical Sciences Research Council (EPSRC) Program Grant EP/I001352/1 and EPSRC Grant EP/I000844/1. K.J.S. acknowledges support by the Swiss National Science Foundation (Grants P2EZP2-152188 and P300P2-161078). S.M. acknowledges financial support from the Gates Cambridge Scholarship.

1. Manousiouthakis VI, Deem MW (1999) Strict detailed balance is unnecessary in Monte Carlo simulation. *J Chem Phys* 110:2753–2756.
2. Frenkel D, Smit B (2002) *Understanding Molecular Simulation* (Academic, San Diego).
3. Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E (1953) Equation of state calculations by fast computing machines. *J Chem Phys* 21:1087–1092.
4. Bhanot G, Kennedy AD (1985) Bosonic lattice gauge theory with noise. *Phys Lett B* 157:70–76.
5. Ceperley DM, Dewing M (1999) The penalty method for random walks with uncertain energies. *J Chem Phys* 110:9812–9820.
6. Xu N, Frenkel D, Liu AJ (2011) Direct determination of the size of basins of attraction of jammed solids. *Phys Rev Lett* 106:245502.
7. Asenjo D, Paillusson F, Frenkel D (2014) Numerical calculation of granular entropy. *Phys Rev Lett* 112:098002.
8. Martiniani S, Schrenk KJ, Stevenson JD, Wales DJ, Frenkel D (2016) Turning intractable counting into sampling: Computing the configurational entropy of three-dimensional jammed packings. *Phys Rev E* 93:012906.
9. Martiniani S, Schrenk KJ, Stevenson JD, Wales DJ, Frenkel D (2016) Structural analysis of high-dimensional basins of attraction. *Phys Rev E* 94:031301.
10. Martiniani S, Schrenk KJ, Ramola K, Chakraborty B, Frenkel D (2017) Numerical test of the Edwards conjecture shows that all packings become equally probable at jamming. *Nat Phys*, in press.
11. Frenkel D, Mooij GCAM, Smit B (1991) Novel Monte Carlo scheme to study structural and thermal properties of continuously deformable molecules. *J Phys Condens Matter* 3:3053–3076.
12. Lyubartsev AP, Martsinovski AA, Shevkunov SV, Vorontsov-Velyaminov PN (1992) New approach to Monte Carlo calculation of the free energy: Method of expanded ensembles. *J Chem Phys* 96:1776–1783.
13. Marinari E, Parisi G (1992) Simulated tempering: A new Monte Carlo scheme. *Europhys Lett* 19:451–458.
14. Frenkel D (2004) Speed up of Monte Carlo simulations by sampling of rejected states. *Proc Nat Acad Sci USA* 101:17571–17575.
15. Barker AA (1965) Monte Carlo calculations of radial distribution functions for a proton-electron plasma. *Aust J Phys* 18:119–134.
16. Shirts MR, Chodera JD (2008) Statistically optimal analysis of samples from multiple equilibrium states. *J Chem Phys* 129:124105.
17. Yan Q, de Pablo JJ (1999) Hyper-parallel tempering Monte Carlo: Application to the Lennard-Jones fluid and the restricted primitive model. *J Chem Phys* 111:9509–9516.
18. Bunker A, Dünweg B (2000) Parallel excluded volume tempering for polymer melts. *Phys Rev E* 63:016701.
19. Fukunishi H, Watanabe O, Takada S (2002) On the Hamiltonian replica exchange method for efficient sampling of biomolecular systems: Application to protein structure prediction. *J Chem Phys* 116:9058–9067.

APPLIED MATHEMATICS