Markov Processes

Let us summarize our discussion of Monte Carlo integration. We have learned that it is possible to evaluate integrals by a process in which the integration space is sampled randomly. Values of the integrand evaluated at these randomly chosen points can be summed and normalized, just as we do with methodical schemes based on equally spaced abscissas, to give a good estimate of the integral. These Monte Carlo schemes gain significant advantage over methodical approaches when applied to high-dimensional integrals. In these situations, and in particular when applied to statistical-mechanics integrals, the integral may have significant contributions from only a very small portion of the entire domain of integration. If these regions of integration can be sampled preferentially, with a well-characterized bias, then it is possible to correct for the biased sampling when summing the contributions to the integral, and thereby obtain a higher-quality estimate of the integral. This idea is known as importance sampling. The basic equation of importance-sampling Monte Carlo integration can be written in a compact form

$$I \equiv \langle f \rangle = \left\langle \frac{f}{\pi} \right\rangle_{\pi}$$

This formula states that the integral, defined here as the unweighted average of a function *f*, can be expressed as the weighted average of *f/π* , where *π* is the weight applied to the sampling. We are now ready to address the question of how to sample a space according to some weight *π*.

A *stochastic process* is a procedure by which a system moves through a series of well-defined states in a way that exhibits some element of randomness. A *Markov process* is a stochastic process that has no memory. That is, the probability that the system moves into a particular state depends only upon the state it is currently in, and not on the history of the past visitations of states. Thus, a Markov process can be fully specified via a set of transition probabilities π_{ij} that describe the likelihood that the system moves into state j given that it is presently in state i. The full set of transition probabilities can be viewed as a matrix Π.

As a simple example, we can consider a system that can occupy any of three states. The probability of moving from one state to another in a Markov process is given via the transition probability matrix (TPM)

$$\Pi \equiv \begin{pmatrix} \pi_{11} & \pi_{12} & \pi_{13} \\ \pi_{21} & \pi_{22} & \pi_{23} \\ \pi_{31} & \pi_{32} & \pi_{33} \end{pmatrix} = \begin{pmatrix} 0.1 & 0.5 & 0.4 \\ 0.9 & 0.1 & 0.0 \\ 0.3 & 0.3 & 0.4 \end{pmatrix} \tag{1.1}$$

where for the sake of example we have filled in the matrix with specific values for the transition probabilities. Now consider what happens in a process that moves from one state to another, each time selecting the new state according to the transition probabilities given here (for example, say the system presently is in state 1; generate a random number uniformly on (0,1); if the value is less than 0.1, stay in state 1; if between 0.1 and 0.6, move to state 2, otherwise move to state 3). One could construct a histogram to describe the number of times visited in each of the three states during the process. After a long period of sampling, steady state is reached and the histogram does not change with continued sampling. The histogram so obtained is called the <em>limiting distribution</em> of the Markov process. Examine the applet in Illustration 1 to see Markov sampling in action.

So what is the connection to Monte Carlo integration? The scheme is to devise a Markov process to yield a limiting distribution that covers the important regions of our simulated system. In this manner we can do importance sampling of a complex region of integration by specifying only the transition probabilities of the Markov process. To accomplish this we need to develop the connection between the transition probabilities and the limiting distribution.

Several important features should be noted: first, each probability is properly specified, <em>i.e.</em>, it is nonnegative and does not exceed unity; second, each row sums to unity, indicating a unit probability for going from one state to another in a given step; third, the diagonal elements are not necessarily zero, indicating that an acceptable outcome for a Markov step leaves the system in its present state. More on this detail later. In all of what follows it is important that we have a transition-probability matrix that corresponds to an ergodic process. Ergodicity was discussed in a previous section. In this context, it means that it is possible to get from one state to another via a sufficiently long Markov chain. Note it is not required that each state be accessible from every other state in a single step—it is OK (and very common) to have zero for some of the transition probabilities.

Limiting distribution

We consider now how the limiting distribution relates to the TPM. Consider the product of Π with itself

$$\Pi^2 \equiv \begin{pmatrix} \pi_{11} & \pi_{12} & \pi_{13} \\ \pi_{21} & \pi_{22} & \pi_{23} \\ \pi_{31} & \pi_{32} & \pi_{33} \end{pmatrix} \times \begin{pmatrix} \pi_{11} & \pi_{12} & \pi_{13} \\ \pi_{21} & \pi_{22} & \pi_{23} \\ \pi_{31} & \pi_{32} & \pi_{33} \end{pmatrix}$$

$$= \begin{pmatrix} \pi_{11}\pi_{11} + \pi_{12}\pi_{21} + \pi_{13}\pi_{31} & \pi_{11}\pi_{12} + \pi_{12}\pi_{22} + \pi_{13}\pi_{32} & etc. \\ \pi_{21}\pi_{11} + \pi_{22}\pi_{21} + \pi_{23}\pi_{31} & \pi_{21}\pi_{12} + \pi_{22}\pi_{22} + \pi_{23}\pi_{32} & etc. \\ \pi_{31}\pi_{11} + \pi_{32}\pi_{21} + \pi_{33}\pi_{31} & \pi_{31}\pi_{12} + \pi_{32}\pi_{22} + \pi_{33}\pi_{32} & etc. \end{pmatrix}$$

Look closely at the first (1,1) element. It is the sum of three terms. The first term, $\pi_{11}\pi_{11}$ is the probability of staying in state 1 for two successive steps, given that the system started in state 1. Similarly, the second term in the sum $\pi_{12}\pi_{21}$ the probability that the system moves in successive steps from state 1 to state 2, and then back again. Finally the third term is the probability of moving from 1 to 3 and back to 1. Thus the (1,1) terms in the product matrix contains all ways of going from state 1 back to state 1 in two steps. Likewise, the (1,2) term of the product matrix is the probability that the system moves from state 1 to state 2 in exactly two steps. The same interpretation holds for all the terms in the product matrix. Thus the square of Π is a two-step transition-probability matrix, and in general the multiple product Π^n is the n-step TPM

$$\Pi^n \equiv \begin{pmatrix} \pi_{11}^{(n)} & \pi_{12}^{(n)} & \pi_{13}^{(n)} \\ \pi_{21}^{(n)} & \pi_{22}^{(n)} & \pi_{23}^{(n)} \\ \pi_{31}^{(n)} & \pi_{32}^{(n)} & \pi_{33}^{(n)} \end{pmatrix}$$

where each term $\pi_{ij}^{(n)}$ is defined as the probability of going from state *i* to state *j* in exactly *n* Markov steps.

Let us define $\pi_i^{(0)}$ as a unit state vector, thus (for a 3-state system)

$$\pi_1^{(0)} = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \quad \pi_2^{(0)} = \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} \quad \pi_3^{(0)} = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}$$

Then $\pi_i^{(n)} \equiv \pi_i^{(0)}\Pi^n$ is a vector describing the probabilities for ending at each state after *n* Markov steps beginning at state *i*

$$\pi_1^{(n)} = \pi_1^{(0)}\Pi^n \equiv \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \pi_{11}^{(n)} & \pi_{12}^{(n)} & \pi_{13}^{(n)} \\ \pi_{21}^{(n)} & \pi_{22}^{(n)} & \pi_{23}^{(n)} \\ \pi_{31}^{(n)} & \pi_{32}^{(n)} & \pi_{33}^{(n)} \end{pmatrix} = \begin{pmatrix} \pi_{11}^{(n)} & \pi_{12}^{(n)} & \pi_{13}^{(n)} \end{pmatrix}$$

The limiting distribution corresponds to $n \to \infty$, and will be independent of the initial state *i* if the TPM describes an ergodic process

$$\pi_1^{(\infty)} = \pi_2^{(\infty)} = \pi_3^{(\infty)} \equiv \pi$$

by which we define π.

The limiting distribution obeys a stationary property. Starting with its expression as a limit

$$\pi = \lim_{n\to\infty} \left[\pi_i^{(0)}\Pi^n \right]$$

we can take out the last multiplication with the TPM

$$\pi = \left(\lim_{n\to\infty} \left[\pi_i^{(0)}\Pi^{n-1} \right] \right)\Pi$$

The limit in parentheses is still gives the limiting distribution

$$\pi = \pi\Pi \tag{1.2}$$

Evidently π is a left eigenvector of the matrix Π, with unit eigenvalue. That such an eigenvector (with unit eigenvalue) exists is guaranteed by the stipulation that each row of Π sum to unity (this is an application of the Peron-Frobenius theorem).

Written explicitly, the eigenvector equation for π corresponds to the set of equalities (one for each state *i* in the system)

$$\pi_i = \sum_j \pi_j \pi_{ji} \tag{1.3}$$

where the sum extends over all states. If the limiting probabilities π_i and the transition probabilities π_{ij} all satisfy the following relation

$$\pi_i \pi_{ij} = \pi_j \pi_{ji} \tag{1.4}$$

then they also satisfy the eigenvector equation as presented in Eq. (1.3), as is easily shown

$$\begin{aligned} \pi_i &= \sum_j \pi_j \pi_{ji} \\ &= \sum_j \pi_i \pi_{ij} \\ &= \pi_i \sum_j \pi_{ij} = \pi_i \end{aligned}$$

The relation given in Eq. (1.4) is known as *detailed balance*, or the *principle of microscopic reversibility*. As demonstrated, it presents a sufficient condition for the probabilities to satisfy Eq. (1.3), but it is not a necessary condition. In fact, for a given, well-formed TPM it is likely that the limiting-distribution probabilities *do not* satisfy detailed balance. For example, the particular TPM introduced in Eq. (1.1) clearly cannot satisfy detailed balance; one of the elements (π_{23}) is zero, while its detailed-balance counterpart (π_{32}) is not zero. Equation (1.4) cannot be satisfied for this pair (unless perhaps π_3 is zero, but this clearly is not the case here since there is a route to state 3 via state 1).

Deriving transition probabilities

The utility of detailed balance is not in determining the limiting distribution from a set of transition probabilities. In fact, our need is the opposite: we have a specification for the distribution of states, and we want to apply a Markov process to generate states according to this distribution; how can we construct appropriate transition probabilities to achieve this aim? As demonstrated in Illustration 2, there are many possible choices for the set of transition probabilities that yield a given limiting distribution. Of course, we need to generate only one set to perform the Markov process. The choice of transition probabilities can be dictated by convenience and performance (that is, how well they sample all relevant states for a finite-length Markov chain).

Detailed balance is an extremely very useful guiding principle, because it leads us to generate a valid set of transition probabilities while considering them only a pair at a time. This contrasts with the full eigenvector equation, which involves all the states at once. The implication there is that all of the transition probabilities must be specified together and at one time, so to satisfy this relation between them. By focusing instead on the sufficient condition of detailed balance, a great burden is removed. We do not have to evaluate all transition probabilities for all states at once, and in fact we do not have to evaluate all transition probabilities, period. Instead we can get away with evaluating them only as we need them. The calculation of the transition probabilities can be tied right into the Markov chain, so that only those encountered during a given sequence are actually computed. The number of microstates in a typical statistical mechanics system is huge, so there is immense savings in exploiting this "just-in-time" calculation scheme given to us by detailed balance. Still, one should not lose sight of the fact the microscopic reversibility is not required, and that it may be advantageous to violate the principle at some times; but one should take caution that alternative transition probabilities are consistent with the expected limiting distribution. Oddly enough, in a molecular simulation (and unlike the simple examples given here) there is no easy way to check, even *a posteriori,* that the limiting distribution generated by the Markov sequence actually coincides with the target distribution. Consequently it is quite possible for an error in the formulation of the transition probabilities to remain undetected.

The Metropolis Algorithm

One way to implement a Markov sequence is as described earlier: at each step make a decision about which state to move to next, with the likelihood of selecting any state given exactly by the corresponding transition probability; once the selection is made, then move to that state (with complete certainty, with probability 1). Detailed balance could be used to specify all the transition probabilities, but the problem with this scheme is that it again requires us to specify all transition probabilities beforehand; it is not a just-in-time approach. An algorithm that makes full use of detailed balance was developed by Metropolis, Rosenbluth, Rosenbluth, Teller and Teller in 1953. This truly seminal work represents one of the first applications of stochastic computational techniques to the treatment of deterministic physical problems.

The idea of the Metropolis scheme is to select new states according to any convenient transition probability matrix (called the *underlying transition matrix*), but not to accept every state so generated. Instead, each state is accepted with a probability that ensures that the overall transition probability is, via detailed balance, consistent with the desired limiting distribution. The acceptance probabilities are evaluated only when the acceptance question arises, so only those needed are computed. The overall transition probability depends on the transition probability of the underlying matrix, and the acceptance probability. Within this framework, and even for a given underlying transition matrix, there are many ways to formulate the overall transition probabilities. The Metropolis method represents one choice. At a given point in the Markov chain, let the present state be state *i*. The recipe is:

- With probability τ_{ij}, choose a trial state *j* for the move

- If $\pi_j > \pi_i$, accept *j* as the new state, otherwise accept state *j* with probability $\chi = \tau_{ji}\pi_j/\tau_{ij}\pi_i$. This is accomplished by selecting a random number R uniformly on (0,1); acceptance occurs if $R < \chi$.

- If the trial state (*j*) is rejected, the present state (*i*) is retained and is taken as the next one in the Markov chain. This means that the transition probability π_{ii} is, in general, nonzero.

What are the transition probabilities for this algorithm? We can write them as follows

$$\begin{aligned} \pi_{ij} &= \tau_{ij} \min(1,\chi) \\ \pi_{ji} &= \tau_{ji} \min(1,1/\chi) \\ \pi_{ii} &= 1-\sum_{j\neq i}\pi_{ij} \end{aligned} \qquad (1.5)$$

We can examine these against the detailed balance criterion

$$\pi_i \pi_{ij} \overset{?}{=} \pi_j \pi_{ji}$$

$$\pi_i \tau_{ij} \min(1,\chi) \overset{?}{=} \pi_j \tau_{ji} \min(1,1/\chi)$$

Regardless of whether χ is greater or less than unity, this can equation becomes

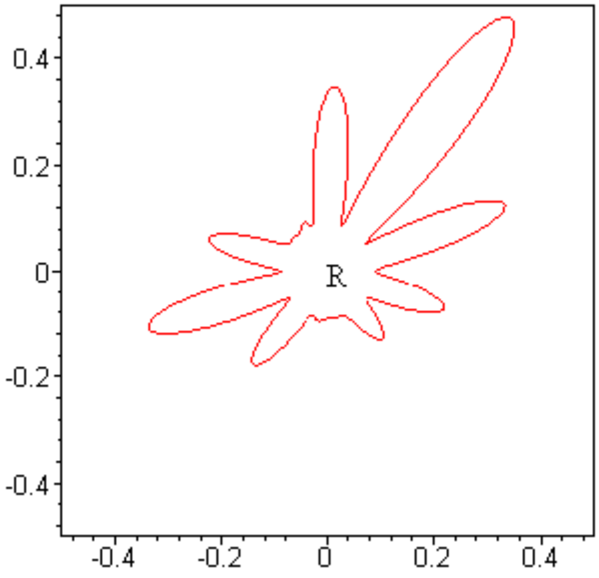
$$\pi_i \tau_{ij} \chi \overset{?}{=} \pi_j \tau_{ji}$$

Upon insertion of the definition of χ, this equation becomes an identity and detailed balance is verified.

The original formulation of this algorithm by Metropolis *et al*. is based on a symmetric underlying TPM, such that $\tau_{ij} = \tau_{ji}$, but this restriction is not necessary if one is careful to account for the asymmetry when formulating the acceptance criterion. Very efficient algorithms can be developed by exploiting this degree of flexibility.

Markov chains and importance sampling

Before we turn to the application of Markov chains and the Metropolis method to molecular simulation, let us finish this topic by returning to the simple example we used previously to introduce the concepts of Monte Carlo integration and importance sampling. Illustration 3 contains our prototype of a two-dimension region R having non-



trivial shape. The formula for the mean-square distance of each point in R from the origin can be given by an integral over the square region V that contains R, with a switching function that sets the integrand to zero for points outside R

$$\left\langle r^2 \right\rangle = \frac{\int_{-0.5}^{+0.5} dx \int_{-0.5}^{+0.5} dy(x^2+y^2)s(x,y)}{\int_{-0.5}^{+0.5} dx \int_{-0.5}^{+0.5} dys(x,y)} = \frac{\left\langle r^2 s \right\rangle_V}{\left\langle s \right\rangle_V}$$

We will look at two methods for evaluating this integral with Markov-based importance sampling. In method 1 we take the weight π_1 for our importance-sampling formulation to be

$$\pi_1(x,y) = s(x,y)/q_1 \qquad (1.6)$$

where q_1 is a constant that ensures that π_1 is normalized over V. With this choice the integral transforms as follows

$$\left\langle r^2 \right\rangle = \frac{\left\langle \frac{r^2 s}{\pi_1} \right\rangle_{\pi_1}}{\left\langle \frac{s}{\pi_1} \right\rangle_{\pi_1}} = \frac{\left\langle q_1 r^2 \right\rangle_{\pi_1}}{\left\langle q_1 \right\rangle_{\pi_1}} = \frac{q_1 \left\langle r^2 \right\rangle_{\pi_1}}{q_1} = \left\langle r^2 \right\rangle_{\pi_1}$$

This result tells us something that we might have guessed at the outset, namely that we can evaluate the mean square distance by summing values of r^2 for points generated uniformly inside of R. In a moment we will consider how Metropolis sampling prescribes the means to generate these points. But first, it is of interest to consider an alternative importance-sampling scheme. It could make sense to weight the sampling even more toward the region of importance to r^2, and choose

$$\pi_2(x,y) = r^2 s / q_2 \qquad (1.7)$$

This choice works too, but in with a less obvious formulation

$$\left\langle r^2 \right\rangle = \frac{\left\langle \frac{r^2 s}{\pi_2} \right\rangle_{\pi_2}}{\left\langle \frac{s}{\pi_2} \right\rangle_{\pi_2}} = \frac{\left\langle q_2 \right\rangle_{\pi_2}}{\left\langle q_2 / r^2 \right\rangle_{\pi_2}} = \frac{q_2}{q_2 \left\langle 1/r^2 \right\rangle_{\pi_2}} = \frac{1}{\left\langle r^{-2} \right\rangle_{\pi_2}}$$

With points generated via a Markov chain with limiting distribution given by Eq. (1.7), we sum the reciprocal of r^2, and then at the end take the reciprocal of the average to obtain the desired average. As we see, in reaching too far in our importance scheme, we end up outsmarting ourselves. The important part of this reciprocal-average is just the opposite of the region we are now emphasizing. Nevertheless, the performance of this approach is comparable to that obtained by the first importance scheme we examined.

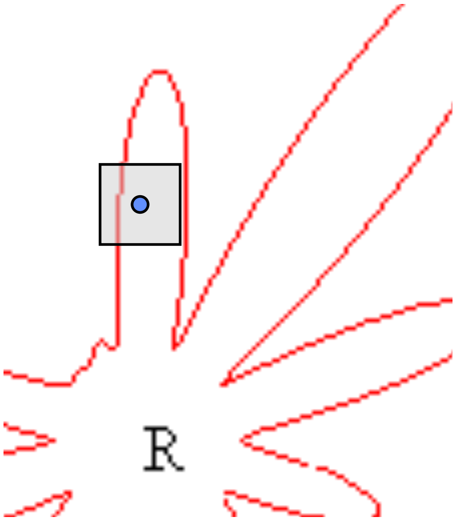
The Metropolis algorithm can be applied to each of these formulations. With it we proceed as follows

(1) Given a point in the region R, generate a new point in the vicinity of it. For convenience let this nearby region be defined as a square centered on the current point.

Importantly, the size of the square is fixed, and does not depend on the position of the point in R. Thus the new (x,y) coordinate is

$$x^{new} = x + \delta \times rand(-1,1) \quad y^{new} = y + \delta \times rand(-1,1)$$

where δ is a parameter defined the size of the local region for stepping to a new point and rand(-1,1) is a random number generated uniformly on (-1,1) (generating separate values for the x and y displacements). See Illustration 4.



(2) Accept this new point with probability $\min(1, \pi^{new} / \pi^{old})$. Thus, for the first method described above

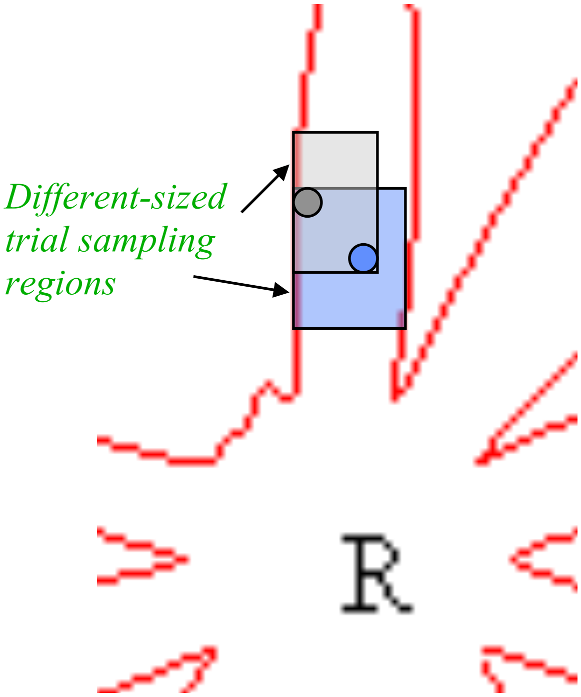
$$\frac{\pi_1^{new}}{\pi_1^{old}} = \frac{s^{new} / q_1}{s^{old} / q_1} = \frac{s^{new}}{s^{old}}$$

It is very fortunate that the normalization constant dropped out of the acceptance criterion, as it is often a highly nontrivial matter to determine this constant. Thus our acceptance decision is as follows

- Method 1: accept all moves that leave the state inside R, reject all attempts to move outside R

- Method 2: reject all trials that attempt to move outside R; if the move stays within R, accept it with probability $\min\left[1, r_{new}^2 / r_{old}^2\right]$, which obviously gives preference to the points more distant from the origin

There is an important point to be emphasized in this example. The underlying transition probability matrix is set by the trial-move algorithm, in which a new point is generated in a square centered on the original point. Since the size of this square is independent of the position of the current point, underlying trial matrix is symmetric—the probability of selecting a point *j* from *i* is proportional to 1/A, where A is the area of this square displacement region. Requiring A to be the constant ensures that $\tau_{ij} = \tau_{ji}$. It is tempting to introduce an efficiency in the algorithm, and to pre-screen the trials points so that they do not generate a Metropolis trial that falls outside of R. As shown in Illustration 5, this has the effect of making A smaller for points closer to the boundary, and thus makes the underlying transition-probability matrix asymmetric. The net result is that the boundary points are underrepresented in the Markov chain, and the ensemble average is skewed (*i.e.*, incorrect). It is important that rejected trials be included in the Markov chain, *i.e.*, π_{ii} is not zero, but is given by Eq. (1.5).



As our final topic of this section, we will hint at one of the limitations of importance sampling methods that will occupy much of our attention later. What if we want the absolute area of the region R, and not an average over it? Formally, this can be given by an integral over V

$$A = \int_{-0.5}^{+0.5} dx \int_{-0.5}^{+0.5} dy s(x,y) = \langle s \rangle_V$$

As before, it would be good to apply importance sampling to this integral, particularly if the area of R is much smaller than the area of V. Proceeding as before, let the importance weight be given by Eq. (1.6), $\pi_1 = s/q_1$. Then

$$A = \left\langle \frac{s}{\pi_1} \right\rangle_{\pi_1} = \left\langle q_1 \right\rangle_{\pi_1} = q_1$$

Unlike before, we now do need to know the normalization constant q_1, but the integral that gives this constant is exactly the integral that we are trying to evaluate! The lesson here is that absolute integrals can be very hard to evaluate by Monte Carlo methods, and importance sampling, by itself, does not rescue us from this difficulty.