Random Walks

In a random walk (Markov chain), one changes the state of the system randomly according to a fixed *transition rule*, $\mathcal{P}(s \to s')$, thus generating a random walk through state space, $\{s_0, s_1, s_2 \ldots\}$. The definition of a Markov process is that the next step is chosen from a probability distribution that depends only on the "present" position. This makes it very easy to describe mathematically. The process is often called the drunkard's walk. $\mathcal{P}(s \to s')$ is a probability distribution so it satisfies

$$\sum_{s'} \mathcal{P}(s \to s') = 1 \tag{1}$$

and

$$\mathcal{P}(s \to s') \geq 0. \tag{2}$$

Let $f_n(s)$ be the probability distribution of the walker after s steps. Then it is easy to write the evolution of f_n in terms of $\mathcal{P}$.

$$f_{n+1}(s') = \sum_s f_n(s)\mathcal{P}(s \to s') \tag{3}$$

or in vector-matrix notation:

$$f_{n+1} = \mathcal{P} f_n = \mathcal{P}^n f_1. \tag{4}$$

The first question is what is the final probability distribution (equilibrium state): $\lim_{n\to\infty} f_n$? Any equilibrium state would be an eigenfunction of $\mathcal{P}$ with unity eigenvalue. The next question is how many such eigenfunctions are there?

If the transition probability is *ergodic*, the distribution f_n will always converge to a *unique equilibrium state*. That means there is a unique solution to:

$$\sum_s f(s)\mathcal{P}(s \to s') = f(s'). \tag{5}$$

The transition is ergodic if:

1. One can move from any state to any other state in a finite number of steps with a nonzero probability, *i.e.*, there are no barriers that restrict any walk to a subset of the full configuration space.

2. It is not periodic. An example of a periodic rule is if the hopping on a bipartite lattice always proceeds from the A sites to the B sites and vice-versa so that one never forgets which site one started on. Non-periodic rules hold if $\mathcal{P}(s \to s) > 0$; if there is always some chance of staying put.

3. The average return time to any state is finite. This is always true in a finite system (*e.g.* periodic boundary conditions). It would be violated in a model of the expanding universe where the system gets further and further from equilibrium because there is no possibility of energy flowing between separated regions after the "big bang".

Under these conditions we can show that if $f_n(s)$ is the probability distribution of random walks after n steps, with $f_0(s)$ the initial condition, then:

$$f_n(s) = \pi + \sum_\lambda \epsilon_\lambda^n c_\lambda \phi_\lambda(s), \tag{6}$$

where the $\epsilon_\lambda < 1$. Hence the probability distribution converges exponentially fast to the stationary distribution π. Furthermore, the convergence is monotonic (it does not oscillate). Specifically, what we mean is that the distance between f_n and π is strictly decreasing: $|f_n - \pi| > |f_{n+1} - \pi|$.

The transition probabilities often satisfy the *detailed balance* property for same function: the transition rate from s to s' equals the reverse rate,

$$\pi(s)\mathcal{P}(s \to s') = \pi(s')\mathcal{P}(s' \to s). \tag{7}$$

If the pair $\{\pi(s), \mathcal{P}(s \to s')\}$ satisfy detailed balance and if $\mathcal{P}(s \to s')$ is ergodic, then the random walk must eventually have π as its equilibrium distribution. To prove this fact, sum the previous equation over s and use Eq.(1) to simplify the right-hand-side. Detailed balance is one way of making sure that we sample π; it is a sufficient condition. Note that detailed balance is not necessary; common methods work directly with Eq. (5) as we will see.

The Metropolis Monte Carlo Method

The Metropolis (rejection) method is a particular way of ensuring that the transition rules satisfy detailed balance. It does this by splitting the transition probability into an "a priori" *sampling distribution* $T(s \to s')$ (a probability distribution that we can directly sample) and an *acceptance probability* $A(s \to s')$ where $0 \leq A \leq 1$.

$$\mathcal{P}(s \to s') = T(s \to s')A(s \to s'). \tag{8}$$

In the generalized Metropolis procedure, (Kalos and Whitlock, 1986), trial moves are accepted according to:

$$A(s \to s') = min\left[1, q(s' \to s)\right], \tag{9}$$

where

$$q(s \to s') = \frac{\pi(s')T(s' \to s)}{\pi(s)T(s \to s')}. \tag{10}$$

It is easy to verify detailed balance and hence asymptotic convergence with this procedure by looking at the 3 cases:

- $s = s'$ (trivial)
- $q(s \to s') \leq 1$
- $q(s \to s') \geq 1$

You can avoid two common errors in Metropolis if you remember that: 1) if you can move from state s to s' then the reverse move must also be possible ($T(s \to s')$ and $T(s' \to s)$ should be zero or non-zero together) and 2) moves that are not accepted are rejected and remain at the same location for at least one more step. Accepted or rejected steps contribute to averages in the same way.

Here is the generalized Metropolis algorithm:

1. Decide what distribution to sample ($\pi(s)$) and how to move from one state to another, $T(s \to s')$

2. Initialize the state, pick s_0.

3. To advance the state from s_n to s_{n+1}:

   - Sample s' from $T(s_n \to s')$
   - Calculate the ratio:
   $$q = \frac{\pi(s')T(s' \to s_n)}{\pi(s_n)T(s_n \to s')} \tag{11}$$
   - Accept or reject:
   If $q > 1$ or if $q > u_n$ where u_n is a uniformly distributed r.n. in $(0, 1)$ set $s_{n+1} = s'$.
   Otherwise set $s_{n+1} = s_n$

4. Throw away the first κ states as being out of equilibrium where κ is the "warm-up" time.

5. Collect averages every so often and block them to get error bars.

Here is a code for a generalized Metropolis:

```
real*8 vold,vnew,sold(1:ndim),snew(1:ndim),pnew,pold
integer nsteps,istep,naccept,ndim
delta=?
call initstate(sold)              !initialize the state
vold=action(sold)
naccept=0
do istep=1,nsteps

    call sample(sold,snew,pnew,1,delta,ndim) !sample snew, pnew is prob
    vnew = action(snew)           !find the new action
    call sample(snew,sold,pold,0,delta,ndim) !find the prob of going backward
    a=exp(-vnew+vold)*pold/pnew   !acceptance ratio

    if(a.gt.sprng()) then         !accept the move
        sold=snew
        vold=vnew
        naccept=naccept+1
endif

    call averages(sold)

enddo
write (*,*) 'acceptance ratio =",float(naccept)/float(nsteps)
call pr_averages
end

subroutine sample(sold,snew,prob,ifnew,delta,ndim)
integer ndim,ifnew,idim
real*8 sold(1:ndim),snew(1:ndim),prob,delta
```

```
!classic Metropolis sampling in 1-D
   prob=1./delta
   if(ifnew.eq.1) then
       snew=sold
       idim=1+ndim*sprng()   !pick particle at random
       snew(idim)=sold(idim)+(sprng()-.5)*delta !move it
    endif
    end
```

Consider the sampling of a classical Boltzmann distribution, $\exp(-\beta V(s))$. In the original Metropolis procedure, $T(s \to s')$ was chosen to be a constant distribution inside a cube and zero outside. This is the *classic* rule: a single atom at a single time slice is displaced uniformly and the cube side Δ is adjusted to achieve 50% acceptance. Since T is a constant, it drops out of the acceptance formula. So the update rule is:

$$\mathbf{r}' = \mathbf{r} + (\mathbf{u} - \frac{1}{2})\Delta \tag{12}$$

with the acceptances based on $q = exp(-\beta(V(s') - V(s))$. Moves that lower the potential energy are always accepted. Moves that raise the potential energy are often accepted if the energy cost (relative to $k_B T = 1/\beta$) is small. Hence the random walk does not simply roll downhill. Thermal fluctuations can drive it uphill.

Some things to note about Metropolis:

- The acceptance ratio (number of successful moves/total number of trials) is a key quantity to keep track of and to quote. Clearly if the acceptance ratio is very small, one is doing a lot of work without moving through phase space. On the other hand, if the acceptance ratio is close to 1, you could probably use larger steps and get faster convergence. There is a rule-of-thumb that it should be 1/2, but in reality we have to look at the overall efficiency.

- One nice feature is that particles can be moved one at a time. Note that N steps of Metropolis takes the same amount of time as 1 step of Molecular Dynamics. Consider what would happen if we moved N hard spheres all together. Let p be the probability of getting an overlap (and hence rejection) in the move of one hard sphere. Then the probability of getting an acceptance with N hard spheres is $(1-p)^N = \exp(N \ln(1-p))$ assuming no correlation. In order to get a reasonable acceptance δ would have to be small enough so that $p \approx 1/N$ which would require extremely small steps.

- Note that we need both the forward probability and the reverse probability if one has a non-uniform transition probability. Also note that we cannot calculate the normalization of π nor is it ever needed. Only ratios enter in.

- One can show that the Metropolis acceptance formula is optimal among formulas of this kind which satisfy detailed balance. (The average acceptance ratio is as large as possible.)

- In some systems, it is necessary to have several different kinds of moves, for example, moves that change path variables and other moves that change the permutation. So it is necessary to generalize the Metropolis procedure to the case in which one has a *menu* of possible moves.

There are two ways of implementing such a menu. The simplest is to choose the type of move randomly, according to some fixed probability. For example, one can choose the particle to be updated from some distribution. One must include in the definition of $T(s \to s')$ the probability of selecting that move from the menu (unless you can argue that it cancels out.) A more common procedure is to go through all possible atoms systematically. After one *pass*, moves of all coordinates have been attempted once. In this case, the transition probability does not satisfy detailed balance but it is easy to show that composition of moves will give a random walk with π as equilibrium distribution as long as each type of move individually satisfies detailed balance. Having many types of moves makes the algorithm much more robust, since before doing a calculation one does not necessarily know which moves will lead to rapid movement through phase space.

Practical Measures of Convergence

Since asymptotic convergence is easy to guarantee, the main issue is whether configuration space is explored thoroughly in a reasonable amount of computer time. Let us define a measure of the convergence rate and of the efficiency of a given Markov process. This is needed to compare the efficiency of different transition rules, to estimate how long the runs should be, and to calculate statistical errors. The rate of convergence is a function of the property being calculated. Generally one expects that there are local properties which converge quickly and other properties (such as order parameters near a phase boundary) which converge very slowly.

Let $\mathcal{O}(s)$ be a given property and let its value at step k of the random walk be $\mathcal{O}_k$. Let the mean and intrinsic variance of $\mathcal{O}$ be denoted by

$$\bar{\mathcal{O}} = \langle \mathcal{O}_k \rangle \tag{13}$$

and

$$\sigma^2_{\mathcal{O}} = \langle (\mathcal{O}_k - \bar{\mathcal{O}})^2 \rangle \tag{14}$$

where the averages $\langle \ldots \rangle$ are over π. These quantities depend only on the distribution π, not on the Monte Carlo procedure. We can show that the standard error of the estimate of the average, $\bar{\mathcal{O}}$, over a Markov chain with P steps, is

$$error[\bar{\mathcal{O}}] = \sqrt{\frac{\kappa_{\mathcal{O}} \sigma^2_{\mathcal{O}}}{P}}. \tag{15}$$

The *correlation time*, $\kappa_{\mathcal{O}}$, defined as

$$\kappa_{\mathcal{O}} = 1 + 2 \sum_{k=1}^{\infty} \frac{\langle (\mathcal{O}_0 - \bar{\mathcal{O}})(\mathcal{O}_k - \bar{\mathcal{O}}) \rangle}{\sigma^2_{\mathcal{O}}}, \tag{16}$$

gives the average number of steps to decorrelate the property $\mathcal{O}$. The correlation time will depend crucially on the transition rule and has a minimum value of 1 if one can move so far in configuration space that successive values are uncorrelated. In general, the number of independent steps which contribute to reducing the error bar from Eq. (15) is not P but $P/\kappa_{\mathcal{O}}$.

Hence to determine the true statistical error in a random walk, one needs to estimate the correlation time just as we had to do with MD simulations. To do this it is very important that the

total length of the random walk be much greater than $\kappa_{\mathcal{O}}$. Otherwise the result and the error will be unreliable. Runs in which the number of steps is $P \gg \kappa_{\mathcal{O}}$ are called *well converged*. In general, there is no mathematically rigorous procedure to determine $\kappa_{\mathcal{O}}$. Usually one must determine it from the random walk. It is a good practice occasionally to run very long runs to test that the results are well converged. Error bars can be conveniently determined by blocking the result over enough steps so that successive blocks are independently distributed.

The correlation time defined above is an equilibrium average. There is another correlation time relevant to Markov chains, namely, how many steps it takes to reach equilibrium from some starting state. Normally this will be at least as long as the equilibrium correlation time, but in some cases it can be much longer. The simplest way of testing convergence is to start the random walk from several, radically different, starting places and see if a variety of well-chosen properties converge to the same values. A starting place appropriate for a dense liquid or solid is with all the atoms sitting on lattice sites. However, it may take a very large number of steps for the initial solid to melt. Meta-stability and hysteresis are characteristic near a (first-order) phase boundary. A random starting place is with placing each variable randomly in the total space. It may be very difficult for the system to go to the equilibrium distribution from this starting place. More physical starting places are well-converged states at neighboring densities and temperatures.

The *efficiency* of a random-walk procedure (for the property $\mathcal{O}$) is defined as how quickly the errors bars decrease as a function of computer time,

$$\xi_{\mathcal{O}} = \frac{1}{\kappa_{\mathcal{O}} \sigma_{\mathcal{O}}^2 T}, \tag{17}$$

where T is the computer time per step. Hence the efficiency is independent of the length of the calculation and is the figure-of-merit for a given algorithm. The efficiency depends not only on the algorithm but also on the computer and the implementation. Methods that generate more steps per hour are, other things being equal, more efficient. We are fortunate to live in a time when the efficiency is increasing because of rapid advances in computers. Improvements in algorithms can also give rise to dramatic increases in efficiency. If we ignore how much computer time a move takes, an optimal transition rule is one which minimizes $\kappa_{\mathcal{O}}$, since $\sigma_{\mathcal{O}}^2$ is independent of the sampling algorithm.

There are advantages in defining an *intrinsic efficiency* of an algorithm since one does not necessarily want to determine the efficiency for each property separately. It is best to optimize an algorithm to compute a whole spectrum of properties. Diffusion of paths through phase space provides at least a intuitive measure of convergence. Let us define the *diffusion constant* D_R of an algorithm by

$$D_R = \left\langle \frac{[(R_{n+1} - R_n)]^2}{T} \right\rangle, \tag{18}$$

where $R_{n+1} - R_n$ is the total change in one Monte Carlo step and T is the CPU time per step. Note that this change is zero if a move is rejected. For the "classic" Metropolis procedure we see that the diffusion constant is roughly:

$$D_R \propto \langle A \rangle \Delta^2. \tag{19}$$

Hence one wants to increase Δ until the acceptance ratio starts decreasing too rapidly. This leads to an optimal choice for Δ. The values of these diffusion constants depend not only on the computer

and the algorithm, but also on the physics. Diffusion of the atoms in a solid is much less than in a liquid, irrespective of the algorithm.

Heat Bath Rules

Usually transition rules are local; at a given step only a few coordinates are moved. If we try to move too many variables simultaneously, the move will almost certainly be rejected, leading to long correlation times. Given a transition rule, we define the *neighborhood*, $\mathcal{N}(s)$, for each point in state space as the set of states s' that can be reached in a single move from s. (It is essential for detailed balance that the neighborhoods be reflexive. If s' is in the neighborhood of s, then s is in the neighborhood of s'.) With the *heat-bath* transition rule, one samples elements from the neighborhood with a transition probability proportional to their equilibrium distribution,

$$T_{HB}(s \to s') = \frac{\pi_{s'}}{C_s}, \tag{20}$$

where the normalization constant is

$$C_s = \sum_{s'' \in \mathcal{N}(s)} \pi_{s''}. \tag{21}$$

Then one sees, by substitution into the acceptance probability formula, that the acceptance probability will be

$$A(s \to s') = \min\left[1, \frac{C_s}{C_{s'}}\right]. \tag{22}$$

If the neighborhood of s equals the neighborhood of s' then all moves will be accepted. For all transition rules with the same neighborhoods, the heat-bath rule will converge to the equilibrium distribution fastest and have the smallest correlation time. Within the neighborhood, with heat bath one comes into equilibrium within a single step.

This heat-bath rule is frequently used in lattice spin models where one can easily compute the normalization constant, C_s needed in the acceptance ratio formula and to perform the sampling. The heat-bath approach is not often used in continuum systems because the normalizations are difficult to compute; note that the integral in Eq. (21) extends over all space. In Monte Carlo on a classical system, the new atom could be anywhere in the box. One has to compute a one-particle partition function at each step. A repulsive potential will cut holes in the uniform distribution where another atom is present. Although it would be possible to develop sophisticated ways of sampling T_{HB}, it has been found more efficient to further approximate T_{HB} by some function that can be sampled quickly and let the Metropolis algorithm correct the sampling, since all that matters in the end is the efficiency. For continuum systems the idea is to find a method close to the heat-bath rule, so that the correlation time is small, but with a transition rule which is able to be executed quickly.

References:

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, 1953, J. Chem. Phys. **21**, 1087.

Allen, M. P., and D. J. Tildesley, 1987, *Computer Simulation of Liquids* (Oxford University, New York). pgs. 114-123.
Kalos, M. H., and P. A. Whitlock, 1986, *Monte Carlo Methods Volume I: Basics* (Wiley, New York). pgs. 73-86.
Hammersley, J. M., and D. C. Handscomb, 1964, *Monte Carlo Methods* (Chapman and Hall, London). pgs. 113-122.