

## Topic 3: Monte Carlo Simulation of Magnetic Materials (continued)

The Metropolis algorithm generates a Markov chain of configurations

$$x_0 \rightarrow x_1 \rightarrow x_2 \rightarrow \dots \rightarrow x_i \rightarrow x_{i+1} \rightarrow \dots$$

distributed according to a given probability function  $P(x)$ . Why does this algorithm work?

### *Equilibration and the Detailed Balance Condition*

It can be shown that any algorithm that satisfies a *detailed balance condition* will come to equilibrium. To explain this, let's introduce a fake "time"  $t$  which is simply the index of a configuration in the Markov chain. Let  $P(x, t)$  be the probability that configuration  $x$  occurs in the chain at time  $t$ . What needs to happen is for  $P(x, t)$  to become independent of  $t$  after a sufficient number of thermalization steps, i.e.,  $\lim_{t \rightarrow \infty} P(x, t) = P(x)$  which is the distribution we are trying to generate!

Now the Metropolis algorithm specifies a probability  $W(x \rightarrow x')$  to obtain the future configuration  $x'$  given the current configuration  $x$ . In one Metropolis step, the probability  $P(x, t)$  can change due to two reasons: (1) configuration  $x$  can go to some other configuration  $x'$  thus decreasing the probability, and (2) another configuration  $x'$  can go to  $x$  thus increasing the probability:

$$P(x, t + 1) = P(x, t) - \sum_{x'} P(x, t)W(x \rightarrow x') + \sum_{x'} P(x', t)W(x' \rightarrow x) .$$

This is the *master equation* for the stochastic process. The condition for equilibrium  $P(x, t + 1) = P(x, t) = P(x)$  is

$$\sum_{x'} P(x, t) W(x \rightarrow x') - \sum_{x'} P(x', t) W(x' \rightarrow x) = 0 ,$$

which will be satisfied if

$$\frac{W(x \rightarrow x')}{W(x' \rightarrow x)} = \frac{P(x')}{P(x)} .$$

This is called the *detailed balance condition*.

Let's verify that the Metropolis algorithm satisfies this condition. Suppose that  $x'$  is "uphill" compared with  $x$ . Then,

$$W(x \rightarrow x') = 1 ,$$

because the walker always accepts an "uphill" move, while

$$W(x' \rightarrow x) = \frac{e^{-E(x)/(k_B T)}}{e^{-E(x')/(k_B T)}} = \frac{P(x)}{P(x')} ,$$

because the walker will take a "downhill" step only if the change in energy is not too large.

However, if  $x'$  is "downhill" compared with  $x$ , then

$$W(x \rightarrow x') = \frac{e^{-E(x')/(k_B T)}}{e^{-E(x)/(k_B T)}} = \frac{P(x')}{P(x)} ,$$

and

$$W(x' \rightarrow x) = 1 .$$

Thus the Metropolis algorithm satisfies detailed balance. This implies that a Metropolis walker should come to equilibrium after a sufficiently long time. However, this proof does not provide an estimate of how long it might take for the Markov chain to come to equilibrium.

### *Uncertainties in simulations and experiments*

Simulations using random or *stochastic* methods are very much like experiments done on real systems. Experimental measurements involve uncertainties that cannot be avoided. Every time an experiment is repeated one obtains a different answer. Therefore it is *never* sufficient to perform a measurement only once. Measurements must be repeated a number of times to determine the most probable value and the likely deviations from this value. If a series of  $N$  “independent” measurements yield values  $x_i, i = 1, \dots, N$ , one can determine the

average (mean) value:  $\langle x \rangle = \frac{1}{N} \sum_{i=1}^N x_i$

variance:  $\sigma^2 = \langle (x - \langle x \rangle)^2 \rangle = \left( \frac{1}{N} \sum_{i=1}^N x_i^2 \right) - \left( \frac{1}{N} \sum_{i=1}^N x_i \right)^2 ,$

standard deviation:  $\sigma = \sqrt{\langle x^2 \rangle - \langle x \rangle^2} .$

The result of an experiment is expressed as  $\langle x \rangle \pm \sigma$ . Strictly speaking, this applies to random or *statistical uncertainties* in the measurement. There may also be *systematic uncertainties* which can arise for example from defects in the measuring apparatus or experimental protocol which introduce biases in the data. In a Monte Carlo simulation, a good random number generator will produce statistical errors in measured quantities. Computer simulations are also subject to systematic errors: these can be caused for example by discrete approximations or by floating point truncation errors.

### Monte Carlo Error Estimates

In a Monte Carlo computer simulation, a sequence of  $N$  configurations or sample points  $x_i, i = 1, \dots, N$  is generated using a random Markov algorithm and the average of a quantity  $f(x)$  is computed using the formula

$$\langle f \rangle_N = \frac{1}{N} \sum_{i=1}^N f(x_i) .$$

Quantities of physical interest will usually have a well-defined variance  $\sigma_f^2$

$$\sigma_f^2 = \lim_{N \rightarrow \infty} \left[ \left( \frac{1}{N} \sum_{i=1}^N f(x_i)^2 \right) - \left( \frac{1}{N} \sum_{i=1}^N f(x_i) \right)^2 \right] .$$

An extremely important result is that the statistical uncertainty in a Monte Carlo average based on a sequence of finite length  $N$  is determined by  $\sigma_f$ : the mean value and standard deviation from the mean are given by

$$\langle f \rangle_N \pm \frac{\sigma_f}{\sqrt{N}}.$$

Example: In the Monte Carlo calculation of the value of  $\pi$ , one computes

$$\pi = 4\langle f \rangle, \quad \text{where} \quad f(x, y) = \begin{cases} 1 & \text{if } x^2 + y^2 < 1, \\ 0 & \text{otherwise.} \end{cases}$$

so it is easy to see that

$$\langle f \rangle = \langle f^2 \rangle = \frac{\pi}{4}, \quad \sigma_f = \sqrt{\frac{\pi}{4} - \left(\frac{\pi}{4}\right)^2}.$$

If the variance is not known *a priori*, it can be estimated from the same Monte Carlo sequence of length  $N$  which is used to determine  $\langle f \rangle$ :

$$\sigma_f \simeq \sqrt{\left(\frac{1}{N} \sum_{i=1}^N f(x_i)^2\right) - \left(\frac{1}{N} \sum_{i=1}^N f(x_i)\right)^2}.$$

### Check of Error Estimate by Binning

A simple check on the consistency of the Monte Carlo error estimate  $\sigma_f/\sqrt{N}$  is to divide the Markov chain into a sequence of sub-chains of length  $n = N/B$ , where  $B$

is the number of sub-chains, or “bins” or “blocks”. One can compute the average in each bin:

$$\langle f \rangle_{n,b} = \frac{1}{n} \sum_{i=1}^n f(x_{n(b-1)+i}), \quad b = 1, \dots, B.$$

If the chain is truly stochastic (random), then these  $B$  averages can be considered to be independent measurements, and their variance can be computed using

$$\sigma_{\langle f \rangle_n}^2 = \left( \frac{1}{B} \sum_{b=1}^B \langle f \rangle_{n,b}^2 \right) - \left( \frac{1}{B} \sum_{b=1}^B \langle f \rangle_{n,b} \right)^2 = \frac{1}{B} \sum_{b=1}^B (\langle f \rangle_{n,b} - \langle f \rangle_N)^2.$$

because

$$\frac{1}{B} \sum_{b=1}^B \langle f \rangle_{n,b} = \langle f \rangle_N$$

by definition. This gives us another estimate of the Monte Carlo average and error:

$$\langle f \rangle_N \pm \frac{\sigma_{\langle f \rangle_n}}{\sqrt{B}}.$$

It can be shown that

$$\frac{\sigma_{\langle f \rangle_n}}{\sqrt{B}} = \frac{\sigma_f}{\sqrt{N}}$$

*only if* the configurations in the Markov chain are truly random and uncorrelated. To see this, let's define

$$g_{b,i} = f(x_{n(b-1)+i}) - \langle f \rangle_N,$$

Then,

$$\begin{aligned}
 \frac{\sigma_{\langle f \rangle_n}^2}{\sqrt{B}} &= \frac{1}{B^2} \sum_{b=1}^B \left[ \frac{1}{n} \sum_{i=1}^n g_{b,i} \right] \left[ \frac{1}{n} \sum_{j=1}^n g_{b,j} \right] \\
 &= \frac{1}{B^2 n^2} \sum_{b=1}^B \sum_{i=1}^n g_{b,i}^2 + \frac{1}{B^2 n^2} \sum_{b=1}^B \sum_{i=1}^n g_{b,i} \sum_{\substack{j=1 \\ j \neq i}}^n g_{b,j} \\
 &= \frac{\sigma_f^2}{N} + \frac{1}{N^2} \sum_{b=1}^B \sum_{i=1}^n g_{b,i} \sum_{\substack{j=1 \\ j \neq i}}^n g_{b,j} .
 \end{aligned}$$

In the second line we have separated the terms with  $i = j$  from those with  $i \neq j$ . The second term involves a sum over products  $g_{b,i}g_{b,j}$  with  $i \neq j$ . If the Monte Carlo sequence is truly random, then these products will be randomly distributed about  $\langle g \rangle = 0$  with equal probability of being positive or negative. Thus the second term will be negligible compared with the first in which all the contributions are positive.

However, if there are significant correlations in the Monte Carlo sequence, then the second term involving products  $g_{b,i}g_{b,j}$  with  $i \neq j$  will *not* be negligible, and the two estimates of the Monte Carlo error will not agree. If this happens, the simulation must be modified to reduce these correlations.