Monte Carlo Methods - Lecture 1

Introduction

In statistical physics Monte Carlo methods are considered to have started in the Manhattan project (1940's) during which J. von Neumann, S. Ulam, N. Metropolis and R. Feynman used random numbers to estimate neutron scattering and absorption rates in materials. The Metropolis method which you will use in the first coding assignment originated in the paper: "Equation of state calculations by very fast computing machines", by N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, M. Teller and E. Teller, J. Chem. Phys. 21. 1087 (1953). A recent retrospective "Marshall Rosenbluth and the Metropolis algorithm", Physics of Plasmas 12, 57303 (2005) describes some of the history and makes it clear that Metropolis actually had little to do with the development of what is now called the Metropolis method. The credit should really go to Rosenbluth and Teller. We will discuss later a form of the Metropolis algorithm which works for polymer chains. This polymer algorithm is called the Rosenbluth algorithm.

The Metropolis method is the most widely used Monte Carlo method. Monte Carlo methods can be considered to be improvements or variations on other random sampling methods such as importance sampling, which have a longer history. They are also the most widely used example of a computational method where we try to deduce the typical behavior of a complex system by repeated stochastic simulations. A few examples of the use of Monte Carlo methods in physics are:
(i) Thermodynamic quantities in statistical physics (e.g. Ising model magnetization)
(ii) Multiple integrals in mathematics and analysis
(iii) Reaction-diffusion problems in chemical engineering and geophysics
(iv) Polymer or protein conformations
(v) Particle showers in Nuclear and HEP
Note that MC methods do not replace a theoretical understanding of the problem. Instead a deep theoretical understanding is required before the correct MC simulation can be set up. Actually the literature is full of incorrect simulations, due to either poor input physics or poor quality data analysis. MC methods could take up an entire course in any one of the areas listed above and there are a large number of special purpose MC codes. The mathematics of the method has also attracted a lot of effort where the Metropolis method is often called Markov chain Monte Carlo, for reasons which will become clear once we define the Metropolis method and a Markov chain.

A first example - Random sampling or Monte Carlo to find π

Of course we can find π by measuring the circumference of a circle and dividing by the diameter, however can we use a random sampling method to find it and how efficient is random sampling? Buffon (1777) constructed the following procedure (algorithm): (i) On a flat table, draw a set of parallel lines separated by distance L. (ii) Find a very thin needle of length $l < L$. (iii) Randomly throw the needle onto the plane N times. (iv) Count the number of times, n, the needle crosses a line. We define x to be the modulus of the angle between the needle axis and the horizontal axis. The probability density of finding a particular angle is $\rho(x) = 1/\pi$. Now consider throwing needles onto the plane. The probability of a needle hitting a line, for a given angle x, is given by

$$p(x) = \frac{l}{L}|cos(x)|. \tag{1}$$

The total probability of a hit is found by integrating $p(x)$ over all angles,

$$P = E[p(x)] = \int_0^\pi p(x)\rho(x)dx = \frac{2l}{\pi L} \tag{2}$$

From the experiment we measure the probability of a hit to be n/N, therefore,

$$P = \frac{n}{N} = \frac{2l}{\pi L}. \tag{3}$$

Since we know l, L, N and we measure n, we can find π from the formula. OK great, but how efficient is this procedure. That is, how many times do we have to throw the needle to find an accurate value of π. There are clearly issues to do with the accuracy of measurment of l and L and the thickness of the needle and the lines. Two important issues are:
(i) How random is our tossing of the needle - random number generators
(ii) How does the accuracy of π depend on N for an ideal random process - the central limit theorem.
We will return to both of these issues later in the course.

Monte Carlo in Statistical Physics e.g. The square lattice Ising model

Monte Carlo methods are used to find the average properties of classical and quantum mechanical many body systems. In quantum mechanics, the average value of an operator $\hat{O}$ is found using

$$< \hat{O} >= \frac{tr(\hat{O}e^{-\beta\hat{H}})}{tr(e^{-\beta\hat{H}})} \tag{4}$$

where $\hat{H}$ is the Hamiltonian and $\beta = 1/(k_B T)$ with k_B Boltzmann's constant and T the temperature. In classical physics, the operator is diagonal and the trace (tr) reduces to a sum over the energies or eigenvalues of the system leading to the classical expression,

$$< O >= \frac{\sum_c O_c e^{-\beta H(c)}}{\sum_c e^{-\beta H(c)}} \tag{5}$$

where c labels a configuration or eigenfunction of the system.

In many body problems, the number of available configurations N_c grows at least exponentionally with the number of particles N. For example if we consider a problem of electron spins that can either be up or down, the number of configurations grows as $N_c = 2^N$. If we consider a gas of particles the number of configurations is $V^N(\Delta P)^N/h^3$. It is impossible to exhaustively enumerate all possible configurations, except in the simplest problems, and Monte Carlo methods provide a procedure for finding the average properties by sampling a small fraction of configuration space - the important configurations. To illustrate this we shall use an example from magnetism, the Ising model.

The Ising model was invented to provide an understanding of phase transitions in ferromagnets, where it is known that mangetism is lost for $T > T_c$ where T_c, is the Curie temperature. The behavior near T_c is complicated and co-operative so that there is actually a singular behavior in many physical properties at T_c. In addition the geometry of the magnetic domains at T_c is fractal. All of these properties can be deduced by using the Monte Carlo method to find the average properties of Ising magnets. The Hamiltonian of a spin 1/2 Ising magnet is given by,

$$H = -J \sum_{<ij>} S_i S_j \tag{6}$$

where $J > 0$ is a ferromagnetic exchange constant, $S_i = \pm 1$ is a spin variable, and the sum is over nearest neighbor sites on a lattice. We take the square lattice as our example. Our calculation will focus upon an operator or quantity that most clearly parallels that of a real magnet, the magnetization, $m = (\sum_i < S_i >)/N$. We use the Monte Carlo method to find the average value of this quantity. Now it is time to introduce the first project.

Project 1

Write a Fortran 90 code to carry out a Monte Carlo simulation of a nearest neighbor, spin 1/2 Ising model on a square lattice. Assume that the interactions are only with the nearest neighbors and that there is no applied magnetic field. Write the code so that the cell that you treat is square and has L sites in each direction. Use free boundary conditions. Your project report should include: (i) A copy of your code; (ii) A plot of the lattice averaged magnetization as a function of time, for at least three values of the temperature. Also do the calculation for two different initial conditions: all spins up; and random spins. (iii) A plot of the time and lattice averaged magnetization as a function of temperature for a lattice of size at least $L = 20$ - starting with all spins up. You should label the axes of your graphs and also write a brief report describing the model, monte carlo procedures, the Metropolis method in particular, and a description of your results.

Outline of method

First set up an array to store the spin at each site of a square lattice. Initialize the lattice so that all spins are up, or are random. Now develop a code to carry out the following Metropolis algorithm:
(i) Choose a spin at random.
(ii) Calculate the energy change δE produced by changing the orientation of that spin.
(iii) Choose a random number, $X \epsilon [0, 1]$.
(iv) Do Metropolis test: If $X < Exp[-\delta E/k_B T]$, then change the orientation of the spin, otherwise leave it alone.
(v) If the number of iterations is reached, exit. Otherwise, return to (i).

In the next lecture we will see why this procedure correctly samples the Boltzmann distribution of Eq. (5) and how we can calculation the magnetisation using the configurations that we sample.