

Topic 3: Monte Carlo Simulation of Magnetic Materials

The two major consumers of supercomputing resources are Monte Carlo (MC) simulation program and Partial Differential Equation (PDE) solvers.

Monte Carlo methods are very versatile and can be applied to solve many different types of problems. The major drawback of MC methods is that they converge *very slowly*: MC error estimates scale like $\mathcal{O}(\sqrt{N})$ where N is the number of configurations generated, i.e., the length of the simulation run. Let's say we have achieved an accuracy of 0.01, i.e., 1%, using computer time T . To increase the accuracy to 0.001, i.e., 0.1%, we would need to increase the run time to $100T$! Clearly, parallel processing might help in this situation.

Monte Carlo computation of π

This application illustrates some important Monte Carlo concepts. The “system” being simulated here is the value of π , which is represented by the “model”:

$$\pi = \frac{4 \times \pi R^2}{(2R)^2} = \frac{4 \times \text{Area of a circle}}{\text{Area of enclosing square}} .$$

Areas are estimated by counting points chosen at random in the enclosing square.

Configuration or Sample: One sample or configuration is one point chosen at random. A point is a pair of real numbers:

```
x = ((double) rand() / RAND_MAX) * 2 - 1;  
y = ((double) rand() / RAND_MAX) * 2 - 1;
```

Why use random numbers? Why don't we just count all the configurations??

Configuration Space: This is the set of all possible configurations. Also called the *sample space* or *phase space*. Mathematically there are an infinite number of points in the enclosing square. Computationally the number of points $\leq M^2$ where M is the period of `rand()`. Note: the number of `double`'s is also finite $\sim 2^{64}$, but *not* uniformly distributed!

Probability Distribution or Weight Function: Configurations are generated according to the probability distribution

$$p(x, y) = \begin{cases} 1, & \text{if } \sqrt{x^2 + y^2} < 1 \\ 0, & \text{otherwise} \end{cases}$$

Monte Carlo Average: Generate N *statistically independent* configurations (x_i, y_i) , $i = 1, \dots, N$. Then

$$\frac{\pi}{4} = \frac{1}{N} \sum_{i=1}^N p(x_i, y_i) = \frac{\text{No. of points inside circle}}{\text{Total number of points}}.$$

The 2-dimensional Ising Model

Magnetism in matter is caused by charged particles moving in closed orbits or spinning around their axes. Recall that a current loop creates a magnetic field according to Ampère's law. A spinning charged particle has a magnetic moment associated with it.

A simple classical approximation to an atomic or electronic magnetic moment is provided by an Ising spin which can take two values

$$s_i = \begin{cases} +1, & \text{represents "spin up"} \\ -1, & \text{represents "spin down"} \end{cases}$$

A two-dimensional magnet can be modeled by a set of N_s spins located on a fixed two-dimensional lattice of sites. For example, we can have a square lattice with L spins in the x direction and L in the y direction such that $L^2 = N_s$.

The force between two magnets falls off rather rapidly, like r^{-3} . So a reasonable approximation is to assume that any spin interacts only with its 4 nearest neighbors—north, south, east and west. The interaction energy can be approximated by

$$E = -J \sum_{\langle ij \rangle} s_i s_j - H \sum_i s_i .$$

If the interaction strength $J > 0$ the system is ferromagnetic: the energy is minimized if the spin point in the same direction $s_i s_j = +1$. If $J < 0$ the system is antiferromagnetic. H represents an external magnetic field which couples to the *magnetization*

$$M = \sum_i s_i ,$$

of the system. The spins prefer to line up with the magnetic field.

Ferromagnetism, Paramagnetism, and Curie Temperature

If $H = 0$, the system can exist in two different phases depending on the temperature T . At low temperatures, the system is permanently magnetized. At sufficiently high temperatures, the magnetization of the system is zero. There is a critical value of the temperature T_c called the *Curie temperature* at which a *phase transition* between the ferromagnetic (permanently magnetized) and paramagnetic phases occurs.

Monte Carlo Simulation

The properties of the model can be measured using Monte Carlo methods similar to the calculation of π .

Configuration or Sample: One sample or configuration of the magnet is a particular assignment of spin values, say

$$s_1 = +1, s_2 = -1, s_3 = +1, \dots, s_{N_s} = +1,$$

in which each spin is set “up” or “down”. According to statistical mechanics, the average value of an observable is got by weighting each configuration with the *Boltzmann factor*. For example, the average magnetization at some fixed temperature T is given by

$$\langle M \rangle = \frac{\sum_{\text{configs}} M e^{-E/k_B T}}{\sum_{\text{configs}} e^{-E/k_B T}}.$$

Configuration Space: The total number of configurations of the system is enormous even for small numbers of spins. For example if $L = 20$, $N_s = 20^2 = 400$, and

$$\text{No. of configs} = 2^{N_s} = 2^{400} = 2.58 \times 10^{120} .$$

If we tried to enumerate the configurations at a billion per second on a very fast computer, it would take 2.58×10^{111} seconds = 8.8×10^{103} years to compute the average magnetization exactly!

Probability Distribution or Weight Function: The basic idea of a Monte Carlo calculation is to generate a reasonable number of configurations at random. The Boltzmann factor is an exponential function of energy which can vary enormously. The random configurations are therefore generated with probability determined by this exponential factor:

$$p(s_1, s_2, \dots, s_{N_s}) = \frac{e^{-E(s_1, s_2, \dots, s_{N_s})/k_B T}}{\sum_{\text{configs}} e^{-E/k_B T}} .$$

Monte Carlo Average: The problem now is to generate N statistically independent configurations that are distributed according to the Boltzmann factor:

$$(s_1^{(i)}, s_2^{(i)}, \dots, s_{N_s}^{(i)}), \quad i = 1, 2, \dots, N ,$$

the average magnetization and energy are given by

$$\langle M \rangle = \frac{1}{N} \sum_{i=1}^N M(s_1^{(i)}, s_2^{(i)}, \dots, s_{N_s}^{(i)}) ,$$

$$\langle E \rangle = \frac{1}{N} \sum_{i=1}^N E(s_1^{(i)}, s_2^{(i)}, \dots, s_{N_s}^{(i)}) .$$

Algorithm of Metropolis et al.

How does one generate configurations distributed according to the Boltzmann factor? A very ingenious algorithm to do this was discovered by N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, *J. Chem. Phys.* **21**, 1087 (1953). Applied to this model, the algorithm generates the next configuration in the sequence as follows:

- Given a configuration, choose a spin s_i , let $s_{i,\text{trial}} = -s_i$.

- Compute the change in energy of the system

$$\Delta E = E(s_1, s_2, \dots, s_{i,\text{trial}}, \dots, s_{N_s}) - E(s_1, s_2, \dots, s_i, \dots, s_{N_s}) .$$

- If

$$e^{-\Delta E/k_B T} > r ,$$

where r is a uniform random deviate let $s_i \leftarrow s_{i,\text{trial}}$, flip this spin.

- Repeat the above for all N_s spins in the configuration.

Starting from some initial configuration, a suitable number of Monte Carlo steps are taken to allow the system to thermalize. After this the production phase begins and statistical averages are measured.