PHYS-E0412 Computational Physics :: Homework 4

Due date 5.2.2019 at 10 am

Simulated annealing of the random bond Ising model

Simulated annealing[1] is a method that can be used to find ground states of systems whose potential energy landscape is so complex that simple deterministic methods fail. The specific model that we study here is a so-called random bond Ising model on the square lattice. The energy of this model is defined similarly as for the usual square lattice Ising model,

$$H = -\sum_{<i,j>} J_{ij} s_i s_j, \tag{1}$$

where the variables s_i take values ± 1, and the sum goes over distinct nearest neighbour pairs. However, each coupling constant J_{ij} has probability p to be -1 and $1-p$ to be $+1$. Thus we have actually defined a large number of Ising models, and the ground state depends on the specific realization of the coupling constants. However, we can study expectation values of ground state quantities over all realizations of the model for a specific p. Thus, for example, we can calculate the expected ground state magnetization $\langle m \rangle$, where

$$m = \frac{1}{N_s} \left| \sum_i s_i \right|, \tag{2}$$

where N_s is the number of sites on the lattice. Another relevant quantity is the staggered magnetization $\langle m_s \rangle$, where

$$m_s = \frac{1}{N_s} \left| \sum_i \mathrm{sgn}(i) s_i \right|, \tag{3}$$

and $\mathrm{sgn}(i) = \pm 1$ so that neighbouring sites always have opposite values of $\mathrm{sgn}(i)$[2]. We consider lattices of size $L \times L$ with periodic boundary conditions. The lattice of size $L = 4$ is a good starting point, but try to reach as large lattices as possible within reasonable running times. Note that lattices with odd L are somewhat anomalous (think about why), so it is best to use even values L.

The simulated annealing procedure we use here is based on the Metropolis Monte Carlo simulation of the Ising model. The idea is simple: We start running the Monte Carlo algorithm at a relatively high temperature, and then gradually lower it to zero, or at least to some very small value. What is "large" and "small" in this case can be estimated for example by considering the known critical temperature $T_c \approx 2.27$ of the model when all of the coupling constants are equal to 1. Thus one should start the annealing procedure from above this temperature, and end up at some temperature well below it. The simplest (though not necessarily the most efficient) way to do this is to just linearly ramp down the temperature as a function of the number of Monte Carlo updates performed. There is a tradeoff between the speed of this ramping and the quality of the results produced: If T is decreased very quickly, the system might get stuck in some metastable state, but decresing it very slowly will take more time.

[1] `https://en.wikipedia.org/wiki/Simulated_annealing`
[2] `https://en.wikipedia.org/wiki/Antiferromagnetism`

a) Write a simulated annealing code for the model. Test your code by considering the case $J_{ij} = 1$ for all nearest neighbour pairs $\langle i, j \rangle$, i.e. $p = 0$. Starting the annealing process from a random initial configuration, you should (nearly) always find the completely magnetized ground state. What is the ground state in the case $p = 1$? (2p)

b) Calculate the expectation values of the magnetization and staggered magnetization as a function of p. For a single value of p, you should calculate the ground state N times (e.g. in the range 10 to 100) for different realizations of the coupling constants, obtaining the value of m and m_s for each ground state. The expectation value is then estimated by the mean of the obtained values. Also estimate the error of the expectation values using the expression $\sigma/\sqrt{N}$, where σ is the sample standard deviation. (2p).

c) It is known that for $L \to \infty$ the model has a critical point at $p = p_c \approx 0.12$. For $p < p_c$ the magnetization $\langle m \rangle \neq 0$, and for $p > p_c$ $\langle m \rangle = 0$. Based on your results and intuition (or an analytical proof if you wish), can you guess the location of another critical point and what happens there? (1p)

Return your solutions to MyCourses. Please remember to return your codes as well.