Homework 6; due Thursday, November 21

PY 502, Computational Physics, Fall 2024

Department of Physics, Boston University

Instructor: Anders Sandvik

EQUILIBRATION TIME SCALING IN THE 2D ISING MODEL

In this assignment you will carry out Monte Carlo simulations of the 2D ferromagnetic Ising model,

$$E = -\sum_{\langle i,j \rangle} \sigma_i \sigma_j, \tag{1}$$

where the sum is over nearest-neighbors on the square lattice with $N = L^2$ spins. There is no external magnetic field. We will focus on the equilibration properties when starting a simulation from a very atypical state (fully polarized) at the given simulation temperature. You can use and modify the program available on the course web site, or write your own from scratch.

Relaxation to equilibrium and magnetization reversals

The initial state used in a Monte Carlo simulation is normally not a typical state of the equilibrium distribution at the given model parameters. Therefore, some time (number of update cycles, what we call Monte Carlo steps) have to be devoted to equilibrate the system before physical observables are "measured" on the generated configurations. The time-scale of the equilibration can depend strongly on the model parameters. In this assignment you will investigate characteristic time scales of the Ising model as it approaches equilibrium when starting from a very atypical state; the one with all spins set to "up" (fully polarized). You will study the magnetization,

$$m = \frac{1}{N} \sum_{i=1}^{N} \sigma_i, \tag{2}$$

as a function of the simulation time.

Fig. 1 shows m versus the simulation time, measured in Monte Carlo steps (each step consisting of N spin-flip attempts using the Metropolis algorithm) in three independent runs, in each case starting from the fully polarized state. The temperature $T = 2$ is below the critical temperature ($T_c = 2\ln^{-1}(\sqrt{2} + 1) = 2.2691853142130...$) and we can see that the magnetization typically fluctuates about the values $m \approx \pm 0.9$. Occasionally there is a "rare event" in which the magnetization goes through 0 and reverses, and also one can see cases where m comes close to 0 or goes through 0 but quickly fluctuates back to typical magnetizations on the side of 0 where it came from. Note that here the system is already quite strongly ordered, i.e., the typical m values are close to 1. As we move closer to T_c the magnetization will fluctuate closer to 0.

By repeating such simulations from the same starting state (here with $m = 1$) many times with different streams of random numbers, one can investigate the average time dependent magnetization $\langle m(t) \rangle$ and how it approaches 0. This will be part 1) of this assignment. We can also investigate the average time it takes for the system to first reach a state with $m = 0$ (which can strictly happen only if the system volume N is even, i.e., the length L is even, which we will assume here), which is
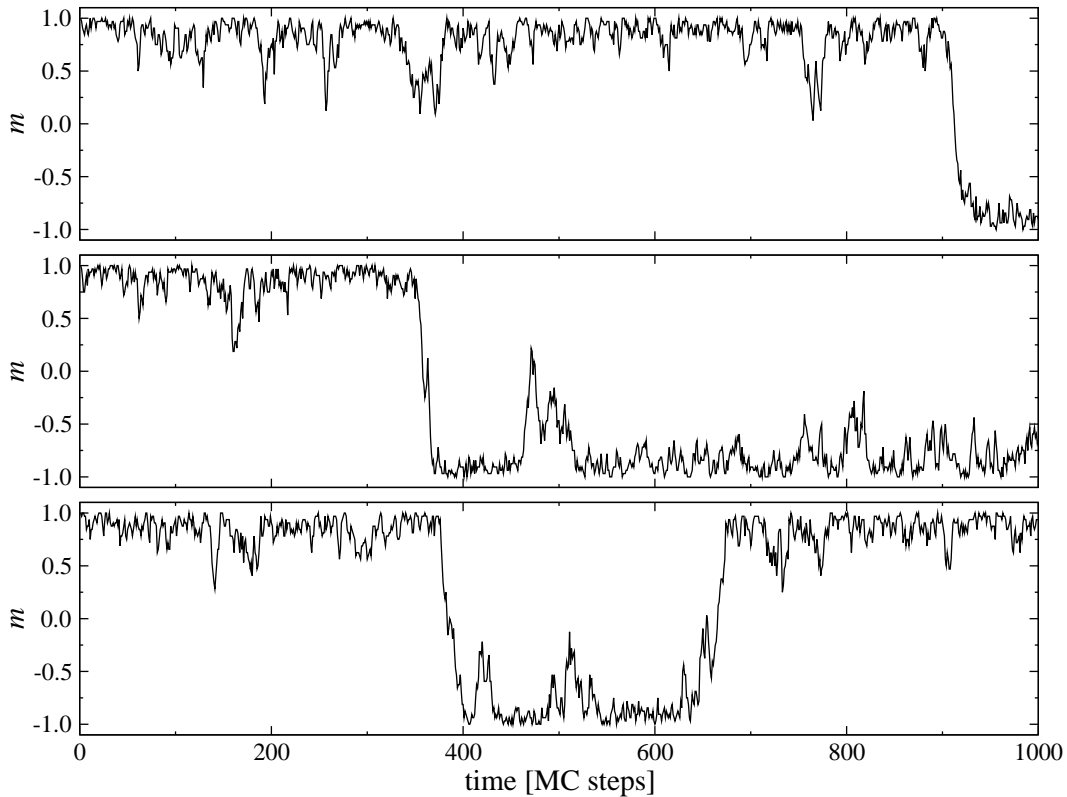
Figure 1: Magnetization as a function of simulation time in three different runs (i.e., with different sequences of random numbers) for the 2D Ising model with $N = 8^2$ spins at temperature $T/J = 2$, in each case starting with the fully polarized ($m = 1$) configuration.

part 2) of the assignment. Up to a factor $1/2$, the average time to reach $m = 0$ should be similar to the time-scale of magnetization reversals, since from an $m = 0$ state the system will move to $m < 0$ or $m > 0$ with equal probability. It will not be exactly equal to the time scale of magnetization reversals defined in the equilibrium, but, as you should be able to see in your results from part 1), the differences when measuring it from the initial $m = 1$ state should be very small.

Programming task

Write a 2D Ising Metropolis simulation code (use elements of the code on the course web site if you like) where the following input data are read from a file `read.in` (or using some other appropriate means to input the data, e.g., if you are working with Jupyter notebook):

`L,T,bins,reps,steps`

Here `L` (integer) is the system length, `T` (real) the temperature, `bins` the number of simulation bins for which averages are written out, `reps` the number of times the equilibration process (simulation) is repeated for each bin, and `steps` the number of Monte Carlo steps used in each simulation. For reference, in Fig. 1 we have `steps=1000` and `reps=3`.

The same program should be used for both part 1) and part 2) of the assignment. For part 1) the average magnetization as a function of simulation time t (Monte Carlo steps) should be computed for $t = 1, 2, \ldots,$ steps. For part 2), we want to let a simulation run until m reaches 0 and therefore the number of steps is not predetermined. In this case the input number of steps should be steps=0, and each time $m = 0$ is found a new simulation is started.

In both parts, the spins should be initialized to all $+1$ before each of the simulations. The total magnetization M (not divided by N; $m = M/N$) is then $M = N$. When the Metropolis spin flips process is carried out, M should be updated each time a spin has been flipped; $M \to M + 2\sigma_i$ after $\sigma_i \to -\sigma_i$. Each simulation should use a different random number stream, which is automatically accomplished when only initializing the random number generator once in the beginning of the program (which does not even have to be done if you use the built-in Julia random number geerator rand).

For part 1) (i.e., if steps is non-zero), after each Monte Carlo step of N spin-flip attempts, the current magnetization $m = M/N$ should be added to a vector of length steps in which the average magnetization over the reps simulations is accumulated; if we call the vector mag, then mag(t) will be used for computing the average magnetization after t Monte Carlo steps. When a bin is completed, write the averages to a file res.dat with lines in the format $t, \langle m(t) \rangle$, for all $t = 1, 2, \ldots, steps$. Each new set of bin data should be appended to the results previously written to the file. The average over the bins and the error bars are computed with the program res.f90 supplied with the assignment on the course web site. Remember to reset the data array mag() to zero ahead of each new bin.

For part 2), a simulation should be interrupted once $M = 0$, whereupon the time taken should be accumulated in order to compute the average time. If the $M = 0$ event happens after the i:th spin-flip attempt during Monte Carlo step t, the time taken is $t - 1 + i/N$. The average time over the reps repetitions should be written to the file res.dat after each bin. Note that in this mode of running the program (steps=0) the average magnetization is not computed, only the average time to reach $M = 0$.

NOTE: For this assignment you have to use the SSC cluster (even though in principle it may be doable on your own computer).

Simulations and analysis, part 1)

Time dependent magnetization: Carry out simulations for systems sizes $L = 4, 6, \ldots, 16$ at $T = 2$ (i.e., below T_c), with at least bins=100, reps=1000, steps=500. To get smaller error bars you can increase bins and reps, but it should not be necessary. You can also extend to larger steps if you like.

Plot the magnetization for all t from 1 to steps (you can also include $m = 1$ at $t = 0$) on a lin-log scale (useful for detecting exponential decays) and also produce another plots focusing on the short-time behavior graphed on a lin-lin plot. Based on these plots you should be able to detect and reason about two different time scales. Discuss these time scales and how they depend on the system size. You don't have to do any data fits, just discuss the physical meaning of the time scales and roughly what they are in this case.

Simulations and analysis, part 2)

Carry out simulations and study the average time to reach $M = 0$ for the temperatures $T = 2.0, 2.1, 2.2$ below T_c, as well as exactly at T_c (the value given in the introduction), and above T_c at $T = 2.5$. In each case use system sizes $L = 4, 6, 8, \ldots$, as high as you can go (or as high as you need to go in order to draw reliable conclusions). Do it systematically and keep track of the results and the actual running times. For low T the time to reach $M = 0$ will typically be very long, leading to long run times and limiting how far you can go (for $T = 2.0$ you don't have to go above $L = 12$, for higher T you should be able to go to larger systems.)

Look at your data with different kinds of plots; lin-lin, lin-log (to detect exponential forms), log-log (to detect power laws), and log-lin (to detect possible logarithmic divergences). Try to reason about your results and present plots that best illustrate your conclusions about the behavior in different temperature regions.