Stat Mech with Python: tutorial project

November 2021

As a part of the final exam for the course of statistical mechanics with Python you must submit your work on a tutorial project described below before the 14th December 2021. The project consists of a programming exercise along the lines of those done in class.

The solution of the Python programming exercises must be implemented in a Google Colab notebook and sent by email to epanizo1@ictp.it. Note that you can work in a small group, but the final solution must be submitted individually: each of you should send their own notebook. Questions about the implementation of the code you sent will be asked during the oral examination, so make sure you understand what you do even if not working alone on that part.

The programming parts regard a 2D Potts model (Sec 1), which is similar to the 2D-Ising model that was implemented in the class and uploaded to your cloud folder. Be careful, because these models are not identical and some care must be used to implement the differences!

1 Problem: The 2D Potts model

In class we have simulated the 2D Ising model: We considered a two-dimensional lattice with $N = L \times L$ *spins* which could have only two values, $s \in \{-1, +1\}$ and interacted only with its nearest neighbours.

The standard Potts model is a generalization where the *spins* are now called *colors* and can take up to q different values $s \in \{1, \ldots, q\}$.

The Hamiltonian of this system, specified by a coupling energy constant J and a magnetic field h reads:

$$\mathcal{H}(\vec{s}) = -\sum_{(i,j)} J\delta(s_i, s_j) - \sum_i h\delta(s_i, 1)$$

Here i, j are lattice positions with $i = (i_x, i_y)$, $j = (j_x, j_y)$, $0 < i_x, i_y, j_x, j_y \leq L$.

The sum over (i, j) is the sum over nearest neighbors only.

$\delta(s_i, s_j)$ is a Kronecker delta, whose value is 1 only if $s_i = s_j$, i.e. the two sites have the same color value.

$\delta(s_i, 1)$ is a Kronecker delta, whose value is 1 only if $s_i = 1$, i.e. the site s_i has a color value of 1 (you can imagine it as a special *color/direction* which is aligned to the external magnetic field).

For this exercise, as in the Ising exercise, we impose periodic boundary conditions, i.e. $(i_x, i_y) = (i_x + L, i_y + L) = (i_x, i_y + L) = (i_x - L, i_y) = \dots$.

2 Programming part

The aim of the project is to set up a Metropolis Monte Carlo simulator for a Potts model of *any* number of colors q.

- Write a function that takes as arguments a 2D array of $L \times L$ spins, $\vec{s} = \{s_{1,1}, \dots, s_{L,L}\}$ ($s \in 0, 1, \dots, q$), the coupling constant J, and the magnetic field h and returns the energy of that configuration $\vec{s}$ (be careful about the boundary condition).

- Write a function that compute the value of the magnetization m_1 of the sequence $m_1(\vec{s})$ defined as the percentage of the sites which are in the $q = 1$ color state (normalized between 0 and 1).

- Check that for a lattice of homogeneous color (all sites with $s = 1$, or $s = 2$ etc. etc.) and $h = 0.1$ the first function that you wrote returns the energy $-J * N * 2 - N * h$ for the case of $s = 1$ and $-J * N * 2$ for all other cases

- Check that for a lattice of homogeneous color (all sites with $s = 1$, or $s = 2$ etc. etc.) the second function shows a magnetization $m_1 = 1$ for the the $s = 1$ color and $m_1 = 0$ for all other colors.

- Check that for a sequence of randomly generated lattices with number of colors q, the energy is distributed around $-J * N * 2/q - N * h/q$ and the magnetization is approximately $1/q$ (check for a few values of q).

2.1 Metropolis Markov chain

To generate samples from the canonical probability distribution over the microstates $\vec{s}$ you will write the basic Metropolis algorithm. The first step is to generate a transition between two configurations. The Metropolis algorithm prescribes that, given $\vec{s}$, you choose a single lattice site, say $\bar{i}$ and randomly change its color to a different one, generating a candidate configuration $\vec{s}'$, where $s'_{\bar{i}}$ can have any of the colors other than the original one: $s'_{\bar{i}} \in \{1, 2, \dots, q\}/s_{\bar{i}}$. Then the energy difference between the two configuration is computed, let us call it $\Delta E = \mathcal{H}(\vec{s}') - \mathcal{H}(\vec{s})$. If $\Delta E \leq 0$, accept the new candidate, if not, accept the candidate only with probability $p = \exp[-\beta \Delta E]$.

- First, write a formula to compute ΔE and test it. Check that, given $\vec{s} = ((1, 2, 2), (2, 1, 1), (2, 1, 1))$, $\vec{s}' = ((2, 2, 2), (2, 1, 1), (2, 1, 1))$, $J = 1$, $h = 0.1$ you must obtain $\Delta E = -1.9$.

- Write a function that takes as arguments a configuration, $\vec{s}$, the number of possible colors q, the coupling constant J, the magnetic field h and the inverse temperature β, and returns a new configuration according to the Metropolis prescription.

- Iterate the function above for some steps starting from a random initial configuration of $N = 20 \times 20$ spins. Check that, when $q = 2$, $h = 0.1$ and choosing a small temperature $\beta = 10$, you end up in a configuration where all the sites have color 1 after enough iterations.

2.2 Generating samples and computing the magnetization

You know that the Markov chain of states obtained through iteration of the previous function does not generate reliable samples. Indeed, you need to *i)* equilibrate the chain and *ii)* also discard a lot of samples to remove correlation.

By using properly generated samples, you can compute the average absolute magnetization over them: $\langle m \rangle = \frac{1}{N_{samples}} \sum_{\vec{s}} |m(\vec{s})|$, where $m(\vec{s}) = \sum_i \delta(s_i, 1)/N$.

- Write a function that generates samples of energy and magnetization with the Metropolis algorithm. The function takes as argument the length of the chain L, the inverse temperature β, the number of colors q, the coupling constant J, the magnetic field h, the equilibration time t_e and the number of samples $N_{samples}$. It returns the list/array of the $N_{samples}$ generated samples. Remember to discard the samples for the equilibration. Each time you get a new good sample add it to a list. Be careful that when you append it you should shallow-copy it (see https://docs.python.org/3/library/copy.html).

3 Numerical investigations

The Potts model presents different phase transitions for different number of colors q. For $q \leq 4$ the model presents the same transition as the Ising model, i.e. a continuous transition between an homogeneous (ferromagnetic) state at low temperature (large β) and a random-mix state at large temperature. The exact temperature of the transition at infinite size for $h = 0$ is $\beta_c(q) = ln(1+\sqrt(q))/J$.

For $q > 4$ the transition becomes first order, so there is a discontinuity (in the $N \to \infty$ limit only) in the energy as the temperature cross the critical value $\beta_c(q)$.

- Take $J = 1$ and a small value of $h = 0.05$ (only to break the symmetry in color). **Study the average energy and magnetization for different temperatures crossing the critical values at number of color $q = 2$ and $q = 5$ and plot them**. In both case, to focus the data points for inverse temperatures β close to the transition in order to distinguish their behavior. (Notice that the transition may occur at a different temperature respect to the infinite case with $h = 0$, so try and scan values of βs first to identify where it lies.)

- For both values of q measure the correlation time for the energy and magnetization at a inverse temperature $\beta > \beta_c$. Use these value to estimate the error on average magnetization and energy from the variance. (Remember: $\delta^2 = \sigma^2/N_{corr}$, where N_{corr} is the number of samples corresponding to the correlation time.) **Add the errors to the plot above**.

Note: The correlation time and the variance increases with the size L and with number of colors q, so start with small values of L and use more and more steps with increasing numbers of colors

3.1 (Really) optional questions

- If $h = 0$ then the system may spontaneously orient to any color. How can you define the magnetization such that it always give 1 for a homogeneous system irrespective of the color? (i.e. doing the equivalent of the absolute value for the Ising magnetization).

- Study the correlation times of magnetization and energy depend on the temperature? Where are they largest and why?

- Changing one spin for MC-step is extremely inefficient, since the two configurations before and after are very correlated. Can you imagine/find in the literature a different change proposal mechanism that could speed up the decorrelation? What condition should it respect?