Computational Physics, Project 4
Metropolis algorithm for the Ising model

Ena Salihovic

Jacobs University Bremen

Due March 31st, 2022

1 Background - Ising model

Approach: interaction between the system and its environment is simulated by random number generator (Monte Carlo method), in order to investigate the Ising model of magnetism.

Magnetism is a quantum phenomenon. Ferromagnetism cannot be described using classical approach. Origin of magnetism are electron's spin and magnetic moment. Ferromagnetism occurs when a collection of electrons' spins cooperate such that their magnetic moments all point in the same direction, yielding a total macroscopic magnetic moment [1]. It is also important to investigate how the magnetic properties depend on temperature, since systems generally lose their magnetism at high temperatures.

Now, consider a collection of magnetic moments, denoted by arrows, each representing an atop of spin $\frac{1}{2}$. Assume that these are situated on a 2D lattice. Since, fully simulating a quantum mechanical system with the requirements of including quantum rules for dealing with spin angular momentum etc, would be too complex of a computation, here a few simplifications and assumptions are made.

Assume only 2 possible orientations of the spin: +z (up) and -z (down) direction. Hence, the i^{th} spin in the system can be $s_i = \pm 1$, so called Ising spins. These Ising spins interact with other spins in lattice, and in a ferromagnet this interaction favors parallel allignment of pairs of spins. In a real magnet, the strength of the interaction decreases rapidly with the increasing separation between two spins. Therefore, the simplest Ising model assumes an interaction only between the nearest neighboring spins so that the energy of the system is given by the following Hamiltonian:

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

where the sum is done over all pairs of nearest neighboring atoms <i, j>, and J is defined as the exchange or coupling constant (measure of strength of interaction between spins). Equation 1 tells us that the two neighboring atoms' spins will have energy of interaction $-J$ if they are parallel, and $+J$ if they are antiparallel. Assume $J > 0$, therefore the interactions favor parallel alignemnt of neighboring spins. Equation (1) completes the description of the Ising model.

The energy of the spin system is the lowest if they are all parallel to each other, but what happens with increasing temperature? We know that for a system that is in equilibrium with its environment, the probability of finding a system in any particular state is given by:

$$P_\alpha = exp(\frac{-E_\alpha}{K_B T}) \tag{2}$$

where E_α is the energy of state α calculated from Equation (1) and P_α is the probability of system being in state α [1]. Particular configuration of the spins is defined as the microstate of the system. Considering a lattice of N Ising spins, each of them having 2 possible states, there are 2^N different possible microstates of the system.

Microscopically, it is interaction of the spin system with the environnment (interaction meaning heat exchange) that causes the system to undergo transitions from one microstate to the other. If an atom gains energy from the environment its spin flips from $+1$ to -1, and conversely if it loses energy to the environent it flips from -1 to $+1$. Macroscopic quantity magnetization, i.e. total magnetic moment, averages over many microstates that the system goes through during the course of measurement. Magnetic moment of a specific microstate M_α is the sum of the values of s_j for all of the spins in that particular state, so that the measured magnetization of the system is given by:

$$M = \sum_\alpha M_\alpha P_\alpha \tag{3}$$

where $M_\alpha = \sum s_j$ with s_j corresponding to the j^{th} spin direction in a microstate α.

We are interested in the systems for which the number of spins $N \to \infty$, so that the number of states is also large. Analytical approaches have yielded only a few exact results, which makes simulations very attractive for this problem [1].

This is a simple spin model. One could also consider spins to be vectors of fixed length which are free to rotate in 3D (Heisenberg model).

1.1 Mean field theory

The magnetization M is closely related to the average spin alignment $< s_i >$ (time average for a system in thermal equilibrium with environment). For an infinitely large system, all spins have the same average alignment, since all spins are equivalent (treated in the same fashion, same restrictions imposed on all), resulting in all of them having same average properties. Therefore, total magnetization of the system of N spins, at temperature T is given by:

$$M = \sum_i < s_i >= N < s_i > \tag{4}$$

This equation is pretty handy because if we can find $< s_i >$ we immediately have magnetization as well. To calculate $< s_i >$ we need something called mean field theory.

If we take into account the magnetic field, out Hamiltonian becomes:

$$H = -J \sum_{<i,j>} s_i s_j - \mu B \sum_i s_i \tag{5}$$

where B is the magnetic field and μ is the magnetic moment associated with each spin. B field will tend to make spins orient parallel to it, such as to lower the total energy of the system.

In our simulation, magnetic field is assumed to be zero.

1.2 Monte Carlo method

We begin with the system in a particular microstate. The interaction with environment is then as follows: a spin is chosen (either at random or systematically) and the energy required to make it flip E_{flip} is calculated, using equation (1). If $E_{flip} < 0$ (energy of the system would be lowered by reversing the spin), the spin is flipped and the system moves into a different microstate. If $E_{flip} > 0$ (energy of the system would be increased), a decision must be made. A random number (uniformly distributed between 0 and 1) is generated and compared to the Boltzman factor $BF = exp(\frac{-E_{flip}}{k_B T})$. If $BF > randnum$ the spin is flipped, else the spin is left undisturbed. So, whether the state moves to a different microstate depends on the random number generated. Then, another spin is chosen, and the steps are repeated. This is known as the Metropolis Algorithm. This is repeated a large number of times so that every step is given several chances to flip or not. We can interpret every Monte Carlo time step as one interaction with environment whose effect depends on Temperature since temperature influences the probability of sign-flipping.

2 Translating this into code

Task 1:
If we consider a 2D lattice with periodic boundary conditions, total number of edges is equal to $2N$, N being the number of sites. The ground state corresponds to all sites having the same spin (either all up or all down). Thus, the ground state energy for every edge is $-J$, and consequently, the total energy of the system is $-2NJ$. Magnetization per site is going to be -1 if all spins are down, and $+1$ if all spins are up. Since we take the absolute value of the sum in our case, the magnetization per site is going to be $+1$ regardless of the lattice size.
In the limiting case, i.e. $T \to \infty$, energy of the system tends to 0 (which can be seen by considering the limiting value of the probability weights as $\lim_{T\to\infty} = \exp\left(-\frac{H}{k_B T}\right)$), whereas magnetization per site is going to tend to 0.

Task 2:
First, a function which generates a state or a spin configuration, assigning every site to be either $+1$ or -1 at random, was written. Configuration of size N was generated, where $N = L \times L$, and L is length of a lattice row or column. For convenience, dimensionless units where were used for temperature and exchange constant and they were defined as $k_B = J = 1$.
In Figure (1), visualization of a generated state can be observed:

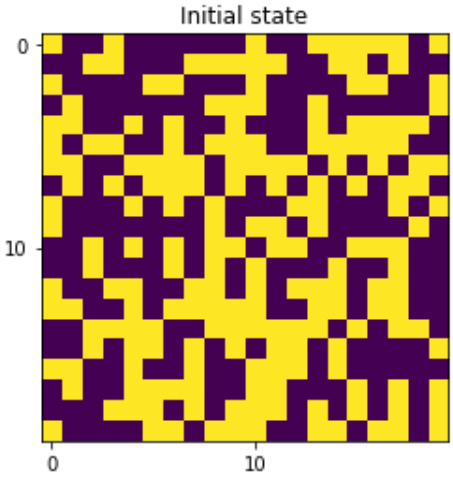

Figure 1: A state with size N=20x20, with spins assigned at random

Task 3:
Then, temperature parameter was set to be $T' = 0.1$. Initial lattice was generated using a function defined in Task 2. A function which computes the total system energy of an N-site lattice was defined according to the following equation:

$$E' = -\sum_{\langle i,j \rangle} s_i s_j \tag{6}$$

which is the same as Hamiltonian seen in Equation (1) if $J = 1$. This equation assumes that there is no applied magnetic field to the system.
In the code, several conditions were applied to account for the boundary of the lattice, since the sites in the corners would only have 2 neighbors, and the sites between the corners on the boundary would have 3 neighbors.

Task 4:
Here, a function which flips a randomly selected spin was implemented. This function flips a sign of the site's spin based on the calculated energy difference between two sites. The function was named $monte\_carlo()$ because this procedure defines one Monte Carlo step, as explained in the Introduction.
My code was slightly different than the algorithm explained in the Introduction. First, randomly selected site's spin was flipped. The energy of this "new" lattice (with a spin flipped was calculated). Then, energy difference was found between the original system and the one in which a spin was flipped (effectively calculating energy difference between 2 sites, since the rest of the sites remained unchanged). Finally, Boltzmann factor r was

defined as:

$$r \equiv \exp\left(\frac{-\Delta E'}{T'}\right) \tag{7}$$

Based on the energy difference calculated, several conditions were applied. If energy difference is smaller than 0, lattice with a flipped sign, as well as its energy was returned (since the energy would be lowered by reversing the spin). If energy difference is larger than 0, a random number between 0 and 1 was compared with Boltzmann factor r. If r was larger than this random number, again, the lattice with a flipped sign, along with its energy was returned, otherwise, the lattice with unflipped (i.e. original lattice) was returned, along with its energy defined as ($oldE$).

Task 5:
Here, a function which updates does N site updates was defined. Since the sites are picked at random, some sites would not be updated, but others would multiple times. However, on average every site would be considered once. This function was named $metropolis()$ in my script.

Task 6:
In this part of the code, the system update was called n0 amount of times, in order to reach convergence, and obtain. For this the $ising()$ function was defined which updates the system n amount of times. In this function, after each system update, energy of the system was computed, as well as the magnetization per site, and these values were stored in arrays E and M respectively.
Magnetization function was defined according to the following equation:

$$|m| = \left|\frac{1}{N}\sum_{i=1}^{N} s_i\right| \tag{8}$$

Then, to obtain a representative sample for energies and magnetization, $ising()$ function was called 10 times, which resulted in 10 arrays, each consisting of $n0$ elements. Then, the i^{th} element of each array was averaged, resulting in a single $n0$-element array (E_fin and M_fin for energy and magnetization respectively).
Finally, a time-series of energies and magnetizations were plotted on separate graphs, and the value n_{min} was estimated. This value n_{min} is the value at which energy and magnetization values converge, and consequently the value at which numerical expectation values are not strongly affected by the temporary behavior for n values larger than n_{min}.
The 2 timeseries can be seen in Figure (2) for energy (left) and magnetization (right).
It was found that the value of n_{min} is around 65, for temperature $T' = 0.1$.

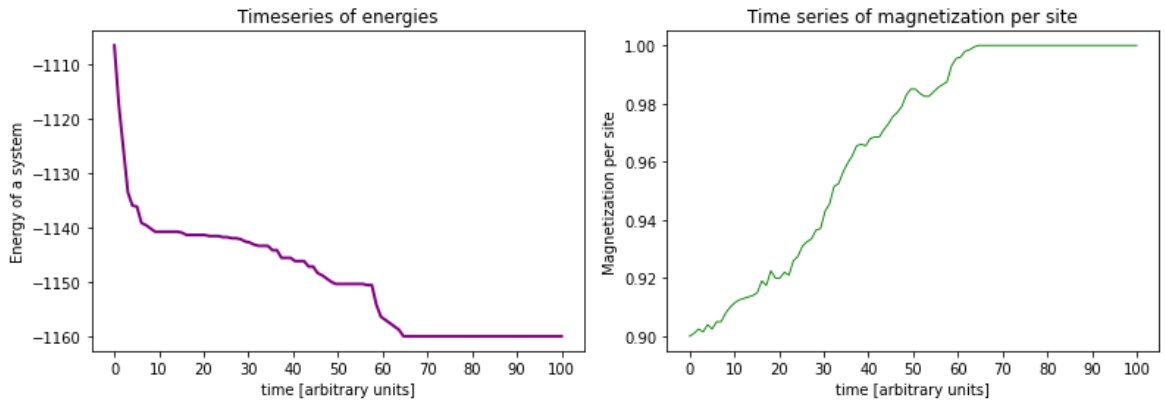

Figure 2: Time-series of energy and magnetization for lattice of size N=20x20 at temperature T=0.1

Task 7:
Value of temperature was modified to be an array of the following values $0.1, 0.5, 1.0, 1.5, 2.0, 2.25, 2.5, 3.0, 10.0$. Numerical expectation values for energy and magnetization at different temperatures were computed. Values of energy and magnetization for a specific temperature value, were averaged over $n = 70 > n_{min}$, and plotted as a function of temperature. These plots can be seen in Figure (3).

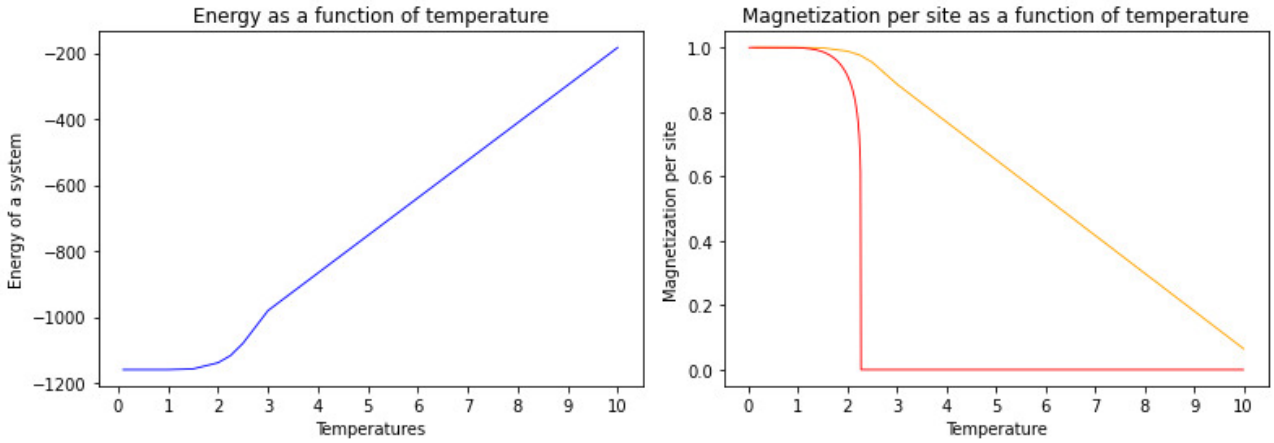

Figure 3: Energy and Magnetization as a function of temperature

Additionally, for each temperature value, a 2D map of the lattice for the final stage at $n = 70$ was plotted, and can be seen in Figure (4).

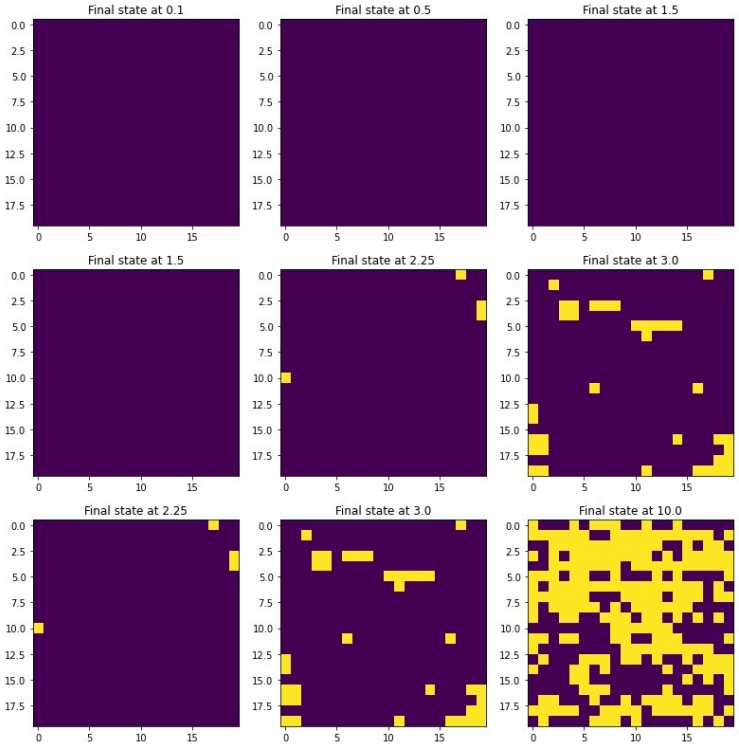

Figure 4: 2D maps of lattices at different temperatures

Task 8: From Figure (3), on the right, one can observe how the numerical results agree with the theoretical ones. Theoretical values for magnetization were calculated using the following formula:

$$|m| = \sqrt[8]{1 - \left(\sinh\left(\frac{2}{T'}\right)^{-4}\right)} \quad 0 < T' < T_{crit} \tag{9}$$

where $T_{crit} = \frac{2}{\ln(1+\sqrt{2})} = 2.2691$ and $|m| = 0$ for temperatures larger than T_{crit}. This is known as Onsager's solution.

In Table (1), one can observe theoretical and numerical values found for different temperatures, as well as the error of numerical simulation.

T	$m_{numerical}$	$m_{theoretical}$	$\Delta_{numerical}$
0.1	1.0000	1.0000	0.000
0.5	1.0000	0.9999998	0.000000225
1.0	1.0000	0.9992758	0.000724248
1.5	0.99714	0.9864996	0.010643254
2.0	0.99007	0.9113193	0.078752051
2.25	0.98200	0.6718540	0.310145973
2.5	0.95492	0.0000	0.954928571
3.0	0.77121	0.0000	0.771214286
10.0	0.05971	0.0000	0.059714288

Table 1: Table showing numerical results from the code, theoretical values, as well as the error of numerical approximation

It can be observed that numerical values best agree with the theoretical for temperatures less than $T = 2.0$ where the relative error of numerical approximation is less than 10%.

Task 9 and 10:
An attempt was made to calculate errors by using the following formula:

$$SE = \frac{\sigma}{\sqrt{x}} \tag{10}$$

where σ is the standard deviation of the data set, and x is the sample size. I attempted to calculate this error for energies and magnetizations, by first extracting all i^{th} elements and storing them in array new_Es, and then computing standard error for each sub-array in new_Es. The compiling took a bit too long so the plots and values were not found for this part of the project, but the code was nonetheless written.

3 Bibliography

[1] Giordano, Nicholas J. “Statistical mechanics, Phase Transitions and the Ising Model.” Computational Physics (2nd Edition)„ edited by Benjamin Cummings, 2005, pp. 235-247