Exercise 03

Submit your files before next Friday at 8.00 am to luca.donati@fu-berlin.de

1 Ising model (25 Points)

1.1 Theory

The Ising model is a physical model used to study the properties of ferromagnetic solid made of localized atoms in a lattice (then they cannot move and the kinetic term can be neglected). The interaction between atoms is represented by the general Hamiltonian:

$$H = -J \sum_{i,j} s_i s_j - B_z \sum_i s_i$$

where:

- i, j are the indices of the atoms of the model
- s_i is the spin of the atom i ($s_i = \pm 1$)
- J is the exchange energy:

  $J > 0 \rightarrow$ ferromagnetic solid (parallalel alignment with B_z)

  $J = 0 \rightarrow$ paramagnetic solid

  $J > 0 \rightarrow$ antiferromagnetic solid (antiparallalel alignment with B_z)
- B_z is an external magnetic field (along the z direction)

1.2 Monte Carlo Metropolis algorithm

Let's consider a square lattice (a matrix) with N^2 sites, where each site (representing the spin of an atom) can interacts with the horizontal and vertical neighbors, according to the periodic boundary conditions (pbc). In the following example two red atoms and their (blue) neighbors are highlighted.

$$\begin{pmatrix} 1 & 1 & -1 & 1 & -1 & -1 & -1 & \textbf{-1} \\ 1 & -1 & 1 & 1 & \textbf{1} & -1 & 1 & 1 \\ -1 & 1 & -1 & \textbf{1} & \textbf{-1} & \textbf{-1} & 1 & -1 \\ 1 & -1 & 1 & -1 & \textbf{1} & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 \\ -1 & -1 & 1 & 1 & 1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 & 1 & 1 & -1 & \textbf{-1} \\ \textbf{-1} & 1 & -1 & 1 & -1 & 1 & \textbf{-1} & \textbf{1} \end{pmatrix}$$

To simulate the time-evolution of the Ising model, we use the Monte Carlo Metropolis algorithm. Given an initial configuration (with the spins randomly distributed like in the previous example), to evolve the system of one timestep, we have to pass all the sites (using a double for loop) and compute the energy of each atom. Then we decide if an atom can flip or not, using the following algorithm:

1. Compute the energy of the site s_i:

   $$E_0 = -J \cdot s_i \cdot (s_u + s_d + s_r + s_l - B_z)$$

   where s_u, s_d, s_r, s_l are the four neighbors.

2. Flip the spin of the site s_i, then the new energy is:

$$E_1 = -E_0$$

3. Compute $\Delta E = E_1 - E_0 = -2E_0$
4. if $\Delta E <= 0$, then the site s_i changes sign: $s_i = -s_i$.
5. if $\Delta E > 0$:
   (a) We extract a random number $r \in [0, 1)$ from a uniform distribution.
   (b) If $\exp\left(-\frac{\Delta E}{k_b \cdot T}\right) > r$, then the site s_i changes sign. Otherwise s_i does not change.

The algorithm (repeated for many timesteps) brings the system to a thermodynamic equilibrium. The number of timesteps necessary depends on the initial parameters.

1.3 Energy and magnetizazion of the system

When the system is in equilibrium, we can study the behaviour of several physical quantities. We will focus on the energy and the magnetization.
The *average energy* of the system is given by the sum of the energy of each atom:

$$\langle E \rangle = -\frac{1}{2}\frac{1}{N^2}\left(J\sum_{i=1}^{N^2}\sum_{j=i_n} s_i s_j - B_z \sum_{i=1}^{N^2} s_i\right)$$

where the factor $\frac{1}{2}$ was added to avoid to double count the pairs. The notation $j = i_n$ means that the atoms j are the four neighbors of the atom i.
The *average magnetization* is defined as:

$$\langle M \rangle = \frac{1}{N^2}\sum_{i=1}^{N^2} s_i$$

The Ising model, when $B_z = 0$ (i.e. there is not an external magnetic field, is characterized by a critical temperature and a transition phase.
If $T << T_C$, at the equilibrium, the spins tend to align, this means that:

$$\lim_{t\to+\infty} \langle E \rangle = -2$$

$$\lim_{t\to+\infty} |\langle M \rangle| = 1$$

If $T >> T_C$, at the equilibrium, the spins are disordered, this means that:

$$\lim_{t\to+\infty} \langle E \rangle = 0$$

$$\lim_{t\to+\infty} \langle M \rangle = 0$$

where t is the time of the simulation. The critical temperature can be found analytically and its value is:

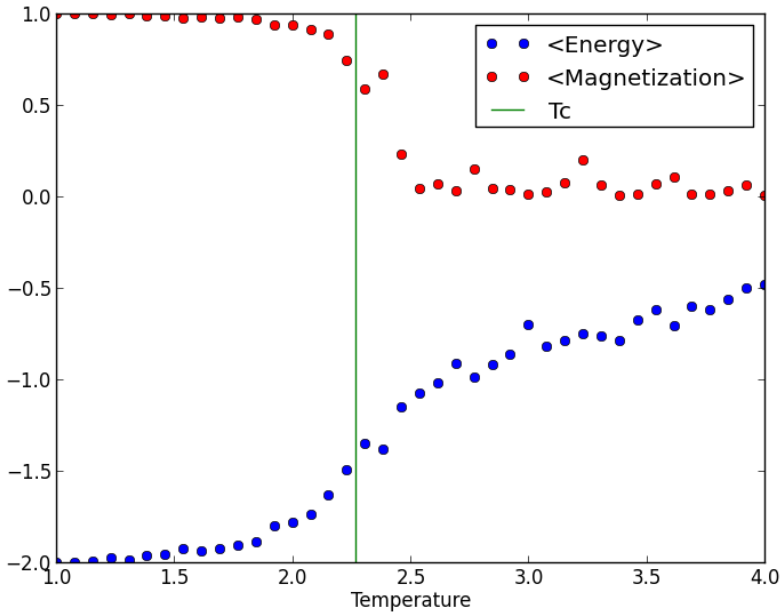
$$T_C = \frac{2J}{k_b \cdot \log(1+\sqrt{2})}$$

Your task, for next week, is to implement the Ising model with Python, to compute $\langle E \rangle$ and $\langle M \rangle$ for different values of the temperature and to check that there is a phase transition at the critical temperature.

1.4 Python implementation

```
import numpy as np
import matplotlib.pyplot as plt

numSim = 40                         #number of simulations
numTimesteps = 80                   #number of timesteps for each simulation
                                    #(each temperature)

N = 30                              #side of the grid
kboltz = 1.
Bz = 0.                             #magnetic field in direction z
J = 1.                              #exchange energy
Tc = ...                            #critical temperature

temperature = ...                   #create an array with 'numSim' temperatures,
                                    #from 1 to 4

Eavg = ...                          #These are two arrays, used to store
Eavg = ...                          #the value of the average energy and the average
                                    #magnetization for each simulation

S = ...                             #matrix NxN with the atoms configuration.
                                    #The elements are randomly generated
                                    #and can assume the value +1 or -1

for n in np.arange(numSim):         #1st loop: update the temperature at each simulation
    temp = temperature[n]           #select the temperature for the simulation 'n'

    for t in np.arange(numTimesteps): #2nd loop: update the timestep of a simulation
        E = 0                       #initialize the energy for each timestep

        for i in np.arange(N):  #3rd loop for the rows
            for j in np.arange(N): #4th loop for the columns

                #Here you should select the neighbors of the site [i,j]
                #Be careful to the PBC
                        #       j-->
                        #    i          up
                        #    |  left [i,j] right
                        #    v         down

                E_ij = ...  #energy of the site [i,j]
                delta_E =  ... #difference of energy after the flip

                #Implement here the metropolis algorithm
                             ...

                E = E + E_ij #sum the energy of each atom
                             # after the metropolis algorithm

    #At the end of each simulation, compute <E> and <M>:
    Eavg[n] = ...
    Mavg[n] = ...


#Plot  <E> and <M>
```

The execution of the program can take several minutes. Using the proposal parameters, you should get a graph similar to the following in few minutes. For a better result, you should increase the number of atoms and of timesteps.



2 Self-adjoint operators and commutators (20 points)

Prove that:

1. If $\hat{A}$ and $\hat{B}$ are self-adjoint, then $\hat{C} = \hat{A} + \hat{B}$ is self-adjoint.
2. If $\hat{A}$ and $\hat{B}$ are self-adjoint, then $\hat{C} = \hat{A}\hat{B}$ is only self-adjoint, if $\hat{A}$ and $\hat{B}$ commute $\left(\left[\hat{A}, \hat{B}\right] = 0\right)$.
3. If $\hat{A}$ is self-adjoint, then $\hat{A}^n$ is self-adjoint.
4. If $\hat{A}$ is self-adjoint, then $\exp(\hat{A})$ is self-adjoint.
5. If $f(x)$ is an analytic function, $\hat{A}$ and $\hat{B}$ are self-adjoint, then $[\hat{A}, f(\hat{B})] = f'(\hat{B})[\hat{A}, \hat{B}]$

NB: A function $f(x)$ is said to be analytic around a point x_0 if $f(x)$ can be expanded in a power series around x_0. If $f(x)$ is analytic, then its Taylor series converges to $f(x)$ for x in a neighborhood of x_0.