Exercise sheet 3

The one-dimensional Ising model
PUE Advanced Computational Physics
University of Vienna - Faculty of Physics

The Ising model

In its simplest form, the Ising model is defined by the energy of state $\nu = \{s_1, \ldots, s_N\}$,

$$E_\nu = -J \sum_{i,j}{}' s_i s_j - H \sum_i s_i, \tag{1}$$

where $J > 0$ is the coupling constant, H is the external field, $\sum'_{i,j}$ indicates a sum over all nearest neighbor pairs, and the s_i are the spins that can take on values of $+1$ and -1. Note that for simplicity, we have set $\mu = 1$ for the magnetic moment of a spin.

8 Implementing an MC simulation of the 1d Ising model

In the following you will create a Metropolis Monte Carlo simulation of the one-dimensional Ising model. The file `mc_ising_1d_incomplete.zip` contains the scaffolding for this simulation, which you will complete in the following steps. Alternatively, feel free to implement everything from scratch in your language of choice.

In the Python example, the simulation is implemented in the class `IsingModelMC_1D`. In the `__init__` method you can see the variables that are used during the simulation (remember: the *self* variable is used to access members of the instance itself):

- **N** - number of spins in the system
- **pbc** - True if periodic boundary conditions are used
- **h_over_j** - the ratio H/J
- **T** - the temperature
- **beta** - the inverse temperature $\beta = 1/k_\mathrm{B}T$ with $k_\mathrm{B} = 1$
- **config** - the configuration as an integer array of size N

We will start building the necessary routines from the bottom up. Finish the implementation of the following methods:

a. **get_neighbors(self,i)** - return the neighbors of spin i with or without periodic boundary conditions depending on the value of **self.pbc**.

b. **get_E(self)**, **get_dE_single(self,i)** - these methods calculate the total potential energy and the change in total energy if spin i is flipped, respectively. Make sure that these two methods are consistent with each other before you proceed.

c. **acceptance(self,dE)** - implement the Metropolis acceptance criterion based on the difference in potential energy, **dE**, and the temperature **T**.

d. **do_move(self)** - implement a single Metropolis Monte-Carlo move using the methods you implemented above. Which steps are necessary? If you want to accept a move, use the **accept_move** method, otherwise just leave the function.

Section 3 of the notebook contains code that initializes a simulation instance. You can use the **system** variable to test various parts of the code by calling methods directly. It is highly recommended that you test each method you implement separately before moving on to the next task.

e. Test your implementation by running a simulation. With the parameters given, you can expect an acceptance ratio of roughly 25%.

9 Calculating the magnetization

The partition function of the 1d Ising model with N spins is given by

$$Z_N = \lambda_+^N + \lambda_-^N = \lambda_+^N \left[1 + \left(\frac{\lambda_-}{\lambda_+} \right)^N \right] \tag{2}$$

with

$$\lambda_\pm = e^{\beta J} \cosh(\beta H) \pm \sqrt{e^{2\beta J} \sinh^2(\beta H) + e^{-2\beta J}}. \tag{3}$$

a. Show that in the limit $N \to \infty$ the magnetization per spin $m = \sum_i s_i / N$ is given by

$$m = \frac{e^{\beta J} \sinh(\beta H)}{\sqrt{e^{2\beta J} \sinh^2(\beta H) + e^{-2\beta J}}}. \tag{4}$$

b. Use this analytical result to check your simulation. Discuss your findings.

10 Correlation functions

In the following we will calculate correlation functions both in time and space. These functions are important tools in statistical mechanics and, hence, are often calculated in molecular simulations.

a. Calculate the normalized time autocorrelation function

$$C_m(k) = \frac{\langle m(t)\, m(t+k)\rangle - \langle m(t)\rangle \langle m(t+k)\rangle}{\langle m^2\rangle - \langle m\rangle^2}$$

at zero field for a system size of your choice. Use the result to provide error bars for the magnetization at $H = 0$. *Hint:* make use of the fact that the simulation is stationary in time and hence you can average over all possible time origins.

b. In the $N \to \infty$ limit the spin-spin correlation function of the 1d Ising model can be calculated analytically:

$$\langle s_i s_j \rangle = \cos^2[2\phi] + \sin^2[2\phi] \left(\frac{\lambda_-}{\lambda_+}\right)^{j-i}. \tag{5}$$

Here ϕ is defined by

$$\cot(2\phi) = e^{2\beta J} \sinh(\beta H), \tag{6}$$

and $\lambda_\pm$ are the eigenvalues from the previous example, Eq. (3).

Calculate the site-site correlation function $\langle s_i s_j \rangle$ by averaging over the configurations encountered during a simulation. Compare the result to the analytic solution.