ELTE Faculty of Science

Magnets via the metropolis algorithm

The Ising model

Alex Olar

2018

I. Motivation

Ferromagnets contain finite size domains in which the spins of all atoms point in the same direction. When an external magnetic field is applied it is the nature of the material that the domains start to shift and align. The material becomes magnetic. However, the process is temperature dependent, therefore when the temperature gets high enough it looses its magnetism. The point where this happens is called the Curie point and is material dependent. This phase transition is an interesting domain to examine.

II. The Ising model

Assuming we have a chain of spins where only the nearest neighbors interact with each other then each spin is in the potential:

$$V_i = -J\mathbf{s}_i\mathbf{s}_{i+1} - g\mu_B\mathbf{s}_i\mathbf{B}$$

Where $\mathbf{B}$ is the externally applied magnetic field and μ_B is the well known *Bohr-magneton*. J is the so called exchange energy between spins.

With the simulations I want to find out the accuracy of the derived parameters depending on the number of particles. For a system in state Ψ_α the energy of the system is expected to be:

$$E_{\Psi_\alpha} = \left\langle \Psi_\alpha \middle| \sum_i V_i \middle| \Psi_\alpha \right\rangle = -J\sum_{i=1}^{N-1} s_i s_{i+1} - B\mu_b \sum_{i=1}^{N} s_i$$

The best method to simulate a magnetic system is the metropolis algorithm. In the next section I'll talk about that:

III. The metropolis algorithm

The steps of the algorithm are the following:

1. Initial spin configuration of the system.
2. Generating a trial configuration by randomly picking a particle and flipping its spin than calculating its energy.
3. If $E_{\Psi_\alpha} \geq E_{\Psi_{trial}}$ then accept the state otherwise accept with relative probability: $P_{rel} = e^{-\frac{\Delta E}{k_B T}}$

4. If $P_{rel} > random_number$ then accept, otherwise accept the same state as the forward state (where *random number* is in the range [0, 1]).

IV. What I am going to do?

I am going to implement a 1D and a 2D Ising-model with arbitrary, hot and cold starting configurations and make great visualizations for it by:

- writing the metropolis algorithm
- calculating the statistical properties of the system, such as magnetization, specific heat and compare them with analytic results
- extend the model to second nearest neighbors as well

My goal is to make a tool (Python notebook or enough time provided a web application) that is interactive and shows visualizations of the processes.

The obstacles are the hard debugging of python code and acquiring the best random generator and finding the best visualization tool out there.