Physics 344 Assignment 5: The Ising Model

Submission Details

- **To submit online:** Your C code, together with any Python notebooks used for data processing. Also, a **separate** report presenting and discussing your results.
- **Report details:** You can use Latex or a Jupyter notebook. Your report should provide a coherent discussion and analysis of your results. Be clear about how you obtained your data, and about the parameters you used in your simulations. If course, you should compare with known results wherever possible.

Simulation Details

You need to implement the Metropolis algorithm for the two-dimensional Ising model at a finite temperature. The spins are arranged on an $L \times L$ square lattice. At each time step, select a particular spin, say s_i, and propose to flip its sign, i.e. to change s_i to $-s_i$. As discussed in class, the acceptance probability for this transition is determined by the ratio of the Boltzmann factors of the current state $\mathbf{s}$ and the state $\mathbf{s}'$ that we are proposing a transition to. In the Metropolis algorithm this probability is given by

$$A_{\mathbf{s}' \leftarrow \mathbf{s}} = \min\left\{1, e^{-\beta \Delta H}\right\} \tag{1}$$

where

$$\Delta H = H(\mathbf{s}') - H(\mathbf{s}) = 2 s_i \left[J \sum_{j \in N_i} s_j + h \right], \tag{2}$$

with N_i the set of $z = 2d$ nearest neighbours of s_i. *Check that you agree with these expressions!*

To eliminate edge effects, implement **periodic boundary conditions**, i.e. have the interactions "wrap around" the edges of the lattice. This ensures that the model is translation invariant, i.e. all the spins are on exactly the same footing. This helps to reduce finite-size effects.

You can select spins for updating either at random or sequentially. The latter approach is faster, since it eliminates the need to generate random numbers for selecting a spin. While sequential updating technically breaks ergodicity, it works fine in practice. One Monte Carlo **sweep** corresponds to L^2 random choices of spins to flip, or a sequential run through the entire lattice.

While one sweep consists of L^2 elementary Monte Carlo steps, there is no real benefit to recording data about observables any more frequently than after each full sweep.

Your C-based simulation should take at least the two dimensionless parameters βJ and βh as input. You can add more parameters as you see fit.

You will be calculating the exponential $e^{-\beta \Delta H}$ in the acceptance probability billions of times during your simulations. This obviously needs to be done as fast as possible. Note that ΔH, and therefore $e^{-\beta \Delta H}$, can only take a finite number of different values. Calculate these beforehand and store them in a lookup table.

Your code should be able to produce estimates for the expectation values $\langle M \rangle$, $\langle |M| \rangle$, $\langle M^2 \rangle$, $\langle E \rangle / J$ and $\langle E^2 \rangle / J^2$. It should also be able to output the values of M and $|M|$ recorded after each sweep. *Not all of this functionality will be necessary for the first parts of the assignment, but you can structure your code with this in mind.*

Assignment - Part I

The first goal is to implement the simulation described above, and to investigate issues surrounding its equilibration and ergodicity. Once the simulation is running, proceed as follows:

1. Consider $L = 4, 10, 50$ (and any other values you want) and values of $\beta J = J/kT$ between about 0.2 and 0.6. Set $h = 0$ for now. Initialise the simulation in the all-up state.

2. Record, for each case you consider, M/L^2 and $|M|/L^2$ after each sweep. Produce plots that show how these values change over the course of the simulation. On the same plot, include a running average of M/L^2 and $|M|/L^2$. (Look up **np.cumsum**)

3. **Discuss** your results in light of what we discussed in class, and the physics of the Ising model as set out in the notes. Is there evidence of an equilibration period? Can you estimate how long this period is? What can you say about the ergodicity of the simulation, i.e. its ability to explore all the relevant parts of the system's state space? Does it look like the time averages are converging? How long does this take? How does the system size affect things? What role does the temperature play? Is there a clear change in behaviour at a specific temperature? *Do not go overboard with the data generation or analysis here! The point of this section is just to get a feeling for what the simulation is doing.*

Assignment - Part II

For $L = 200$, or larger, investigate the average magnetisation per spin $m = \langle M \rangle / L^2$ and average absolute magnetisation per spin $m_{\text{abs}} = \langle |M| \rangle / L^2$ as functions of $\beta J = J/kT$, still for $h = 0$. *Be sure to give your simulation enough time to equilibrate before you start gathering data.* **Discuss** your results.

Note: *Close to the phase transition you will encounter a phenomenon called* ***critical slowing down****. This coincides with the formation of large clusters of aligned spins, which are very slow to destroy via single spin flips. This drastically reduces the simulation's efficiency at exploring the state space, resulting in very large autocorrelation times. As you saw in the previous assignment, this leads to a large statistical error. You will probably need to run the simulation for a long time in order to get reliable results in this region.*

Assignment - Part III

The system's specific heat and magnetic susceptibility can be expressed as

$$\frac{c}{k} = \beta^2 \frac{\langle E^2 \rangle - \langle E \rangle^2}{N} \tag{3}$$

and

$$\chi = \beta \frac{\langle M^2 \rangle - \langle M \rangle^2}{N}. \tag{4}$$

Investigate, for $h = 0$ and an $L \geq 200$, the behaviour of c/k and $J\chi$ as functions of

$$t = \frac{T - T_c}{T_c} = \frac{J}{kT_c} \frac{kT}{J} - 1. \tag{5}$$

You will probably need to use a higher density of plot points close to the phase transition. Investigate the power-law behaviour of $J\chi$ at $t = 0$. Depending on the quality of your data, you can either try to extract the critical exponent γ or, more simply, try to verify the claims made in the notes. Note that the proportionality constant $C_\pm$ in

$$\chi(t, 0) \sim C_\pm |t|^{-\gamma} \quad \text{as} \quad t \to 0^\pm \tag{6}$$

takes different values on the two sides of the phase transition.

Assignment - Part IV

Background:

Consider the **two-point function**

$$G(\mathbf{r}_i, \mathbf{r}_j) = \langle s_i s_j \rangle, \tag{7}$$

where $\mathbf{r}_i$ and $\mathbf{r}_j$ are the positions of spins s_i and s_j on the lattice. This is an interesting and important quantity since it contains information about the correlations between spins. The periodic boundary conditions ensure translational invariance, and so G depends only on the position of the one spin relative to the other, i.e. $G(\mathbf{r}_i, \mathbf{r}_j) = G(\mathbf{r}_i - \mathbf{r}_j)$. Close to the critical point the spins will be correlated over large distances. In this region the angular dependence of G is fairly weak, and we can regard it as a function of the distance $r = |\mathbf{r}_i - \mathbf{r}_j|$ only. This allows us to write

$$G(r) = \langle s_i s_j \rangle, \tag{8}$$

with the understanding that s_i and s_j are a distance r apart. Of course, when calculating r we need to keep the periodic boundary conditions in mind.

When $J = 0$ the Ising model reduces to a paramagnet with no interactions between the spins. In this case the spins are uncorrelated, and so $\langle s_i s_j \rangle = \langle s_i \rangle \langle s_j \rangle$ since s_i and s_j are independent random variables with respect to the Boltzmann distribution. When $J \neq 0$ the spin-spin interactions introduce correlations and so $\langle s_i s_j \rangle \neq \langle s_i \rangle \langle s_j \rangle$. However, at large distances the spins still become uncorrelated, and $G(r)$ will tend to $\langle s_i \rangle \langle s_j \rangle$ with as $r \to \infty$. Note that, as a result of translation invariance, $\langle s_i \rangle = \langle s_j \rangle = m$. The rate at which $G(r)$ approaches $\langle s_i \rangle \langle s_j \rangle = m^2$ therefore quantifies how the correlations between spins decay with distance. With this in mind, we introduce the **spin-spin correlation function**

$$g(r) = G(r) - m^2 = \langle s_i s_j \rangle - \langle s_i \rangle \langle s_j \rangle = \langle (s_i - \langle s_i \rangle)(s_j - \langle s_j \rangle) \rangle, \tag{9}$$

which satisfies $\lim_{r \to \infty} g(r) = 0$. In fact, away from the critical point, and at large distances, $g(r)$ decays roughly exponentially as

$$g(r) \sim e^{-r/\xi}, \tag{10}$$

where ξ is the **correlation length**. The latter is the characteristic length scale of the system, since it quantifies the typical range over which spins are correlated with each other.

Question:

For $h = 0$ and an $L \geq 200$, use your simulation to calculate $G(r)$ for different values of r and βJ. Also calculate m for each βJ in parallel. To speed things up, you should consider multiple pairs of spins at the same distance r for each state in your dataset, and then average the products of each pair's spin states. How you choose these pairs is up to you. For example, you could use spins from the same row, column, or diagonal, or you could include other spin pairs as well. Your choice here will also determine the values of r you consider. *You should also think about the efficiency of your algorithm in terms of the number of states you generate, and the number of spin pairs you use from each one.*

Produce plots of $G(r)$ and $g(r)$ versus r for different temperatures. Using appropriate fits, extract the correlation length from your data, and plot this as a function of βJ or $t = (T - T_c)/T_c$. **Discuss** your results, and the physical interpretation thereof.