Computational Assignment 6: The Correlation Length

Philip Cherian

November 16, 2023

In this assignment's computational exercise you will be extending your code for the 2D Ising Model, so all you need to do modify your earlier code file to include some new functions. You will be computing a quantity known as the *correlation length* ξ. In order to compute ξ, you must first compute something called the spin-spin correlation function, which is essentially a measure of how much correlation we can expect between two spins that are a given distance r apart. To motivate this function, consider what happens to the quantity $\langle \sigma_{i,j} \cdot \sigma_{p,q} \rangle$, where $\sigma_{i,j}$ and $\sigma_{p,q}$ are spins at different positions on a lattice. If these spins were statistically independent, then we could say that

$$\langle \sigma_{i,j} \cdot \sigma_{p,q} \rangle = \langle \sigma_{i,j} \rangle \cdot \langle \sigma_{p,q} \rangle = m^2, \tag{1}$$

since the mean of the product of two independent random variables is the product of their means. In the last step, we have further used the fact that since the Ising model is translationally invariant, the average value of a spin at (i, j) is no different from the average value of a spin at (p, q), and therefore both of these quantities can be replaced by the mean magnetisation per spin, $m = \langle \sigma \rangle$.

Now, consider the quantity $\langle \sigma_{i,j} \cdot \sigma_{p,q} \rangle - m^2$. If the spins were independent, this would be zero. This quantity thus corresponds to the "correlation" of the spins.[1] We are interested in how far a spin's influence reaches. As a result, one would like to find the correlation as a function not of *spin*, but of *distances* between spins. It should be clear that on a square lattice we can define a distance between two spins through

$$r^2 = (p - i)^2 + (q - j)^2. \tag{2}$$

In other words, we can define the correlation as a function of distance r using

$$c(r) = \langle \sigma_{i,j} \cdot \sigma_{p,q} \rangle - m^2. \tag{3}$$

Note that the left-hand side has no mention of the indices (i, j) and (p, q): it is a quantity independent of spins, but which only depends on the distance between them. In other words, *all pairs of spins* separated by a distance r would contribute to the value of $c(r)$. In practice, even though r on a lattice can only take certain discrete values, it is a good idea to decide on a range of equally-spaced "bins" at which we will be computing $c(r)$. Let us say that you have a fixed configuration. A simple algorithm to compute $c(r)$ is as follows:

- Define `r`, an array of "bins" of possible values of r. The smallest value of r is 1, and the largest value is $L/\sqrt{2}$. (Why? *Hint:* Remember the boundary conditions!)

[1] Essentially, this correlation is a measure of "peer pressure": How is the spin at site (p, q) influenced by the state of the spin at site (i, j)? For example, if there is a large magnetic field, and if (p, q) and (i, j) are far apart, then both spins will tend to point in the direction of the field, but this is not because of peer pressure, it is because of "external" influence due to the field. Subtracting m^2 makes sure that we take care of such "external" influences. This nice description was taken from Chapter 9.4: Correlation Functions in the Ising Model of *Statistical Mechanics* by Daniel Styer.

- Define two empty arrays `nr` and `Cr` of the same length as `r` to store (i) the number of spins and (ii) the "correlation" values between r and $r + \mathrm{d}r$.
- Loop over all the spins in your lattice. For each spin, loop over all other spins in the lattice and compute
  - The correlation, i.e. the value of $\sigma_{i,j} \cdot \sigma_{p,q} - m^2$, and
  - The distance between the spins r.
- Find which distance-bin `rbin` this distance corresponds to and increment the number of spins `nr[rbin]`, and add the correlation to `Cr[rbin]`.
- Once you have completed this for the entire lattice (i.e., all possible pairs), divide `Cr` by `nr` to find the correlation function for this fixed configuration.

(a) Write a new function `get_correlation` which accepts a snapshot of the lattice and computes the correlation function. (*Hint:* You can use the `pbc_distance` function you implemented in Assignment 3 to compute the distances between spins.)

If you're having trouble computing the distances between all pairs, you could restrict yourself to pairs along a single row. This should be simpler to compute, although I haven't actually done it myself. [4]

(b) Choose $T = 1.01\,T_c$. Start with an initial random configuration of spins. Allow the system to "warm up" by running around 10,000 Monte-Carlo sweeps. Save the final configuration. Display this final configuration, and the correlation function associated with it. Repeat this for 3 different temperatures above T_c, and show all three lattices and their associated correlation functions. [3]

(c) Now, assume that $c(r) \sim e^{-r/\xi}$ for r sufficiently large. The quantity ξ is known as the *correlation length*. Compute the correlation length for a range of temperatures $T > T_c$. How does the correlation length fall as a function of temperature? Can you draw any conclusions about what $\xi(T = T_c)$ could be? [3]