Willoughby Seago

Theoretical Physics

Statistical Physics

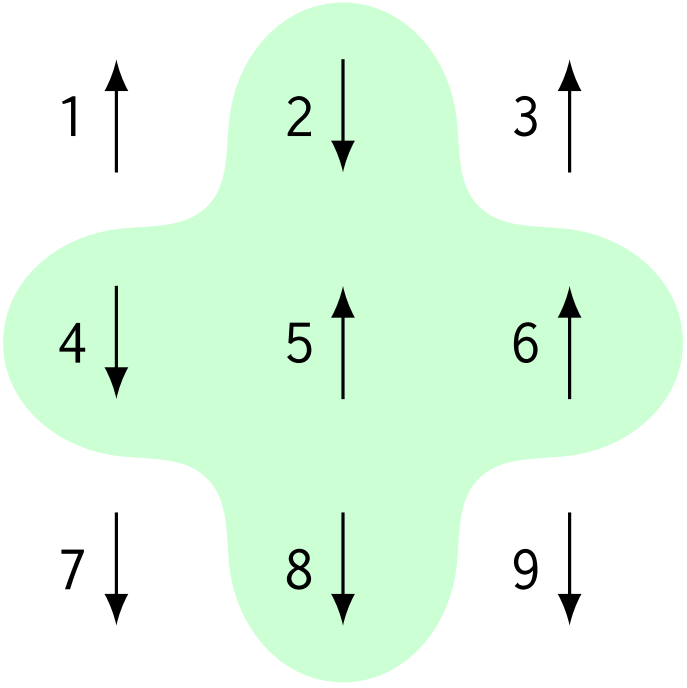
Course Notes

Statistical Physics

Willoughby Seago

These are my notes from the course statistical physics. I took this course as a part of the theoretical physics degree at the University of Edinburgh.

These notes were last updated at 13:45 on January 30, 2022. For notes on other topics see https://github.com/WilloughbySeago/Uni-Notes.



Chapters

Page

Contents

Part I

Monte Carlo and the Ising Model

One

Monte Carlo

Monte Carlo algorithms, named after the Monte Carlo casino, are stochastic methods for computing integrals. In a Monte Carlo computation we approximate the value of an integral by a series of averages.

1.1 Averages

Given a random variable, x_i, the average is defined as

$$\langle x \rangle := \frac{1}{n} \sum_{i=1}^{n} x_i \tag{1.1.1}$$

where the sum is over all measurements of x. Similarly if f is a function of the random variable x_i then the average value of this function is

$$\langle f(x) \rangle := \frac{1}{n} \sum_{i=1}^{n} x_i f(x_i) = \sum_{i=1}^{n} x_i f_i \tag{1.1.2}$$

where we define $f_i = f(x_i)$.

Often we instead want to work with weighted averages, say if certain outcomes are more likely than others. If outcome i has weight w_i then the weighted average of the random variable x_i is

$$\langle x \rangle := \frac{\sum_{i=1}^{n} x_i w_i}{\sum_{i=1}^{n} w_i}. \tag{1.1.3}$$

The weighted average of f is similarly

$$\langle f(x) \rangle := \frac{\sum_{i=1}^{n} f(x_i) w_i}{\sum_{i=1}^{n} w_i}. \tag{1.1.4}$$

1.2 Monte Carlo Integration

Suppose we wish to compute the d-dimensional integral

$$I = \int_V f(\boldsymbol{r}) \, \mathrm{d}^d r. \tag{1.2.1}$$

First note that we can write this as an integral over all space by writing it as

$$I = \int_{\mathbb{R}^d} f(\boldsymbol{r}) \mathbf{1}_V(\boldsymbol{r}) \, \mathrm{d}^d r \tag{1.2.2}$$

where $\mathbf{1}_V$ is the indicator function for V, defined by $\mathbf{1}_V(\boldsymbol{r}) = 1$ if $\boldsymbol{r} \in V$ and $\mathbf{1}_V(\boldsymbol{r}) = 0$ if $\boldsymbol{r} \notin V$.

Using the fact that

$$V = \int_V \mathrm{d}^d r \tag{1.2.3}$$

is the volume of the integration region we can write the integral as

$$I = \frac{\int_V f(\boldsymbol{r}) \mathrm{d}^d r}{\int_V \mathrm{d}^d r} V. \tag{1.2.4}$$

Notice now the similarity to the weighted average with all weights being 1, which is just the normal average, simply change integrals to sums. We therefore have

$$I = \frac{\int_V f(\boldsymbol{r})\, \mathrm{d}^d r}{\int_V \mathrm{d}^d r} V \approx \frac{\sum_i f(\boldsymbol{r_i})}{\sum_i 1} = \langle f(\boldsymbol{r}) \rangle. \tag{1.2.5}$$

The process of calculating an integral becomes the process of computing an average.

If we take n samples then the sum in the denominator is simply n and we have

$$I \approx \frac{\sum_{i=1}^n f_i}{n} V. \tag{1.2.6}$$

This is our Monte Carlo estimate of the integral.

The error on this measurement is associated with the standard error on the mean, since we are calculating a sample mean but really we want the population mean, that is $n \to \infty$. The standard error on the mean is

$$\frac{\sigma}{\sqrt{n}} \sim n^{-1/2} \tag{1.2.7}$$

where σ is the standard deviation. We clearly want n to be large in order to have a small error.

1.2.1 Trapezium Rule

An alternative method for approximating integrals is to split the integration region into sections, we then approximate f as linear over these sections. Define $\overline{f_i}$ to be the average value of this linear approximation of f in the ith section. The integral is then approximated as the total volume of these sections that we have split the integral into. That is

$$I = \int_V f(\boldsymbol{r})\, \mathrm{d}^d r \approx \sum_i \text{volume of } i\text{th trapezium} = \sum_i \delta^d \overline{f_i} \tag{1.2.8}$$

where δ^d is the area of the base of the section, assuming we split the space evenly into a grid spaced δ apart, and $\overline{f_i}$ is the average height of the linear approximation of the function in this region. For a one-dimensional integral these small sections are trapeziums, hence the name trapezium rule.

The error in our approximation of the function is $O(\delta^2)$ since we have approximated f as linear. The error then scales as $n\delta^d\delta^2$ where n is the number of regions.

The number of regions is the integration volume divided by the volume of each region, $n = V/\delta^d$, and so $\delta^d \sim 1/n$, which means $\delta \sim n^{-1/d}$. The error therefore scales as $nn^{-1}n^{-2/d} = n^{-2/d}$.

Comparing this to the error for the Monte Carlo method we see that if $d > 4$ the error on the Monte Carlo method actually decreases faster than the error for the more complex trapezium rule.

1.3 Statistical Physics

In statistical physics the Monte Carlo method can be used to compute thermodynamic averages. A system with a fixed number of particles, N, is a canonical ensemble and the physics is dictated in part by the partition function

$$Z \coloneqq \sum_{i \in \{\text{states}\}} \mathrm{e}^{-\beta E_i} \tag{1.3.1}$$

where $\beta \coloneqq 1/(k_\mathrm{B}T)$ with E_i being the energy of state i, k_B being Boltzmann's constant and T the temperature.

The average of some observable, A, is then given by summing over all states, i:

$$\langle A \rangle \coloneqq \frac{\sum_i A_i \mathrm{e}^{-\beta E_i}}{\sum_i \mathrm{e}^{-\beta E_i}} = \frac{1}{Z} \sum_i A_i \mathrm{e}^{-\beta E_i}. \tag{1.3.2}$$

That is $\langle A \rangle$ is the weighted average with weight $w_i = \mathrm{e}^{-\beta E_i}$.

Writing this same value as

$$\langle A \rangle \approx \frac{\sum_{i=1}^n A_i \mathrm{e}^{-\beta E_i}}{n} \frac{n}{\sum_{i=1}^n \mathrm{e}^{-\beta E_i}}, \tag{1.3.3}$$

here we've introduced a factor of $n/n = 1$, as well as approximating the sums over all states with sums over n states. Notice that the first term is the Monte Carlo estimate of the average of A and the second is the reciprocal of the Monte Carlo estimation of the partition function.

The question is how we sample the states. The obvious answer of uniformly sampling the states turns out not to work, as we will see in the next section.

1.4 The Ising Model

The Ising model is a simple model that can be used to demonstrate several concepts in statistical mechanics. The **Ising model** consists of a lattice of equally spaced points and we imagine that at each point there is a particle which can have either spin up or spin down. We will work with a two-dimensional grid. We index the points from 1 to N, although we could also index each point by its row an column, which is often more useful in code. To each point i we assign a value of $+1$ or -1 depending on if the spin is up or down. We start with the spins chosen randomly and uniformly, so approximately half of the spins will be up and half down.

The energy of the system depends only on how the spins align. We assume the grid spacing is large enough that only interactions with the nearest neighbours have effect. The total energy of the system is then

$$E = -J \sum_{\langle i,j \rangle} S_i S_j \tag{1.4.1}$$

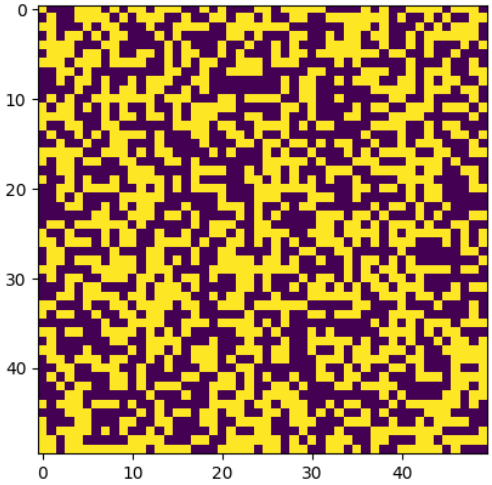


Figure 1.1: Initially the Ising model starts in a random state. Here opposite spins are represented by different colours. The axes are simply the row and column numbers, here $50^2 = 2500$ sites has been used.

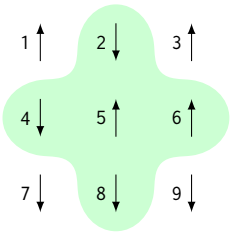


Figure 1.2: A 3×3 Ising model with spins represented by arrows. The shaded region is the nearest neighbours of spin 5, the central spin.

where J is a positive constant with dimensions of energy and $\langle i, j \rangle$ means that we sum over i and then j corresponds to the nearest neighbours.

Figure 1.2 shows a 3×3 Ising model. The nearest neighbours of the 5th point are points 2, 4, 6, and 8. The contribution to the sum when $i = 5$ is

$$-J \sum_{\langle 5,j \rangle} S_5 S_j = -J S_5 (S_2 + S_4 + S_6 + S_8) = -J1(-1 - 1 + 1 - 1) = 2J \tag{1.4.2}$$

Notice that if we flip spin 5, that is $S_5 \longrightarrow -S_5$, the change in energy is

$$\begin{aligned} \Delta E &= -J(-S_5)(S_2 + S_4 + S_6 + S_8) - (-J)S_5(S_2 + S_4 + S_6 + S_8) && (1.4.3)\\ &= 2JS_5(S_2 + S_4 + S_6 + S_8) && (1.4.4)\\ &= 2J1(-1 - 1 + 1 - 1) && (1.4.5)\\ &= -4J && (1.4.6) \end{aligned}$$

This can be used to avoid having to calculate the energy of the entire model when we flip a single spin.

In a general Ising model of N sites, which in our example is 9, there are 2 possible values of the spin at each site and hence there are 2^N possible states. In our example

there are $2^9 = 512$ possible states. A more realistic size would be a 50×50 grid which has $N = 50^2 = 2500$, in which case $2^{2500} \approx= 10^{752}$, which is huge.

The physics of the model is determined entirely by $e^{-\beta E}$. Since $E \propto J$ the only values that appear in the model, apart from the values of the spins, is the product $\beta J = J/(k_B T)$. We could use real values for these, for example J would be some factor of $\hbar$ and $k_B \approx 10^{-23}\,\mathrm{J\,K^{-1}}$. However, these incredibly small numbers would cause huge floating point errors. Instead it is better to use the fact that these parameters only ever appear in ratio, J/k_B, and so we could just use $J/k_B \approx 10^{-12}\,\mathrm{s\,K}$. This is still tiny however and will lead to floating point issues. A better thing to do is to set $J = k_B = 1$, and then use T as the parameter measured in some units such that $J = k_B = 1$.

It is energetically favourable for all of the spins to align, either all up or all down. This means that there are two states which are much more likely than all other states, in particular they are much more likely than disordered states. On the other hand since 2^N is likely very large the probability of getting either of these two states randomly is 2^{-N}, which is approximately zero. Instead we should generate states with weight $e^{-\beta E_i}$. This is called **importance sampling**, where we generate states according to how likely they are, rather than uniform sampling. If we do this then we find that

$$\langle A \rangle = \frac{\sum_i A_i}{n}, \tag{1.4.7}$$

having already accounted for the factor of $e^{-\beta E_i}$ in our sampling.

There are two ways to do this, which will be the topic of the next chapter.

Two

Dynamics

2.1 Markov Chains and the Metropolis Algorithm

As discussed in the previous chapter random uniform sampling is to simplistic for most physics simulations. Things don't happen in isolation, the surroundings have an important effect. Our goal in this chapter will be to develop a way of simulating this that allows us to select states in a way that reflects the underlying physics. We will do so using Markov chains and a version of the Metropolis algorithm.

For our purposes a **Markov chain** is a way of generating a new state based only on the current state in a way that is physically reasonable. If we start with the state μ_1, we can generate the state μ_2, and then from this we generate μ_3. Importantly the probability that μ_3 is a given state should depend only on μ_2, and not on μ_1. We say that the Markov chain is "memoryless" since it "can't remember μ_1" once it leaves this state.

After generating the states we will have a chain of states, $\mu_1 \to \mu_2 \to \mu_3 \to \cdots \to \mu_n$. The process for generating the next state is what we call the dynamics of the system. We will consider two different dynamical algorithms later in this chapter.

The Metropolis algorithm is used in both dynamical algorithms which we will discuss. The general process for the metropolis algorithm is as follows. We start with the state μ_i and we want to generate a state μ_{i+1} in a physically reasonable way.

■ Code 2.1.1 — Metropolis Algorithm

```
generate a new state, μ_{i+1}
compute the energy change, ΔE
if ΔE < 0:
    accept the new state, μ_{i+1}
else:
    accept the new state with probability e^{-βΔE}
repeat until a new state has been accepted
```

An alternative way to state the if clause is to accept the new state with probability $\min\{1, e^{-\beta\Delta E}\}$. This algorithm is very simple, we just have to specify how to generate μ_{i+1}.

2.2 Glauber Dynamics

Glauber dynamics is the simplest algorithm for generating the next state. We simply choose a random site and flip the spin there.

2.2.1 Computing Energy

When deciding if we accept the result of this flip we need to compute ΔE. The naive way to do this is to calculate the total energy after the flip and before the flip and compute the difference. However, when doing this there is a lot of extra computation occurring that is unnecessary. Suppose that we choose to flip the ith spin for some fixed i. Then

$$E_{\text{before}} = -J \sum_{\langle j,k \rangle} S_j S_k = -JS_i \sum_{\langle i,k \rangle} S_k - J \sum_{\substack{\langle j,k \rangle \\ j \neq i}} S_j S_k. \tag{2.2.1}$$

Here S_i is the value *before* the spin is flipped. Note that the sum over $\langle i,k \rangle$ keeps i fixed and sums k over the nearest neighbours of i. The energy after the spin is flipped is

$$E_{\text{before}} = J \sum_{\langle j,k \rangle} S_j S_k = -JS_i \sum_{\langle i,k \rangle} S_k - J \sum_{\substack{\langle j,k \rangle \\ j \neq i}} S_j S_k. \tag{2.2.2}$$

Recall that the spins are represented by ± 1 so a spin flip is equivalent to a sign change. We can then compute the change in energy:

$$\Delta E = E_{\text{after}} - E_{\text{before}} = 2JS_i \sum_{\langle i,k \rangle} S_k. \tag{2.2.3}$$

This is much quicker to compute and better yet can be done in constant time, $O(1)$.

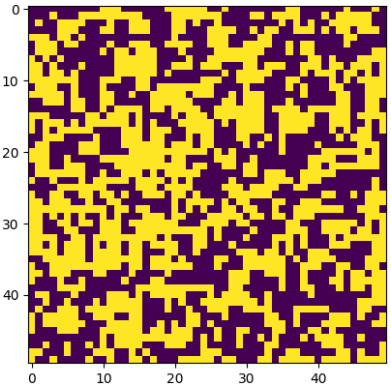
Sometimes we will need to compute the total energy of the system. The naive way to do this is to compute

$$E = -J \sum_{\langle i,j \rangle} S_i S_j \tag{2.2.4}$$

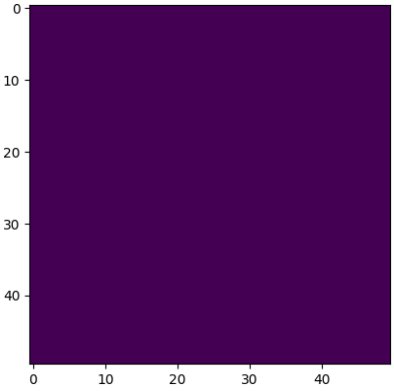
where i now ranges over all sites and j takes on the value of the nearest neighbours indices. A better way to do this relies on recognising that each term in the sum is symmetric in i and j. Therefore when computing, for example, the $S_5 S_6$ term we should also compute the $S_6 S_5$ term. For example, we may sum over the lattice but only account for the neighbours above and to the right, the neighbour to the left will be accounted for when we compute the term in the sum for the spin to the left and similarly with the site below. Both of these sums are $O(N)$ but by exploiting the symmetry like this the sum should be quicker to compute.

2.2.2 Results of Glauber Dynamics

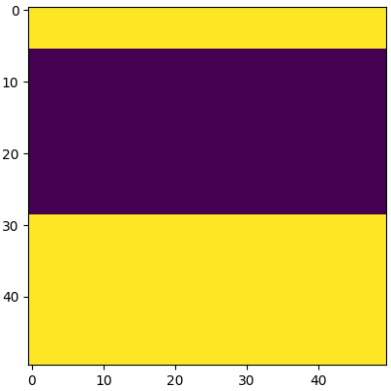
The results with Glauber dynamics depend on the temperature. If the temperature is very high then $e^{-\beta \Delta E} \approx 1$ and so we always swap the spin, resulting in essentially random swaps and no order even after a long time. If the temperature is very low then $e^{-\beta \Delta E} \approx 0$ and we only swap the spin if the energy decreases. This means that we end up with alignment of spins increasing so we will eventually achieve a state where all of the spins are aligned. At low temperatures we can get stuck in a



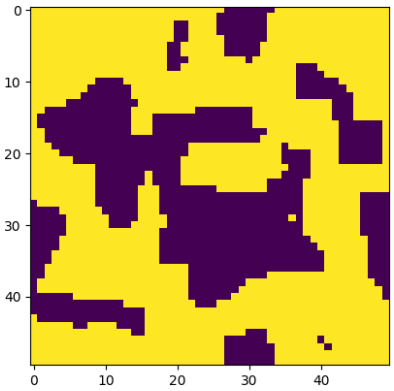
(a) The result of running Glauber dynamics at a high temperature. Here a "high temperature" is $T = 5$ in units where $k_B = J = 1$. Notice that the pattern of spins appears fairly random but there are slightly more yellow sites than purple.



(b) The result of running Glauber dynamics at a low temperature. Here a "low temperature" is $T = 0.01$ in units where $k_B = J = 1$.



(c) A metastable state in the Glauber dynamics. The boundary being straight lines minimises the number of neighbouring unaligned spins. This state will remain for a long time but eventually all spins will align.



(d) The result of 10 sweeps with low temperature Glauber dynamics. One sweep meaning attempting to change N spins, where N is the total number of spins (here $N = 2500$. Even after a relatively short time the spins separate into domains.

Figure 2.1: Various results of running Glauber dynamics for $N = 50^2 = 2500$ spins.

metastable state for a long time where we have two regions with opposite spins and the boundary contributes very little to the total energy and the chance of picking a state on the boundary is also low, resulting in a very slow change towards all the spins being aligned.

2.3 Kawasaki Dynamics

In Glauber dynamics we often end up with all spins aligned. This isn't always reasonable. Sometimes the net **magnetisation** of a system,

$$M \coloneqq \sum_{i=1}^{N} S_i, \tag{2.3.1}$$

is fixed. One way to model this sort of system is with Kawasaki dynamics.

In Kawasaki dynamics we generate the next state by choosing two random spins and then swapping them. We now consider only what happens if the chosen spins are different since if their the same nothing changes. If these spins *aren't* nearest neighbours then the change in energy is simply

$$\Delta E = 2JS_i \sum_{\langle i,k \rangle} S_k + 2JS_j \sum_{\langle j,l \rangle} S_l \tag{2.3.2}$$

where i and j are the indices of the spins we swap. The logic for this is that this is essentially the same as repeating the Glauber dynamics process twice with these two spins chosen.

If the spins are nearest neighbours then we have to be more careful calculating the energy change but it can still be done in an $O(1)$ way by carefully considering which spins are actually effected by the swap.

Since we are only swapping spins the net magnetisation remains the same. This means that the results tend to have spins group into domains where they are all aligned since this minimises the energy. Of course if the temperature is high enough then entropy dominates and we get essentially random states indistinguishable from Glauber dynamics.

2.4 Measurements

Once we have a simulation running one of these dynamical algorithms we want to make measurements of the system. One of our goals will be to find the **critical temperature**, T_c, which is the temperature at which the system goes from ordered to random. This is the temperature which distinguishes the "low temperature" cases from the "high temperature" cases.

Two observables which we have already mentioned are the energy,

$$E = -J \sum_{\langle i,j \rangle} S_i S_j, \tag{2.4.1}$$

and magnetisation,

$$M = \sum_{i=1}^{N} S_i. \tag{2.4.2}$$

We expect that the energy will be much lower for an ordered phase and much higher for a disordered phase since spins align more in the ordered phase. For Glauber dynamics the magnetisation will tend to $\pm N$ for low temperatures as all spins align. For this reason we consider the absolute value of the magnetisation, since there is

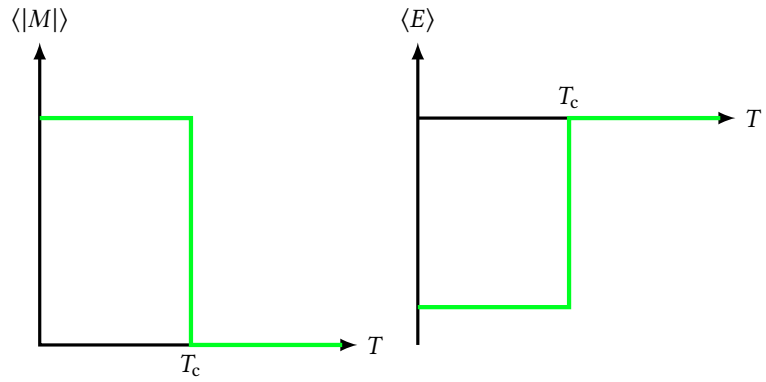


Figure 2.2: Plots of the average absolute magnetisation and energy for an infinite number of spins. Notice the sudden change at the critical temperature, T_c.

nor real logic to assigning one spin as +1 and the other as −1 so the sign cannot have physical meaning. For Kawasaki dynamics M is constant so not that useful.

The average energy is given by the Monte Carlo approximation as

$$\langle E \rangle = \frac{1}{n} \sum_{i=1}^{n} E(\mu_i) \tag{2.4.3}$$

where $\{\mu_i\}$ are all states visited and $E(\mu_i)$ is the energy in state μ_i, and n is the total number of states visited, we take n to be large for this to be a good approximation. Similarly the average absolute magnetisation is

$$\langle |M| \rangle = \frac{1}{n} \sum_{i=1}^{n} |M(\mu_i)| \tag{2.4.4}$$

where $M(\mu_i)$ is the magnetisation in state μ_i.

Consider what happens in Glauber dynamics. At a low temperature we expect that the spins align and so $\langle E \rangle = -2JN$, since either S_i and S_j are both +1 or both −1, either way their product is +1 and so we eventually achieve constant (up to random fluctuations of a few spins) energy $-2JN$ and over enough time the states that occurred before achieving this have negligible impact on the average. At high temperatures the spins are essentially random so S_i and S_j are just as likely to have the same sign as the opposite sign and hence in the energy sum most terms cancel out and we get $\langle E \rangle = 0$, again, up to random fluctuations.

The case of magnetisation is similar with $\langle |M| \rangle = N$ at low temperatures and $\langle |M| \rangle = 0$ at high temperatures.

How can we use this to find the critical temperature? First we consider what happens when the number of spins, N, tends to infinity. In this case we get two very clear regions of low and high temperature. This is shown in Figure 2.2. Notice the discontinuity at the critical temperature, T_c. For small N however this discontinuity can't be seen.

Instead of the discontinuity we use the fact that fluctuations tend to be very large near the critical temperature as the system moves back and forth between the two phases. One measure of the fluctuations is the standard deviation,

$$\sigma_E \coloneqq \langle E^2 \rangle - \langle E \rangle^2. \tag{2.4.5}$$

However, by dividing by $Nk_{\mathrm{B}}T^2$ we can associate this with the specific heat capacity, C, per spin:

$$\frac{\langle E^2\rangle - \langle E\rangle^2}{Nk_{\mathrm{B}}T^2} = \frac{C}{N}. \tag{2.4.6}$$

Similarly for the magnetisation we can relate the standard deviation to the susceptibility, χ, per spin:

$$\frac{\langle M^2\rangle - \langle M\rangle^2}{Nk_{\mathrm{B}}T} = \frac{\chi}{N}. \tag{2.4.7}$$

Plotting these quantities will give a peak at T_{c}. The height of the peak will vary with $\ln N$ whereas away from the peak the values will be approximately constant with respect to N.

Index