Simulation of Two-Dimensional Ising Model

Kim Dahl-Hansen, Fridtjof Ronge Gjengset,
Sebastian Ranum Tenmann & Cecilie Klarpås

(Dated: November 22, 2021)

We implemented a Metropolis Markov chain Monte Carlo algorithm to numerically explore the properties and temperature-dependent phase transition behaviour of ferromagnetic materials through a two-dimensional Ising model, as solving this problem analytically would exceed time. To do this we modelled spin lattices with sizes ranging from 2 by 2 to 100 by 100 spins, and over a range of different temperatures. We observed that our numerically obtained values nicely converged towards the expected analytical values, validating the correctness of the code. In the end we estimated the critical temperature of an infinitely large spin lattice to be $T_c(L=\infty) \approx 2.254 \pm 1.661$, which tended towards the theoretical value of $T_c(L=\infty) = 2.269 J/k_B$, although a lack of sample data made out prediction unreliable. Using a computer with more cores would help us obtain T_c values faster, and improving the estimation of T_c when $L \to \infty$, as our code has some parallelisation optimisations.

Our code can be found at GitHub [a]

I. INTRODUCTION

In general, calculating parameters of any system where particles interact is a laborious process as each particle needs to be individually evaluated to see how it impacts the behavior of the system as a whole, making it tedious to calculate analytically. The calculation of critical temperatures in ferromagnetic materials(FMs) are no exception. The various properties of a system close to a critical phenomena is an area of physics that still lacks a satisfactory understanding.[1] Fortunately, modern computers are very fast at repetitive simple calculations, and we have the numerical methods and models to approximate the behavior of this type of system.

In this project, we used the the Markov chain Monte Carlo (MCMC) method with the **Ising model** to analyse the behavior of FMs undergoing phase transition from a magnetised phase to a no-net magnetic phase as a result of reaching critical temperatures.

As MCMC method is excellent at handling high dimensional problems, it is a good option for calculating several properties and the behavior of FMs. This is because the time complexity of MCMC does not scale with the number of dimensions like other algorithms such as the trapezoid algorithm.

In this project we simplified the ferromagnetic material to be a 2D lattice $\mathbf{s}$ of particles with dimensions $L \times L$, where each particle has a spin s_{ij}. In order to make our results applicable and comparable for varying system sizes L, we scaled our equations of energy, magnetisation, heat capacity and magnetic susceptibility to be expressed in quantities per spin.

Before looking at the results from our simulation we provide some analytical calculations for comparison. We also look at some analytical results we used to speed up our numerical solution. We then present the results from our simulation.

First, we look at how our numerical solution compares to our analytical solution, how many Monte Carlo cycles we need to run to approach the analytical results, and how much burn-in time we can expect from our simulation.

Secondly, we look at and discuss what kind of probability distribution we get when we look at *energy per spin* ϵ for different temperatures T.

Thirdly, we discuss some ways we were able to optimise the code using parallelisation, why we can parallelise certain parts of the code, and how much speedup we see after enabling parallelisation.

Lastly, we examine the phase transition of the system by looking at the behavior of the absolute expected *magnetisation per spin* $\langle|m|\rangle$, the expected *energy per spin* $\langle\epsilon\rangle$, the *specific hear capacity* C_V and the *susceptibility* χ, as a function of different lattice sizes L and the temperatures T.

We aim to provide a useful discussion for each system parameter we have touched on, as well as some overarching comments about the performance and value of the simulation we created.

[a] Check out git: https://github.com/cecilmkl/comp_phys/tree/main/project4

II. METHODS

A. Analytical approach

We study two-dimensional lattices of size $L \times L$ of $N = L^2$ spins with two possible states $s_i \in \{-1, +1\}$. The lattice refers to the spin state of the system and is defined by the spin configuration (or microstate)

$$\mathbf{s} = \begin{bmatrix} s_{11} & \dots & s_{1L} \\ \vdots & \ddots & \vdots \\ s_{L1} & \dots & s_{LL} \end{bmatrix}, \tag{1}$$

where we imposed periodic boundary conditions such that each spin has exactly four neighbours.

The total energy of the system is given by

$$E(\mathbf{s}) = -J \sum_{<kl>}^{N} s_k s_l, \tag{2}$$

where $< kl >$ denotes the sum over all neighbouring pairs without double-counting. The J is the coupling constant that sets the energy associated with each interaction between neighbouring spin pairs. From equation 2, we can note that the energy is minimised if all the spins point in the same direction, as $\sum_{<kl>}^{N} s_k s_l$ will have no cancelling terms and be positive. Keep in mind that this is a relative energy not an absolute, and can therefore be negative.

The total magnetisation of the system is the sum of all spins,

$$M(\mathbf{s}) = \sum_{i}^{N} s_i, \tag{3}$$

and is therefore unitless, since s_i er unitless.

Then, the normalised values are given as the energy per spin

$$\epsilon(\mathbf{s}) = \frac{1}{N} E(\mathbf{s}). \tag{4}$$

and the magnetisation per spin

$$m(\mathbf{s}) = \frac{1}{N} M(\mathbf{s}). \tag{5}$$

Since both the energy and the magnetisation are defined as functions of the spin configuration $\mathbf{s}$, we can use its probability density to find the expectation values of the energy and magnetisation per spin. The expectation values are the weighted sum of all possible outcomes where each weight corresponds to the probability of the value and is therefore interesting to explore.

The spin configuration follows a probability density function defined by the Boltzmann distribution

$$p(\mathbf{s}; T) = \frac{1}{Z} \exp\{-\beta E(\mathbf{s})\}, \tag{6}$$

which is a special case of the exponential distribution. The partition function Z is given by

$$Z = \sum_{\mathbf{s} \in D} \exp\{-\beta E(\mathbf{s})\}, \tag{7}$$

where $\mathbf{s} \in D$ indicates all possible spin configurations of the system and

$$\beta = \frac{1}{k_B T}, \tag{8}$$

is the “inverse temperature” with temperature of the system, T, and Boltzmann’s constant k_B.

The expectation values are then found by

$$\langle \epsilon \rangle = \frac{1}{N} \langle E \rangle = \frac{1}{N} \sum_{\mathbf{s} \in D} E(\mathbf{s}) \cdot p(\mathbf{s}; T), \tag{9}$$

and

$$\langle |m| \rangle = \frac{1}{N} \langle M \rangle = \frac{1}{N} \sum_{\mathbf{s} \in D} M(\mathbf{s}) \cdot p(\mathbf{s}; T), \tag{10}$$

where $\mathbf{s} \in D$ indicates all possible spin configurations of the system. The domain for $\mathbf{s}$ for N spins consist of 2^N possible states which is difficult to explore for large N. To avoid this, we estimated the expectation values by using the mean of the generated samples, as this is an unbiased estimator. This results in:

$$\langle \epsilon \rangle \approx \bar{\epsilon} = \frac{1}{n_s} \sum_{i=1}^{n_s} \epsilon_i, \tag{11}$$

and

$$\langle |m| \rangle \approx \bar{|m|} = \frac{1}{n_s} \sum_{i=1}^{n_s} |m|_i, \tag{12}$$

where n_s is the number of samples.

The expectation values are not the full story of the values of ϵ we might expect and the probability of such. We therefore wish to explore the probability distribution $p_\epsilon(\epsilon)$ and some additional information about it.

The variance is a measure of how spread out $p_\epsilon(\epsilon)$ is and is defined as

$$\sigma^2 = \left\langle (\epsilon - \langle \epsilon \rangle)^2 \right\rangle, \tag{13}$$

which can be estimated as

$$\sigma^2 \approx \frac{1}{n_s - 1} \sum_{i=1}^{n_s} (\epsilon_i - \bar{\epsilon}). \tag{14}$$

This will in turn give the standard deviation of the ϵ-samples

$$\sigma = \sqrt{\sigma^2}. \tag{15}$$

The expectation values can then be used to calculate the specific heat capacity

$$C_v = \frac{1}{N}\frac{1}{k_B T^2}\Big(\left\langle E^2\right\rangle - (\langle E\rangle)^2\Big), \tag{16}$$

and the susceptibility

$$\chi = \frac{1}{N}\frac{1}{k_B T}\Big(\left\langle M^2\right\rangle - (\langle |M|\rangle)^2\Big), \tag{17}$$

where both are normalised to number of spins. This can be done using the relation $E = N\cdot\epsilon$ and $|M| = N\cdot|m|$.

In an effort to verify our numerical estimations of $\langle\epsilon\rangle$, $\langle|m|\rangle$, C_v and χ, we calculated each expression for the simple 2×2 lattice of $N = 4$ spins and compared these to our numerical estimates. The analytical expressions can be found in V B.

B. Critical temperature for phase transitions

In the two dimensional Ising model, there is a second order phase transition happening at a critical temperature T_c that depends on the size of the system. What this means is that for a temperature higher than the critical temperature the system experience no net magnetisation, whereas below T_c it experiences spontaneous magnetisation.[1] We have observed this by simulating the system for lattice sizes ranging from 40×40, 60×60, 80×80 and 100×100, and with temperatures ranging from 2.1 J/k_B-2.4 J/k_B. From this we made plots of each parameter $\langle\epsilon\rangle$, $\langle|m|\rangle$, C_v and χ. This enabled us to observe the phase transitions, and compare it to the analytical value. We could further compare the results for our finite system, to the ones found by Lars Onsager for an infinite system $(L\to\infty)$[2]:

$$T_c(L=\infty) = \frac{2}{\ln 1+\sqrt{2}} J/k_B \approx 2.269 J/k_B. \tag{18}$$

We do this by looking at the critical temperatures and lattice sizes for our plots, and use a scaling relation:

$$T_c(L) - T_c(L\to\infty) = aL^{-1}, \tag{19}$$

where a is a constant, to estimate a critical temperature for an infinite lattice.

We can use this relation to perform a linear fit to critical temperatures of different sized lattices against the inverse of their lattice sizes. The intercept of this linear regression will then be be a numerical estimation of the critical temperature for an infinitely large lattice as $\frac{1}{L}\to 0 \implies L\to\infty$.

C. Improving numerical efficiency

When solving this problem numerically, we would naturally have to compute the Boltzmann factor several times. However, looking at a 2d lattice with an arbitrary size containing the spins of each particle we can notice a useful pattern:

$$\Delta E = E_{\text{after}} - E_{\text{before}}. \tag{20}$$

The total energy shift of the system ΔE when flipping the spin of a single particle can only take one of five unique values, which is dependent on the particles initial spin and the spin of the neighbouring particles[1].

As ΔE can only take one of five values, the Boltzmann factor $e^{-\beta\Delta E}$ can also only take one of five values for a given temperature T. This allows us to create a lookup table for the Boltzmann factor for a given temperature instead of repeatedly calling the expensive exponential function. This made makes our program for efficient and decreases run-time.

We want to simulate a lattice that wraps around, that is, the leftmost particle's left neighbour is the rightmost particle in the same row, and the same goes for all directions. It is tempting to use an if-statement to calculate the neighbouring spin for the particles at the edge of the lattice, however if-statements would significantly slow down our code. We opt instead to use the modulo operator to solve this problem. As $(i+L)\%L$ when i goes from 0 to L will give numbers from 0 to $L-1$ and then for $i = L$ give $(L+L)\%L = 0$, we effectively have an iteration that wraps around to the first index. We can use this method of indexing to avoid the use of if-statements for the particles at the end of the lattice.[2]

The algorithm

The MCMC method produce a Markov chain of spin configuration samples generated using a proposal probability density function that only depends on the current state.

To ensure that the samples $\mathbf{s}$ were distributed according to the Boltzmann distribution, we used the Metropolis-Hastings algorithm as the rule of acceptance. How this is enabled is shown in algorithm 1

The rule of acceptance ensures that only values that upholds

$$r < \frac{p(\mathbf{s}')}{p(\mathbf{s}_i)} \tag{21}$$

where $r\sim U(0,1)$, $\mathbf{s}'$ is the candidate spin configuration after a possible spin flip and $\mathbf{s}_i$ is the previous state. The

[1] This is shown in detail in appendix V D

[2] A bit more detailed explanation is offered in the appendix V C

Algorithm 1 Markov chain Monte Carlo Metropolis

This algorithm shows a cycle of the MCMC method on the Ising model, with the Metropolis condition.

```
procedure
    Generate initial lattice                              ▷ Random or ordered
    Pre-calculate possible energy changes $w$ from a spin flip
and their corresponding probabilities according to eq. 6.
    for each spin in lattice do
        spin *= -1                                        ▷ Flip a spin
        E += dE                     ▷ Compute new energy of system
resulting from flip.
        r = uniform(rng)   ▷ Generate random number from
uniform distribution r.
        If $r < w$ then accept the new state.
        Update total energy and magnetisation of the sys-
tem.
```

right hand side can be evaluated efficiently by reducing the expression to the following

$$\frac{p(\mathbf{s}')}{p(\mathbf{s}_i)} = \frac{\exp\{-\beta E(\mathbf{s}')\}/Z}{\exp\{-\beta E(\mathbf{s}_i)\}/Z} \tag{22}$$

$$= \frac{\exp\{-\beta E(\mathbf{s}')\}}{\exp\{-\beta E(\mathbf{s}_i)\}} \tag{23}$$

$$= \exp\{-\beta E(\mathbf{s}') + \beta E(\mathbf{s}_i)\} \tag{24}$$

$$= \exp\Big\{-\beta\Big(E(\mathbf{s}_i) - E(\mathbf{s}')\Big)\Big\} \tag{25}$$

$$= \exp\{-\beta\Delta E\}. \tag{26}$$

D. Parallelisation

We further improved our code by parallelising the simulation of the system for different temperatures. As these calculations are completely independent of each other, it makes for an easy way to make our program more efficient when calculating for different temperatures. We chose to use **OpenMP** to parallelise our code.

Algorithm 2 Code parallelisation

This is a pseudo-code of how we parallelised our code for different temperatures T, where ΔT is just a small temperature change.

```
#pragma omp for                                ▷ Start parallel region
for T=2.0, T< 2.4, $T + \Delta T$ do
    Ising()                                          ▷ Create model
    Simulate(T)                      ▷ Simulate for temperature T
    #pragma omp critical                          ▷ Prevent conflict
    SaveSample()                                ▷ Save results to file
show(results)                                  ▷ End of parallel region
```

From algorithm 2, note that we use "**#pragma** omp crictial" to prevent the different threads from overwriting each other when writing to file.

III. RESULTS & DISCUSSION

A. Verifying our model with analytical results

For our model to carry any weight at all, we first needed to verify that it does what it is meant to do. We did this by running our model with a 2x2 lattice with unordered spins, temperature $T = 1.0\ J/k_B$, so that we could calculate the analytical results using the analytical expression found in appendix V B. Note that one MC cycle corresponds to N attempted flips, where N is the number of particles in the system, in this case $N = 4$. We then plotted our results for $\langle\epsilon\rangle$, C_v, $\langle|m|\rangle$ and χ:

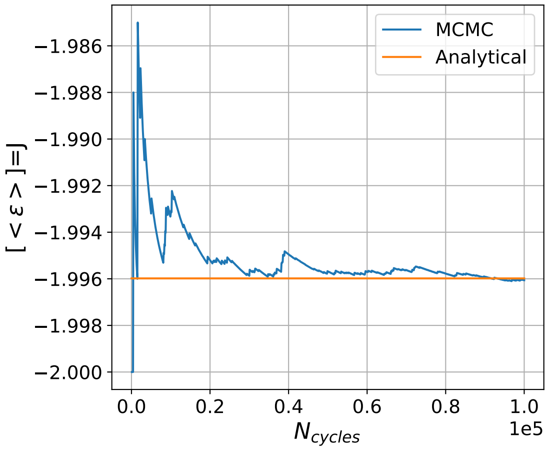

FIG. 1. This figure shows the expected energy per particle $\langle\epsilon\rangle$, as a function of how many Monte Carlo Cycles we used. The simulation was ran with a 2x2 lattice with unordered spins.

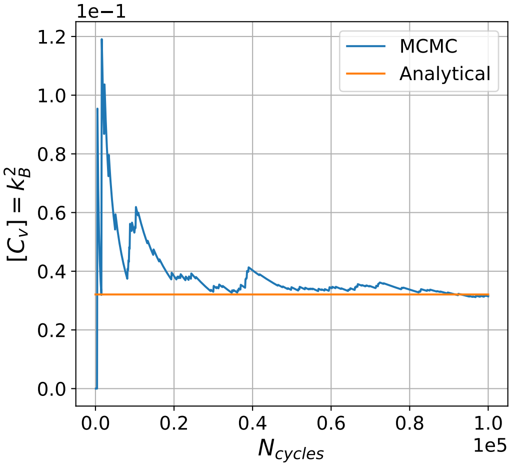

FIG. 2. This figure shows the specific heat capacity C_v (Normalised to number of spins), as a function of how many Monte Carlo Cycles we used. The simulation was ran with a 2x2 lattice with unordered spins.

We see that figure 1 and 2 bares a striking resemblance, with the only distinguishing factor being the values on the y-axis. However, this is to be expected as the total energy of the system that is used to calculate C_v is dependent on the energy per spin. As there is no other varying dependencies when calculating C_v, we expect to see the same shape of the graph, but with different scale and a different y-axis intercept, as we do in 1 and 2.

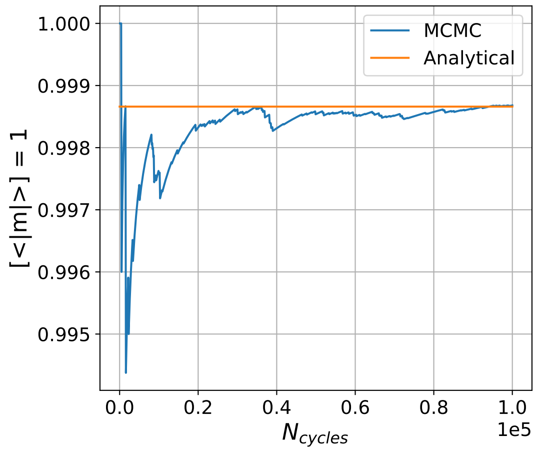

FIG. 3. This figure shows the expected magnetisation per particle $\langle|m|\rangle$, as a function of how many Monte Carlo Cycles we used. The simulation was ran with a 2x2 lattice with unordered spins.

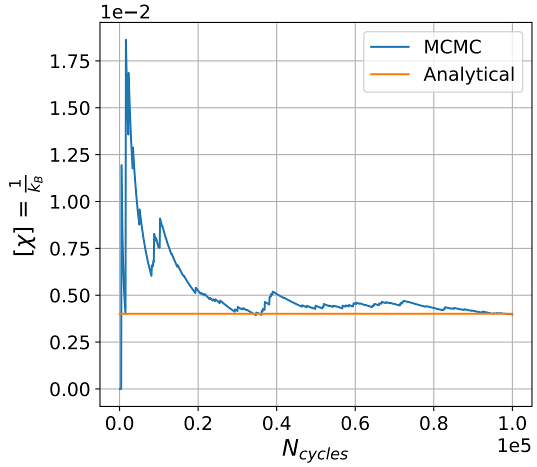

FIG. 4. This figure shows the susceptibility χ (Normalised to number of spins), as a function of how many Monte Carlo Cycles we used. The simulation was ran with a 2x2 lattice with unordered spins.

We see the same similarities between figure 3 and 4 as we did with 1 and 2, and it can be explained with the same rationale. As we use the total magnetisation of the system M, which is dependent on the magnetisation per spin m, to calculate χ with no other *varying* contributing factors, we expect to see the same shaped graph. In this case the graph is also flipped in the x-direction, as well as having a different scale and y-axis intercept.

We see in figures 1, 2, 3 and 4, that all the values $\langle\epsilon\rangle$, C_v, $\langle|m|\rangle$ and χ has a drastically better approximation of the analytical values at around 50 000 MC cycles. However, they settle on their analytical values after around 100 000 MC cycles.

B. Examining burn-in time

As we saw in the previous section, our model spends a certain number of cycles before it approaches the analytical values. This is due to what is called **burn in time**. We examine the burn in time by simulating a 20 × 20 lattice for different temperatures. We run the simulation both for the case where the spins of the particle are initially ordered with all positive spin, and for the case where the initial spin for each particle is chosen at random.

We start by looking at the plots for $\langle\epsilon\rangle$ and $\langle|m|\rangle$ case where $T = 1.0\ J/k_B$.

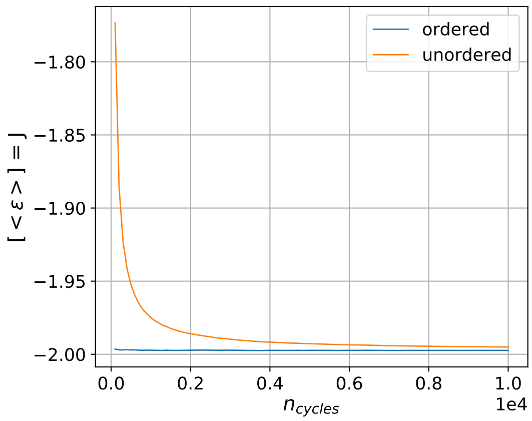

FIG. 5. This figure shows $\langle\epsilon\rangle$ as a function of the number of MC cycles for $T = 1.0\ J/k_B$. For the ordered graph, all the particles initially have a positive spin, while in the unordered, each particles initial spin is chosen at random.

Looking at figure 5 and 6 we see a similar pattern. In the unordered case, both $\langle\epsilon\rangle$ and $\langle|m|\rangle$ take a longer time to converge than in the ordered case, which uses practically none, to understand why this is, we need to consider the problem itself.

Each particle has a higher chance of flipping if the energy of the system decreases when the particle flips, the most likely state of the system is therefore a state where the energy of the system is minimised. However,

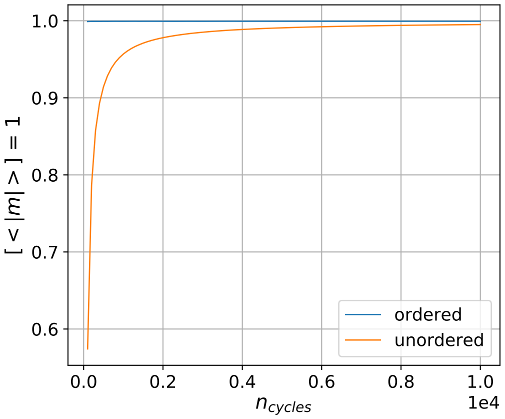

FIG. 6. This figure shows $\langle m\rangle$ as a function of the number of MC cycles for $T = 1.0\ J/k_B$. For the ordered graph, all the particles initially have a positive spin, while in the unordered, each particles initial spin is chosen at random.

in the case of ordered spins, where the particles spin of each particle is 1, the energy is already minimised, as the sum in the equation for the total energy 2, will be positive and have no cancelling terms.

This places the initial state with ordered spin in a high probability region, which will cause a low to no burn in time. With particles having a random initial spin, we are more likely to find ourselves in a low probability region where the energy is maximised, and we have to “find” the high probability region, which will increase burn in time.

This is exactly what we see in both figure 5 and 6. The burn in time for the unordered state seems to be $\sim$ 1000 cycles before it starts giving similar(high probability) values. For the ordered case we see no burn in time, as we are already in the high probability region.

Moving on, we look at the plot for the same values, but for temperature $T = 2.4\ J/k_B$:

In figure 7 and 8 we see that the valuers for $\langle\epsilon\rangle$ and $\langle|m|\rangle$ stabilise at a different value than in figure 5 and 6. This is to be expected as we changed the temperature. However, we now see that the ordered and unordered cases have about the same burn in time of $\sim$ 300 cycles. As the temperature is raised, this places the unordered initial state of the system for $\langle\epsilon\rangle$ closer to the high probability region, which will explain why it has a lower burn in time. We also expect the $\langle|m|\rangle$ to decrease for higher temperatures, which is what we see if we compare 6 and 8.

We now know that we can expect a burn in time of about $\sim$ 1000 cycles at a maximum, as we saw from the extreme case in 5 and 6. Disregarding the values that are produced during burn in time, will make out prediction more accurate, as we don’t include the cases where the system is in a very low probability region.

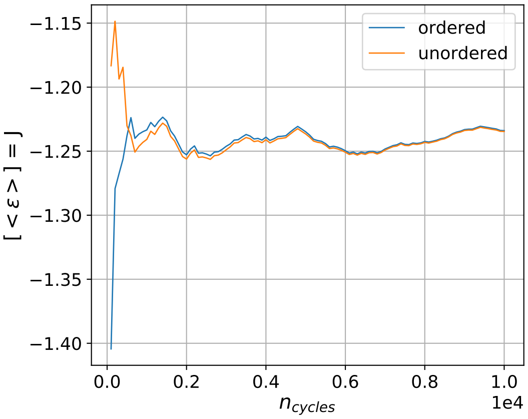

FIG. 7. This figure shows $\langle\epsilon\rangle$ as a function of the number of MC cycles for $T = 2.4\ J/k_B$. For the ordered graph, all the particles initially have a positive spin, while in the unordered, each particles initial spin is chosen at random.

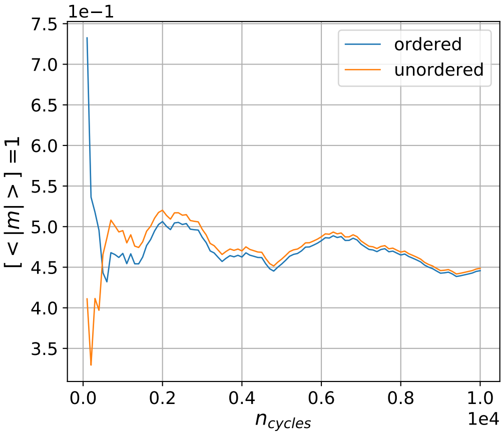

FIG. 8. This figure shows $\langle m\rangle$ as a function of the number of MC cycles for $T = 2.4\ J/k_B$. For the ordered graph, all the particles initially have a positive spin, while in the unordered, each particles initial spin is chosen at random.

It is worth noting that when looking at figure 1 for the 2×2 lattice, we could estimate a burn in time of about $\sim$ 10 000 cycles, which is way higher. Because the system is so small(only 4 particles), a change in one spin will therefore also produce radically different $\langle\epsilon\rangle$. Even after the system has reached it equilibrium, flipping one particle will place it back in the low probability region. We believe that we get a better estimation, not because the system is in a high probability region, but rather because we calculate the average value over all

	T = 1.0	T = 2.4
n_{samples}	288242	107600807
$\bar{\epsilon}$	-1.99	-1.21
σ^2	0.00017	0.019
σ	0.013	0.14

TABLE I. This table shows the number of accepted ϵ sample values for 10^6 MC cycles in addition to the estimated expectation value, variance and standard deviation of ϵ for $T = 1.0$ and $T = 2.4$

the cycles. This would explain why we would need a large number of cycles to get a good estimation.

C. Probability Distribution

We will now look at the probability distribution of ϵ, that is p_ϵ, for the two temperatures $T = 1.0\ J/k_B$ and $T = 2.4\ J/k_B$ in a 20×20 lattice. The normalised histogram of each ϵ value can be used to estimate the probability distribution when $n_s \to \infty$ samples. We chose to run 10^6 MC cycles and obtained every ϵ value for each accepted spin flip. The normalised occurrence of each value is scaled logarithmically on the y-axis.

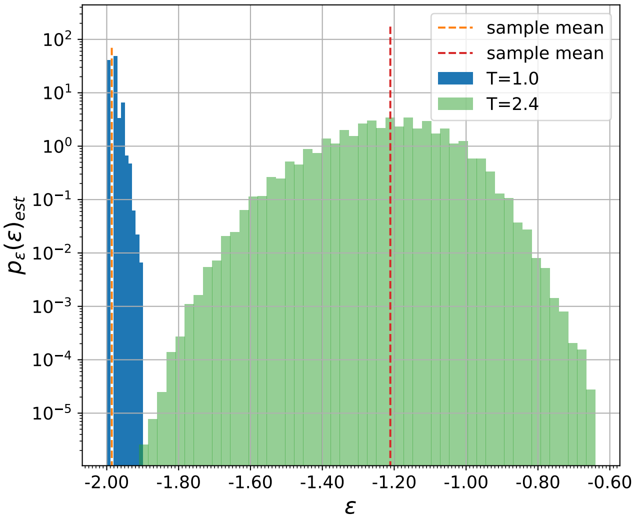

FIG. 9. This figure shows a histogram estimate of the probability density function p_ϵ for ϵ, at the two temperatures $T = 1.0\ J/k_B$ and $T = 2.4\ J/k_B$. The ϵ values are shown on the x-axis, and the probability density p_ϵ is shown on the y-axis on logarithmic scale. The lowest temperature were plotted with 10 bins, while the highest temperature has 50 bins. The dotted lines show the mean value of the two distributions.

Looking at figure 9 we see that the two distributions for $T = 1.0\ J/k_B$ and $T = 2.4\ J/k_B$ differ significantly. First we see that for $T = 1.0\ J/k_B$, we have a narrow distribution with a small variance of $\sigma^2 = 0.0001706$, while the distribution for $T = 2.4\ J/k_B$ covers more values with a variance of $\sigma^2 = 0.018659$ as shown in Table I. This is expected as the chance that a particle flips is dependent on the energy of the system. In the initial state for $T = 1\ J/k_B$ the energy of the system is random. When a particle flips and decreases the energy of the system, it will lower the chances of other particles flipping without lowering the energy of the system. We know that the energy of the system is minimised when the spin of all the particles are equal, and we saw from figure 5 ϵ is very close to this state when $T = 1\ J/k_B$. This means we can that for $T = 1\ J/k_B$ the lower limit for ϵ is -2. In other words, the energy can't be minimised a lot more, and the energy of the system is low, so the probability that a particle flips so that the energy of the system increase is very low. This gives us the distribution we see in figure 9 for $T = 1.0\ J/k_B$

We see that for $T = 2.4\ J/k_B$ we have a much larger standard deviation. As the energy of the system is higher than for $T = 1\ J/k_B$, the particles have a higher chance of flipping into a state that increases the energy, giving a higher deviation as well.

Where the system finds it's equilibrium is dependent on the temperature, as the temperature is used to calculate β which in turn is used to calculate the Boltzmann factor. The Boltzmann factor determines the probability of a particle flipping without minimizing the energy. We can therefore view the temperature as a factor that disturbs the systems ability to minimise it's total energy.

We can compare figure 9 to figure 5 and 7, we see that the state we saw most often for ϵ in figure 9 is the same value we see that $\langle \epsilon \rangle$ converges towards in figure 5 and 7, which is a confirmation that our simulation is working as intended.

D. Parallelisation

When parallelising our code, we measured an average speedup of 2.45375, over 10 runs, when running on 4 cores. The speedup is as expected, as all parallelisation comes at some overhead cost. This is also a trivial parallelisation as it mainly just runs the same function on different cores with different parameters. A limitation of the parallelisation as outlined in algorithm 2 is that the different threads have to wait for their turn when writing to file, to avoid data being overwritten, which will cause some downtime for the the different threads. We could expect a speedup closer to 4 if we parallelised at a deeper level in our code, for example for each flip spin, but we would never reach a speedup of 4 due to the overhead of parallelizing.

E. Phase Transitions

Here we provide our plots for the system parameters over temperatures in the region we expected to find a phase transition $2.0J/k_B - 2.62J/k_B$. Figure 10 shows $\langle\epsilon\rangle$. We find a stronger incline between the temperatures $2.2J/k_B$ and $2.4J/k_B$, as we would expect around the critical temperature, but this is too faint to be any clear indication of a phase transition.

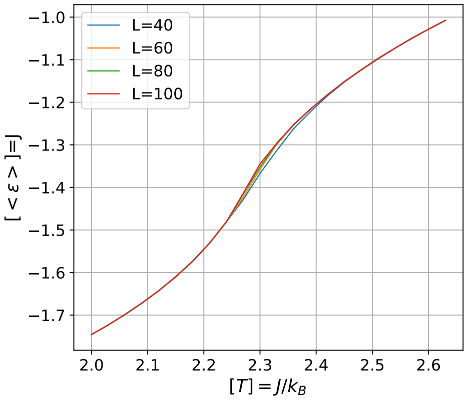

FIG. 10. This figure shows the expectation value for energy per spin $\langle\epsilon\rangle$ as a function of time for temperatures between $2.0J/k_B$ - $2.63J/k_B$ and for lattice side lengths of 40, 60, 80 and 100 spins.

In figure 11 we see the same plot but for $\langle|m|\rangle$. Here we can observe a more significant change after $T = 2.2J/k_B$. As we know that the mean magnetisation approaches zero after the critical temperature, we can see that this agrees with our plots.This further explain the effects shown and discussed in section III C, as the temperature $T = 2.4J/k_B$ will be over the critical temperature we see here.

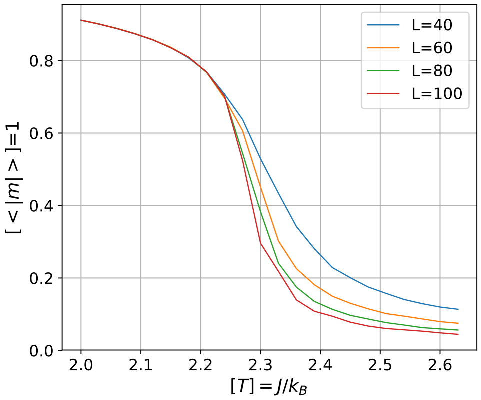

FIG. 11. This figure shows the expectation value for the absolute value of magnetism per spin $\langle|m|\rangle$ as a function of time for temperatures between $2.0J/k_B$ - $2.63J/k_B$ and for lattice side lengths of 40, 60, 80 and 100 spins.

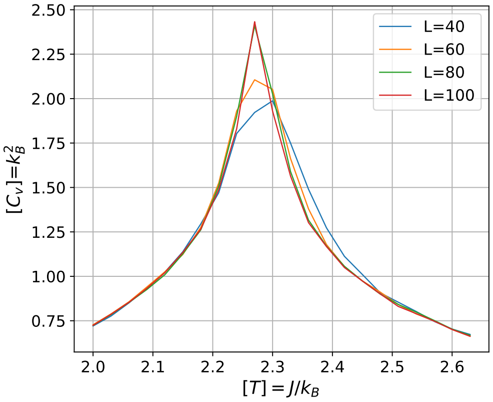

FIG. 12. This figure shows the heat capacity C_v as a function of time for temperatures between $2.0J/k_B$ - $2.63J/k_B$ and for lattice side lengths of 40, 60, 80 and 100 spins.

Lattice size L	$T_c(L)[J/k_B]$	$C_v[k_B^2]$ at $T_c(L)$
40	2.3	1.98775
60	2.27	2.10582
80	2.27	2.41046
100	2.27	2.43277

TABLE II. This table summarises the heat capacity peaks for the various lattice sizes.

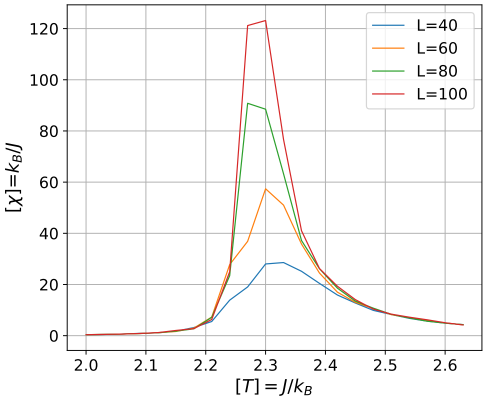

FIG. 13. This figure shows the susceptibility χ as a function of time for temperatures between $2.0J/k_B$ - $2.63J/k_B$ and for lattice side lengths of 40, 60, 80 and 100 spins.

Lattice size L	$T_c(L)[J/k_B]$	$\chi[k_B/J]$ at $T_c(L)$
40	2.33	28.5739
60	2.3	57.4022
80	2.27	90.8113
100	2.3	123.145

TABLE III. This table summarises the susceptibility peaks for the various lattice sizes.

Figure 12 shows the heat capacity C_v. Here we observe a phase transition occurring at $T = 2.3J/k_B$ for the 40 × 40, and $T \approx 2.25J/k_B$ for 60 × 60 and 80 × 80 lattices. We also see that we get a sharp peak at the critical temperature, that gets sharper with an increase in lattice size. Hence we can expect that if the lattice size increased towards infinity, this function would collapse towards a Dirac delta function around the $T = 2.25J/k_B$ area. Our peaks for the various lattice sizes is provided in table II.

In figure 13 we observe a similar trend for the susceptibility as we did for the heat capacity in figure 12. Also here we see a sharpening peak moving towards $T = 2.25J/k_B$ as the lattice size is increased, implying further that this is where our critical temperature T_c is. The peak values is summarised in table III.

From Figure 12&13, and table II&III, we can see that the critical temperature tends towards $T_c(L) \approx 2.27J/k_B$ as the lattice size increases.

We know from equation 18, that we expect the critical temperature to converge towards $T_c(L \to \infty) \approx 2.269J/k_B$ as the lattice size increase. By using the scaling of the critical temperature expressed in equation 19, we have a method for estimating T_c for an infinitely large lattice. We can numerically estimate this value by plotting the critical temperature T_c for a given lattice of size L against $\frac{1}{L}$, and performing a liner regression. The intercept of the linear regression will then give us an estimate of T_c of an infinity large lattice as $\frac{1}{L} \to 0 \implies L \to \infty$.

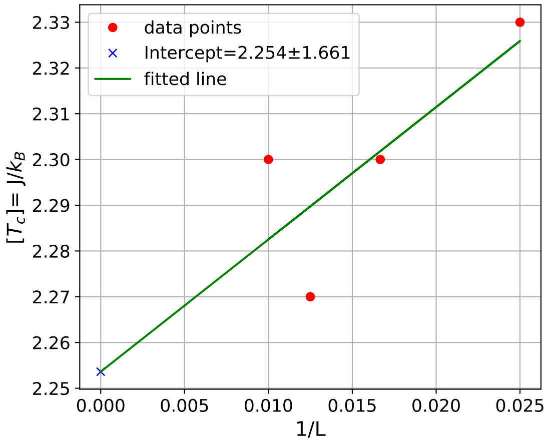

FIG. 14. This figure shows the linear regression of the critical temperatures T_c against $\frac{1}{L}$. The red dots show the data-points we used, the green line shows the regression line, and the blue cross shows in intercept.

We can see from figure 14 that we find the estimated critical temperature for an infinitely large lattice to be $T_c \approx 2.254 \pm 1.661$. Our estimation is pretty close to the theoretical limit that Onsager calculated, although the intercept error is very large. However, there are some improvement's that could be made. We could improve our linear fit by calculating the critical temperature additional sizes of lattices. We could also improve the estimation of the critical temperature for a given lattice of size L. An improved estimation of these critical temperatures could also improve our estimation for the intercept as the data-points used for the linear regression might have been less scattered.

IV. CONCLUSION

Through comparisons with known analytical values for the Ising model, we conclude that we have successfully made a code utilising the MCMC Metropolis method to simulate the Ising model. Doing this, we observed phase transitions occurring at a critical temperature converging towards $T_c \approx 2.254 \pm 1.661$, as the lattice size increase, agreeing with the theoretical value from the literature, but unreliable due to the large errors.

Although the Ising model is useful for simulating the behavior of ferromagnets, it does have some limitations. The run-time for the simulation increases massively as the size of the lattice increases. We can somewhat tackle this issue by parallelising the code, giving some speedup to the simulation, but it still remains an issue. We saw the consequence of this issue when trying to approximate the limit where $L \to \infty$. As we did not have enough samples of T_c, the error of the intercept is very large. Even though we got pretty close to the analytical limit calculated by Onsager, our result is not reliable enough.

Further parallelising the code might improve the run-time of our code a bit, but not a lot as parallelising the code always comes at some overhead cost, and we are only running the program on four cores. As the simulation run for several hours, even after a substantial speedup, we would still require hours to provide a reliable result for T_c when $L \to \infty$, as we would need to calculate for several more sizes of L. We could also use a polynomial fit for the susceptibility to improve our T_c values for a given L, which might also help improve the reliability of T_c when $L \to \infty$, without increasing run-time substantially.

[1] M. Hjort-Jensen, *Computational Physics* (Department of Physics, University of Oslo, 2015) lecture notes Fall 2015.

[2] L. Onsager, Phys. Rev. **65**, 117 (1944).

V. APPENDIX

A. Table of all possible states of the 2 x 2 system

Number of +1 spins	Total energy, $[E(\mathbf{s})] = J$	Total magnetisation, $M(\mathbf{s})$	Degeneracy
0	-8	-4	1
1	0	-2	4
2	0	0	4
2*	8	0	2
3	0	2	4
4	-8	4	1

TABLE IV. This table summarises all possible state of the 2 × 2 lattice system, where * indicates the states with 2 positive spins along one diagonal.

B. Analytical expressions for a 2 x 2 lattice

1. *Partition function*

The partition function is

$$\begin{aligned}
Z &= \sum_{\mathbf{s}\in D} \exp\{-\beta E(\mathbf{s})\} && (27)\\
&= \exp\{-\beta\cdot(-8)\} + 4\cdot\exp\{-\beta\cdot 0\} + 4\cdot\exp\{-\beta\cdot 0\} + 2\cdot\exp\{-\beta\cdot 8\} + 4\cdot\exp\{-\beta\cdot 0\} + \exp\{-\beta\cdot(-8)\} && (28)\\
&= 2\cdot\exp\{-\beta\cdot(-8)\} + 2\cdot\exp\{-\beta\cdot 8\} + 12\cdot\exp\{-\beta\cdot 0\} && (29)\\
&= 2\cdot\exp\{8\beta\} + 2\cdot\exp\{-8\beta\} + 12 && (30)
\end{aligned}$$

where $\mathbf{s}\in D$ indicates all possible states of the system and β is the inverse temperature T, $\beta = \frac{1}{T}$, given in units of the Boltzmann's constant $[\beta] = 1/k_B$ where $k_B = 1.38064852\cdot 10^{-23}$ m^2 kg s^{-2} K^{-1}.

2. *Expectation value of the energy per spin*

$$\begin{aligned}
\langle\epsilon\rangle &= \left\langle \frac{E(\mathbf{s})}{N} \right\rangle = \frac{1}{N}\langle E(\mathbf{s})\rangle = \frac{1}{N}\sum_{\mathbf{s}\in D} E(\mathbf{s})\cdot p(\mathbf{s};T) && (31)\\
&= \frac{1}{N}\sum_{\mathbf{s}\in D} E(\mathbf{s})\cdot\frac{1}{Z}\exp\{-\beta E(\mathbf{s})\} = \frac{1}{NZ}\sum_{\mathbf{s}\in D} E(\mathbf{s})\cdot\exp\{-\beta E(\mathbf{s})\} && (32)\\
&= \frac{1}{NZ}\Big(-8\cdot\exp\{8\beta\} + 2\cdot 8\cdot\exp\{-8\beta\} - 8\cdot\exp\{8\beta\}\Big) && (33)\\
&= \frac{1}{NZ}\Big(16\cdot\exp\{-8\beta\} - 16\cdot\exp\{8\beta\}\Big) && (34)\\
&\overset{N=4}{=} \frac{4}{Z}\Big(\exp\{-8\beta\} - \exp\{8\beta\}\Big) && (35)
\end{aligned}$$

where $p(\mathbf{s};T)$ is the probability density function for $\mathbf{s}$ as defined in 6

3. *Expectation value of energy per spin squared*

$$
\begin{aligned}
\langle \epsilon^2 \rangle = \left\langle \left(\frac{E(\mathbf{s})}{N} \right)^2 \right\rangle &= \frac{1}{N^2} \left\langle [E(\mathbf{s})]^2 \right\rangle = \frac{1}{N^2} \sum_{\mathbf{s} \in D} [E(\mathbf{s})]^2 \cdot p(\mathbf{s}; T) && (36) \\
&= \frac{1}{N^2} \sum_{\mathbf{s} \in D} [E(\mathbf{s})]^2 \cdot \frac{1}{Z} \exp\{-\beta E(\mathbf{s})\} = \frac{1}{N^2 Z} \sum_{\mathbf{s} \in D} [E(\mathbf{s})]^2 \cdot \exp\{-\beta E(\mathbf{s})\} && (37) \\
&= \frac{1}{N^2 Z} \Big(128 \cdot \exp\{8\beta\} + 128 \cdot \exp\{-8\beta\} \Big) && (38) \\
&\overset{N=4}{=} \frac{8}{Z} \Big(\exp\{8\beta\} + \exp\{-8\beta\} \Big) && (39)
\end{aligned}
$$

4. *Expectation value of the absolute value of the magnetisation per spin*

$$
\begin{aligned}
\langle |m| \rangle = \left\langle \left| \frac{1}{N} M(\mathbf{s}) \right| \right\rangle &= \frac{1}{N} \langle |M(\mathbf{s})| \rangle = \frac{1}{N} \sum_{\mathbf{s} \in D} |M(\mathbf{s})| \cdot p(\mathbf{s}; T) && (40) \\
&= \frac{1}{N} \sum_{\mathbf{s} \in D} |M(\mathbf{s})| \cdot \frac{1}{Z} \exp\{-\beta E(\mathbf{s})\} && (41) \\
&= \frac{1}{NZ} \sum_{\mathbf{s} \in D} |M(\mathbf{s})| \exp\{-\beta E(\mathbf{s})\} && (42) \\
&= \frac{1}{NZ} \Big(4 \cdot \exp\{8\beta\} + 4 \cdot 2 \cdot \exp\{0\} + 4 \cdot 2 \cdot \exp\{0\} + 4 \cdot \exp\{8\beta\} \Big) && (43) \\
&= \frac{1}{NZ} \Big(8 \cdot \exp\{8\beta\} + 16 \Big) && (44) \\
&\overset{N=4}{=} \frac{1}{Z} \Big(2 \cdot \exp\{8\beta\} + 4 \Big) && (45) \\
&= \frac{2}{Z} \Big(\exp\{8\beta\} + 2 \Big) && (46)
\end{aligned}
$$

5. *Expectation value of the magnetisation per spin squared*

$$
\begin{aligned}
\langle m^2 \rangle = \left\langle \left(\frac{1}{N} M(\mathbf{s}) \right)^2 \right\rangle &= \frac{1}{N^2} \sum_{\mathbf{s} \in D} [M(\mathbf{s})]^2 \cdot p(\mathbf{s}; T) && (47) \\
&= \frac{1}{N^2} \sum_{\mathbf{s} \in D} [M(\mathbf{s})]^2 \cdot \frac{1}{Z} \exp\{-\beta E(\mathbf{s})\} && (48) \\
&= \frac{1}{N^2 Z} \sum_{\mathbf{s} \in D} [M(\mathbf{s})]^2 \exp\{-\beta E(\mathbf{s})\} && (49) \\
&= \frac{1}{N^2 Z} \Big(16 \cdot \exp\{8\beta\} + 4 \cdot 4 \exp\{0\} + 4 \cdot 4 \cdot \exp\{0\} + 16 \cdot \exp\{8\beta\} \Big) && (50) \\
&= \frac{1}{N^2 Z} \Big(32 \cdot \exp\{8\beta\} + 32 \Big) && (51) \\
&\overset{N=4}{=} \frac{1}{Z} \Big(2 \cdot \exp\{8\beta\} + 2 \Big) && (52)
\end{aligned}
$$

6. Specific heat capacity (normalised to number of spins)

$$
\begin{aligned}
C_v &= \frac{1}{k_B T^2}\frac{1}{N}\Big(\left\langle [E(\mathbf{s})]^2\right\rangle - \left\langle E(\mathbf{s})\right\rangle^2\Big) && (53)\\
&= \frac{1}{k_B T^2}\frac{1}{N}\Big(\left\langle N^2\epsilon^2\right\rangle - \left\langle N\epsilon\right\rangle^2\Big) && (54)\\
&= \frac{1}{k_B T^2}\frac{1}{N}\Big(N^2\left\langle \epsilon^2\right\rangle - N^2\left\langle \epsilon\right\rangle^2\Big) && (55)\\
&= \frac{N}{k_B T^2}\Big(\left\langle \epsilon^2\right\rangle - \left\langle \epsilon\right\rangle^2\Big) && (56)\\
&= \frac{N}{k_B T^2}\Big(\frac{8}{Z}\Big(\exp\{-8\beta\} + \exp\{8\beta\}\Big) - \Big(\frac{4}{Z}\Big(\exp\{-8\beta\} - \exp\{8\beta\}\Big)\Big)^2\Big) && (57)\\
&= \frac{N}{k_B T^2}\left(\frac{8}{Z}\Big(\exp\{-8\beta\} + \exp\{8\beta\}\Big) - \frac{16}{Z^2}\Big(\exp\{-16\beta\} - 2 + \exp\{16\beta\}\Big)\right) && (58)\\
&= \frac{8N}{k_B T^2 Z}\left(\exp\{-8\beta\} + \exp\{8\beta\} - \frac{2}{Z}\Big(\exp\{-16\beta\} + \exp\{16\beta\} - 2\Big)\right) && (59)\\
&\overset{N=4}{=} \frac{32}{k_B T^2 Z}\left(\exp\{-8\beta\} + \exp\{8\beta\} - \frac{2}{Z}\Big(\exp\{-16\beta\} + \exp\{16\beta\} - 2\Big)\right) && (60)
\end{aligned}
$$

with $E(\mathbf{s}) = N\epsilon$.

7. The susceptibility (normalised to number of spins)

$$
\begin{aligned}
\chi &= \frac{1}{k_B T}\frac{1}{N}\Big(\left\langle [M(\mathbf{s})]^2\right\rangle - \left\langle |M(\mathbf{s})|\right\rangle^2\Big) && (61)\\
&= \frac{1}{k_B T}\frac{1}{N}\Big(\left\langle [M(\mathbf{s})]^2\right\rangle - \left\langle |M(\mathbf{s})|\right\rangle^2\Big) && (62)\\
&= \frac{1}{k_B T}\frac{1}{N}\Big(\left\langle N^2 m^2\right\rangle - \left\langle |Nm|\right\rangle^2\Big) && (63)\\
&= \frac{1}{k_B T}\frac{1}{N}\Big(N^2\left\langle m^2\right\rangle - N^2\left\langle |m|\right\rangle^2\Big) && (64)\\
&= \frac{N}{k_B T}\left(\frac{1}{Z}\Big(2\cdot\exp\{8\beta\} + 2\Big) - \Big(\frac{1}{Z}\Big(2\cdot\exp\{8\beta\} + 4\Big)\Big)^2\right) && (65)\\
&= \frac{N}{k_B T}\left(\frac{1}{Z}\Big(2\cdot\exp\{8\beta\} + 2\Big) - \frac{4}{Z^2}\Big(\exp\{16\beta\} + 4\exp\{8\beta\} + 4\Big)\right) && (66)\\
&= \frac{N}{k_B T}\frac{2}{Z}\left(\exp\{8\beta\} + 1 - \frac{2}{Z}\Big(\exp\{16\beta\} + 4\exp\{8\beta\} + 4\Big)\right) && (67)\\
&\overset{N=4}{=} \frac{8}{k_B T Z}\left(\exp\{8\beta\} + 1 - \frac{2}{Z}\Big(\exp\{16\beta\} + 4\exp\{8\beta\} + 4\Big)\right) && (68)
\end{aligned}
$$

with $M(\mathbf{s}) = Nm$.

C. Substituting if-statements with modulo

As we want the lattice to have boundary conditions where the neighbouring spin of a particle is dependent on the spin on the opposite side of the lattice, to simulate a lattice that wraps around, it is tempting to use if statements.

For example, if we wish to calculate the neighbouring spin of a particle with index (0,x) in the matrix we need the spin of particle (L-1,x). It is tempting to make an if statement to fetch the spin in these cases, however we can use modulo to avoid using if statement, and make the code more efficient.

Indexing with with $(i + L)\%L)$ where i literates from 0 to L, the first index will take the values from 0 to L, but when $i = L$, using modulo will make the index 0, as $(L + L)\%L = 0$. Effectively this will make the index wrap around without the use of any if statements.

D. Energy shift for flipping the spin of a single particle

We know that the total energy of the system is given by:

$$E(s) = -J \sum_{<kl>}^{N} s_k s_l \tag{69}$$

where $< kl >$ means the sum over all neighbouring pairs. We can calculate the shift in the total energy of the system in the following way:

$$\begin{aligned} \Delta E &= E_{after} - E_{before} && (70) \\ &= (-J \sum_{<kl>}^{N} s_k s_l)_{after} - (-J \sum_{<kl>}^{N} s_k s_l)_{before} && (71) \\ &= -J((\sum_{<kl>}^{N} s_k s_l)_{before} - (\sum_{<kl>}^{N} s_k s_l)_{after}) && (72) \end{aligned}$$

When flipping the spin of one particle we can view the sum in equation 69 $\sum s_k s_l$ as a sum where s_k is the spin of the particle we want to flip, and s_l is the respective spin of its four neighbouring particles. If we assume s_k to have a positive spin in the initial state, and a negative spin after being flipped, we can calculate the shift in the total energy of the system by looking at equation 72. This shift in energy is shown in table V D. It is worth noting that only the region where the flip occurred will change the system as all the other sums cancel.

	$\sum s_k s_l$		
N_+	initial state	flipped state	ΔE
0	4	-4	-8J
1	2	-2	-4J
2	0	0	0J
3	-2	2	4J
4	-4	4	8J

TABLE V. This table shows the shift in the total energy of the system, when a particle with positive spin is flipped to have negative spin. Column N_+ shows the amount of neighbouring particles with positive spin. Column $\sum s_k s_l$ shows the sum over the particles spins where s_k is the spin of the centre particle, and s_l is the respective spin of its neighbours. As we assume the particle to have an initial positive spin, it is split into two sub columns, one with the initial postitve spin state, and one with the 'after' the paricle has been flipped to have a negative spin. Column ΔE shows the total energy shift in the system.

Looking at table V D, we see that the total energy shift ΔE yields five different unique values. These values are however symmetric around zero. If we were to assume the the initial spin of the particle was negative, and flipped to positive, we would gain the same five values for the shift in energy ΔE, but in opposite order. Flipping the spin of the particle will just flip the sign of each sum calculated in table V D, thus also the shift in energy ΔE. The shift in the total energy ΔE when flipping the spin of a singe particle can therefore only take one of five possible values, as displayed in table V D.

We are going to frequently use the Boltzmann factor $e^{-\beta\Delta E}$, where β is dependent on a temperature T and the Boltzmann constant. As we know that ΔE can only have 5 possible values, the Boltzmann factor will also only have 5 possible values for a given temperature. Therefore we can calculate these 5 different values for the Boltzmann constant once and put them in a list for a given temperature, and using it as a lookup table. This way we avoid having to repeatedly call the expensive exponential function to calculate the Boltzmann factor.