Simulating the phase transition of ferromagnets using the Markov chain Monte Carlo method

Aline Rangøy Brunvoll, Even Tobias Eriksen, and Vilde Ung
(Dated: November 22, 2022)

We studied the critical behaviour of the two-dimensional ferromagnetic Ising model using finite-size scaling methods. We applied Markov chain Monte Carlo (MCMC) methods and the Metropolis algorithm to sample the spin configurations. In order to optimise the runtime of our program, we parallelised our code with `OpenMP`. As a benchmark, we used the analytic solution of a 2×2 lattice. We studied the burn-in period of a 20×20 lattice, finding that > 10.000 MC cycles were needed for the system to equilibriate. Additionally, we estimated the probability distribution of the average energy per spin, finding that the variance increased close to the critical temperature. Finally, we simulated four different lattices across a range of temperatures, and estimated the critical temperature of the infinite lattice to be $T_c = 2.260$, which was 96% accurate relative to Onsager's analytic solution.

GitHub repository: https://github.com/ungvilde/FYS4150/tree/main/Project%204

I. INTRODUCTION

The Ising model is a simple, mathematical model that has been widely studied in statistical physics. Typically, the model variables represent magnetic dipole moments, known as *spins*, which can be used to study the phase transitions of *ferromagnets*. Ferromagnetism is a property that allows a magnetic material to align neighbouring spins, such that their individual magnetic fields line up and amplify one another. Below the *critical temperature*, a ferromagnet with no net magnetization undergoes a phase transition, where the magnetic fields of the atoms organize in one common direction, so that the net magnetization of the material is no longer zero. In addition to studying critical phenomena in ferromagnets, the Ising model is used for a wide range of applications, such as modelling neuronal synchrony [1] and the binding of oxygen to hemoglobin [2].

Solving the Ising model analytically is challenging, as the number of possible states scales exponentially with the size of the lattice. In this article, we apply Monte Carlo methods to numerically simulate phase-transitions in the two-dimensional Ising model. Monte Carlo methods are a central part of computational physics, in particular when simulating systems with a large number of possible states. With the Metropolis algorithm, we can sample this vast number of possible states by importance, in order to more efficiently estimate the observables of the system. Using finite-size scaling relations, we can apply the numerical results of finite lattices to estimate the critical temperature of the infinite lattice, which Lars Onsager solved analytically in 1944 [3].

In Section II we thoroughly introduce the concepts, equations, and numerical methods applied. Section III presents the relevant results from our simulations, and Section IV provides an analysis of our findings and a discussion of the physical implications. Finally, our experiment and main results will be summarized briefly in Section V.

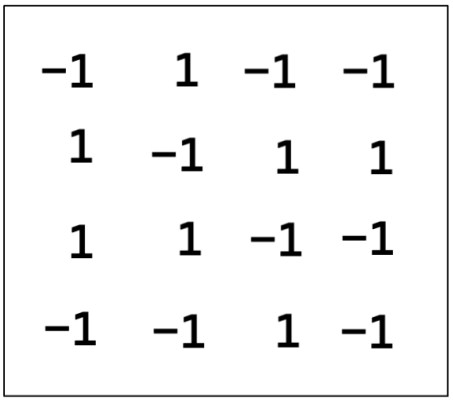

Figure 1: An example of the layout of different spin states $s_i \in \{-1, 1\}$ for each grid point in a randomly initiated square lattice with length $L = 4$.

II. METHODS

A. The Ising model

The Ising model is made up of a lattice geometry with discrete variables s_i (called spins) at each site, where every spin interacts with its nearest neighbours. Here, we apply a two-dimensional square lattice of length L and a total number of $N = L^2$ spins. The individual spins can be in one of two possible states $s_i \in \{-1, +1\}$, representing the magnetic dipole moment of an atom. We denote the spin configuration of the lattice as $\mathbf{s} = (s_1, s_2, \ldots, s_N)$. See Figure 1 for a visualisation of the lattice. In our model, we apply periodic boundary conditions, where each spin will have exactly four neighbouring spins, effectively modelling the lattice as a torus.

The total energy state of the system is defined by the

Hamiltonian, given by

$$E(\mathbf{s}) = -J \sum_{\langle kl \rangle}^{N} s_k s_l. \tag{1}$$

Here, J is the *coupling constant*, which defines the strength of the interaction between the neighbouring spin pairs. We will apply $J > 0$, in which case the energy is minimized when neighboring spins are aligned, making the interactions "ferromagnetic". Note that we consider the case where there is no external magnetic field, which simplifies the Hamiltonian function. The $\langle kl \rangle$-term means that we sum over neighbouring pairs of spins only. In a lattice with N spins, there will be $2N$ such pairs to sum over.

1. Probabilities and observables

We are interested in computing the expected energy and magnetization of the system, where the total magnetization is given by

$$M(\mathbf{s}) = \sum_{i=1}^{N} s_i, \tag{2}$$

which is the sum over all spins in the lattice. In order to compare the energy and magnetization of lattices of different sizes, we use the average energy and average magnetization per spin, given by

$$\epsilon(\mathbf{s}) = \frac{E(\mathbf{s})}{N} \tag{3}$$

$$m(\mathbf{s}) = \frac{M(\mathbf{s})}{N}, \tag{4}$$

where N is the total number of spin states.

The probability of the system being in state $\mathbf{s}$ for a given temperature T is determined by the Boltzmann distribution, given by

$$p(\mathbf{s}; T) = \frac{1}{Z} e^{-\beta E(\mathbf{s})}, \tag{5}$$

where $\beta = 1/k_B T$ is the *inverse temperature*, k_B is the Boltzmann constant, and Z is the *partition function*. The partition function is a normalization constant given by

$$Z = \sum_{\text{all possible } \mathbf{s}} e^{-\beta E(\mathbf{s})}. \tag{6}$$

The Boltzmann distribution can be applied to compute the expectation values of the total energy, the total magnetization, energy per spin and magnetization per spin,

Parameter	Unit
$[E]$	J
$[T]$	J/k_B
$[\epsilon]$	J
$[C_V]$	k_B/J^2
$[\chi]$	$1/J$

Table I: Units of the parameters used in this study. J is the coupling constant and has units of energy. k_B is Boltzmann's constant.

so that we can write

$$\langle \epsilon \rangle = \sum_{\mathbf{s}} \epsilon(\mathbf{s}) p(\mathbf{s}; T)$$

$$\langle E \rangle = \sum_{\mathbf{s}} E(\mathbf{s}) p(\mathbf{s}; T)$$

$$\langle E^2 \rangle = \sum_{\mathbf{s}} E(\mathbf{s})^2 p(\mathbf{s}; T)$$

$$\langle |m| \rangle = \sum_{\mathbf{s}} |m(\mathbf{s})| p(\mathbf{s}; T)$$

$$\langle |M| \rangle = \sum_{\mathbf{s}} |M(\mathbf{s})| p(\mathbf{s}; T)$$

$$\langle M^2 \rangle = \sum_{\mathbf{s}} M(\mathbf{s})^2 p(\mathbf{s}; T).$$

These relations can be further applied to compute the *heat capacity* C_V and *magnetic susceptibility* χ. These values can be calculated using

$$C_V = \frac{1}{N} \frac{1}{k_B T^2} (\langle E^2 \rangle - \langle E \rangle^2) \tag{7}$$

$$\chi = \frac{1}{N} \frac{1}{k_B T} (\langle M^2 \rangle - \langle |M| \rangle^2). \tag{8}$$

The units of the observables relevant to this study are given in Table I.

2. An analytic solution

In order to verify our numerical solutions we will compare it to an analytic solution for a 2×2 lattice. Computing analytic solutions for larger lattice models becomes increasingly difficult, as the number of possible states grows exponentially, so we limit ourselves to $L = 2$. In this case, there are $2^4 = 16$ possible states. Table II shows the total energy, total magnetization, and degeneracy for every number of spins in state $+1$. The total energy and total magnetization are calculated using (1) and (2). The degeneracy counts the number of possible states that produce the given energy and magnetisation values. The partition function can be solved as

$$Z = 4 \left(\cosh \beta 8J + 3 \right).$$

+1 spins	E(**s**) [J]	M(**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 II: All states of a 2×2 lattice, where we show the possible number of spins in state +1, total energy, total magnetization and degeneracy of the system.

The expected energy per spin is solved as

$$\langle \epsilon \rangle = -\frac{2J \sinh \beta 8J}{\cosh \beta 8J + 3}.$$

The expected value of the squared energy per spin is solved as

$$\langle \epsilon^2 \rangle = \frac{8J^2 \cosh \beta 8J}{\cosh \beta 8J + 3}.$$

The the expected value of absolute magnetization per spin is solved as

$$\langle |m| \rangle = \frac{\left(e^{\beta 8J} + 2\right)}{2\left(\cosh \beta 8J + 3\right)}.$$

The expected value of the squared magnetization per spin is solved as

$$\langle m^2 \rangle = \frac{\left(e^{\beta 8J} + 1\right)}{2\left(\cosh \beta 8J + 3\right)}.$$

The specific heat capacity (normalized to the number of spins) is solved as

$$C_V = \frac{64J^2}{Nk_BT^2} \frac{3\cosh \beta 8J + 1}{(\cosh \beta 8J + 3)^2},$$

where the expected values for the total energy is solved as

$$\langle E^2 \rangle = \frac{64J^2 \cosh \beta 8J}{\cosh \beta 8J + 3}$$
$$\langle E \rangle^2 = \frac{64J^2 \sinh^2 \beta 8J}{(\cosh \beta 8J + 3)^2}.$$

The susceptibility (normalized for number of spins) is given as

$$\chi = \frac{4}{Nk_BT^2} \left(\frac{e^{-\beta 8J} + 3e^{\beta 8J} + 3}{(\cosh \beta 8J + 3)^2} \right)$$

where the expected value of the magnetization is given as

$$\langle M^2 \rangle = \frac{8\left(e^{\beta 8J} + 1\right)}{\cosh \beta 8J + 3}$$
$$\langle |M| \rangle = \frac{4\left(e^{\beta 8J} + 2\right)^2}{(\cosh \beta 8J + 3)^2}$$

The full derivations of these results can be found in Appendix A.

B. Phase transitions and critical exponents

The two-dimensional Ising model will undergo a phase transition at a critical temperature T_c, which means that below T_c the model spontaneously magnetises [4, p. 430]. Lars Onsager found analytically that the critical temperature for the two-dimensional infinite lattice is

$$T_c(L = \infty) = \frac{2}{\ln\left(1 + \sqrt{2}\right)} J/k_B \approx 2.269\, J/k_B, \qquad (9)$$

where J is the coupling constant and k_B is the Boltzmann constant. At temperatures above T_c the spins are randomly oriented and there is no net magnetisation. As the temperature decreases, the spins become more and more correlated, creating "patches" of spins pointing in the same direction. Still, without a magnetic field present, the net magnetisation is zero. When the temperature is lowered below T_c, the lattice breaks symmetry with respect to positive and negative spins, causing it to have non-zero magnetisation [5, p. 30]. The correlation length ξ, i.e. the distance with which the spins correlate and point in the same direction, diverges near the critical point. Close to the critical temperature, the relationship between the temperature and the correlation length can be expressed as a power law

$$\xi(T) \sim |T_c - T|^{-\nu},$$

with $\nu = 1$.

Near the critical temperature, many physical quantities can be described in terms of power laws. The *critical exponent* is the exponent that characterises the power law. For example, we have

$$C_V \sim |T_c - T|^{-\alpha}$$
$$\chi \sim |T_c - T|^{-\gamma},$$

with critical exponents $\gamma = -7/4$ and $\alpha = 0$. For the infinite lattice ($L = \infty$), the heat capacity, correlation length and susceptibility diverge at T_c [4, p. 431]. This is not the case with finite sized lattices, but we expect to observe a sharp increase in these quantities around the critical temperature. The maximum becomes sharper with larger lattices.

With numerical experiments we can only simulate finite-sized lattices, where the correlation length at most is $\xi = L$. Periodic boundary conditions can somewhat improve this limitation, by maximizing the possible number of interactions in the model. However, through finite-sized scaling relations [4, p. 432] we can relate the behaviour of finite lattices with the infinite-lattice case. The critical temperature can then be found through the relation

$$T_c(L) - T_c(L = \infty) = aL^{-1}. \qquad (10)$$

This motivates a linear regression model based on estimates of $T_c(L)$ from different lattice sizes, where $T_c(L = \infty)$ is the intercept of the regression fit.

C. Markov Chain Monte Carlo methods

With Markov Chain Monte Carlo (MCMC) methods we sample states from a probability distribution function. This allows us to estimate ensemble averages and their probability distributions where analytical results are unavailable or difficult to obtain. In the case of the Ising model, MCMC methods are desirable due to the exponentially increasing number of possible states for large lattices.

A Markov chain is a stochastic process evolving in time, where each new state can occur with some probability that is independent of the history of the process. The goal is that, as time goes on, the Markov process eventually reaches a stationary distribution. In the case of the Ising model, this is

$$P_{\mathbf{s}}^{\text{eq}} = \frac{e^{-\beta E(\mathbf{s})}}{Z},$$

where $\mathbf{s}$ is a given spin configuration, and $P_{\mathbf{s}}^{\text{eq}}$ is the stationary probability of that configuration. In order to achieve this, there are two important conditions that need to be met. The first is *ergodicity*, which requires that the Markov chain, in theory, can visit all possible states of the system in finite time. The second requirement is *detailed balance*, which essentially puts a constraint on the transition probabilities between states so that the process is reversible. In mathematical terms, detailed balance requires

$$P_{\mathbf{s}}^{\text{eq}} w_{\mathbf{s}\to\mathbf{l}} = P_{\mathbf{l}}^{\text{eq}} w_{\mathbf{l}\to\mathbf{s}},$$

where $w_{\mathbf{s}\to\mathbf{l}}$ denotes the transition probability from state $\mathbf{s}$ to state $\mathbf{l}$. This implies that

$$\frac{w_{\mathbf{l}\to\mathbf{s}}}{w_{\mathbf{s}\to\mathbf{l}}} = \frac{P_{\mathbf{s}}^{\text{eq}}}{P_{\mathbf{l}}^{\text{eq}}} = e^{-\beta(E(\mathbf{s})-E(\mathbf{l}))} = e^{-\beta\Delta E},$$

where ΔE is the energy difference between state $\mathbf{s}$ and $\mathbf{l}$ [5, p. 23].

1. *The Metropolis algorithm*

The Metropolis algorithm selects transition probabilities in a way that satisfies the given constrains. Meaning, we do not know the true transition probabilities $w_{\mathbf{s}\to\mathbf{l}}$, but can choose an acceptance function that satisfies the given constraints. Assuming that the transition probability $A(\mathbf{l}\to\mathbf{s})$ that we choose should be a function of $e^{-\beta\Delta E}$, we can select

$$A(\mathbf{l}\to\mathbf{s}) = \min(1, e^{-\beta\Delta E}). \tag{11}$$

This satisfies detailed balance and ensures that the transition probability is in the range $0 \leq A(\mathbf{l}\to\mathbf{s}) \leq 1$. Additionally, with (11) we maximise the number of accepted transitions, which makes the algorithm more effective. In practise, we will do a spin flip whenever the transition lowers the energy of the system, while also allowing there to be a transition to higher energy states with probability $e^{-\beta\Delta E}$.

D. Implementation of MCMC methods for the Ising model

The lattice we use for our simulations is represented by an $L \times L$-matrix with N spins valued to either -1 or 1. To account for periodic boundary conditions when computing the Hamiltonian given in Equation (1), we apply the modulo operator to the indices of the spins in the lattice. The lattice is initiated to a desired configuration, either as an ordered lattice where all spins point in the same direction, or as a random configuration.

One Monte Carlo (MC) cycle is the process of attempting N randomly selected spin flips, where we apply the Metropolis algorithm when evaluating if we accept the spin flip or not. When attempting a spin flip, we compute the resulting local energy change of the flip ΔE. If $\Delta E \leq 0$, we do the flip. If $\Delta E > 0$, we choose a random number $r \sim \mathcal{U}(0,1)$ and do the flip if $r < e^{-\beta\Delta E}$. In this way, we do not need to call the `exp`-function every time we attempt to do a flip, which decreases the computational cost of our simulation. Whenever we do a spin flip, the energy and magnetisation values of the lattice are updated by the local change that follows from doing the flip. In this way, we do not need to compute the energy and magnetisation of the full lattice more than once, i.e. from the initial configuration.

We do a total number of n MC cycles, but we will only store the energy and magnetisation values after $n_0 < n$ cycles. This is referred to as a *burn-in* period. We risk initiating the lattice in an improbable configuration, which means that it might take a long time before we sample states that are representative of the stationary distribution. This is because MCMC samples are autocorrelated, making the selected samples highly dependent on the initial state. To get more accurate estimates of the ensemble averages and their distributions, we discard the first n_0 samples. This also motivates a choice of n that is as large as is computationally feasible, in order to get the most accurate results. A further note on this, is that when we simulate the lattice near the critical temperature T_c, the increase in correlation length will cause the samples to be more autocorrelated, making it harder to draw statistically independent samples, particularly with larger lattices [6]. This is known as *critical slowing down*. Although there are algorithms that can alleviate this problem, we will not explore them further in this study.

Finally, we note that $\langle\epsilon\rangle$ and $\langle|m|\rangle$ are computed as the sample average of the $n - n_0$ stored values, and that the heat capacity and susceptibility are computed using the sample variance of the energy and magnetisation, re-

spectively. Specifically,

$$\langle \epsilon \rangle = \frac{1}{n-n_0} \sum_{k=n_0}^{n} \epsilon_k$$

$$\langle |m| \rangle = \frac{1}{n-n_0} \sum_{k=n_0}^{n} |m_k|$$

$$C_V = \frac{1}{k_B T^2} \frac{1}{n-n_0} \sum_{k=n_0}^{n} (\epsilon_k - \langle \epsilon \rangle)^2$$

$$\chi = \frac{1}{k_B T} \frac{1}{n-n_0} \sum_{k=n_0}^{n} (|m_k| - \langle |m| \rangle)^2,$$

where $\epsilon_k = E_k/N$ is the energy per spin of sample k and $m_k = M_k/N$ is the magnetisation per spin of sample k. The algorithm we used in our simulations is presented in Algorithm 1.

Algorithm 1 One MC cycle with the Metropolis algorithm

```
L ← Lattice configuration
N ← Number of spins in L
E ← Energy of lattice
M ← Magnetisation of lattice
for all k = 1, ..., N do
    i, j ← Choose random indices
    E_flip ← Compute Hamiltonian at L(i, j)
    E_no flip ← −E_flip
    ΔE ← E_flip − E_no flip        ▷ Energy change when spin is flipped.
    u ← U(0, 1)                    ▷ Random number between 0 and 1
    if ΔE ≤ 0 then
        L(i, j) ← −L(i, j)         ▷ Do the flip
        E ← E + ΔE
        M ← M + 2L(i, j)
    else if u ≤ e^{−βΔE} then
        L(i, j) ← −L(i, j)         ▷ Do the flip
        E ← E + ΔE
        M ← M + 2L(i, j)
    else
        Reject the flip
```

In order to reduce the runtime of our simulations, thus allowing us to run more MC cycles and giving us a better numerical result, we parallelize our code using `OpenMP` [7]. `OpenMP` is an application programming interface (API) which assumes an underlying shared-memory architecture. It enables for multiple threads to access the same memory and execute multiple tasks at the same time. This allows us to run Monte Carlo simulations for different temperatures simultaneously in several different threads. If we compare the computation time of our program when we compile it for several threads with the time it takes for only one processor to get that same solution, we get what is called the *speed-up factor*. This can be used to see just how much the computational efficiency is increased by parallelizing with `OpenMP`. The speed-up factor can be expressed mathematically as

$$\text{Speed-up factor} = \frac{\text{Old execution time}}{\text{New execution time}}.$$

III. RESULTS

A. Comparing the analytic and numeric solutions

We first validated our model by comparing the results of our numeric solution of the 2×2-lattice with the analytic solution found in Section II A 2. We applied the temperature $T = 1.0\ J/k_B$, and ran the simulation for $n \in \{5000, 50.000, 500.000\}$ Monte Carlo cycles. The lattice was initialised in a random configuration in all cases. We computed the estimates of $\langle \epsilon \rangle$, $\langle |m| \rangle$, C_V, and χ using all the MCMC samples, i.e. we did not implement burn-in at this stage. Similarly, we used the analytic equations for the 2×2-lattice to compute the equivalent values analytically. The results of these computations can be seen in Table III.

B. Exploring the burn-in period

We then studied the burn-in period of a lattice with $L = 20$. We looked at the burn-in at $T = 1.0\, J/k_B$ and $T = 2.4\, J/k_B$, and initiated the lattice in both a random and an ordered configuration. We ran 50.000 MC cycles, and computed the expected energy and magnetisation per spin as a function of MC cycles. The results for $T = 1.0\, J/k_B$ can be seen in Figure 2 and 3. The results for $T = 2.4\, J/k_B$ can be seen in Figure 4 and 5.

C. Estimating the probability distribution

We looked at how the energy per spin was distributed over the different samples, in order to estimate the probability distribution of ϵ. We first had a burn-in period of $n_0 = 10.000$, followed by sampling and storing 100.000 ϵ-values. Again, we did this for $T = 1.0\, J/k_B$ and $T = 2.4\, J/k_B$. In simulating at $T = 1.0\, J/k_B$, we initiated the lattice in an ordered configuration, as this is a more representative state of the lattice at this temperature. At $T = 2.4\, J/k_B$ the lattice was randomly initiated. The resulting normalised histograms for $T = 1.0\, J/k_B$ and $T = 2.4\, J/k_B$ can be seen in Figure 7 and 6, respectively. For $T = 1.0\, J/k_B$ we computed the sample variance $\text{Var}(\epsilon) = 6.0 \cdot 10^{-5}$ and sample mean $\langle \epsilon \rangle = -1.997$. For $T = 2.4\, J/k_B$ we computed the sample variance $\text{Var}(\epsilon) = 1.95 \cdot 10^{-2}$ and sample mean $\langle \epsilon \rangle = -1.236$.

D. Reducing runtime with parallelisation

We looked at how parallelising the code affected the runtime. We tested running a simulation with $L = 20$ for

MC cycles	ϵ	Rel. error	$\lvert m\rvert$	Rel. error	C_V	Rel. error	χ	Rel. error
5.000	-1.99840	$< 1\%$	0.99950	1%	0.01279	60%	0.00140	65%
50.000	-1.99648	$< 1\%$	0.99889	$< 1\%$	0.02811	12%	0.00314	21%
500.000	-1.99586	$< 1\%$	0.99862	$< 1\%$	0.03308	3%	0.00412	2%
Analytic	-1.99598	–	0.99799	–	0.03208	–	0.00401	–

Table III: Comparing numerical and analytical quantities for the 2×2 lattice with $T = 1.0\, J/k_B$. The numeric estimates are computed from 5000, 50.000 and 500.000 MCMC samples. For each observable, we have computed the relative error percentage w.r.t. the analytic solution.

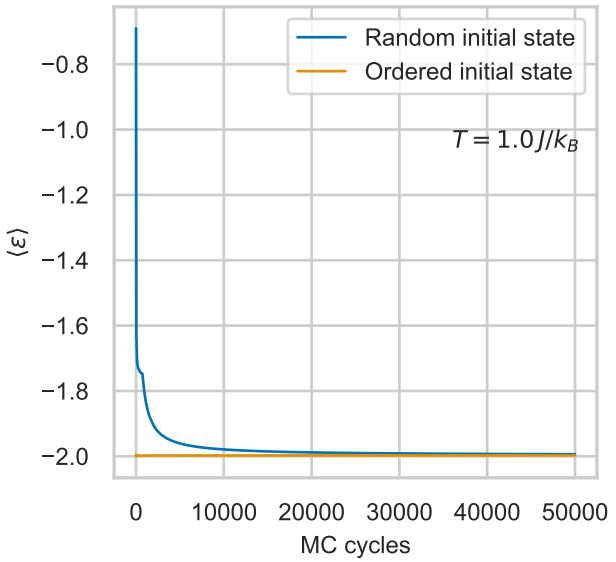

Figure 2: $\langle \epsilon \rangle$ as a function of MC cycles for a 20×20-lattice with temperature $T = 1.0\ J/k_B$. The results from grids with both random and ordered initialisation have been plotted.

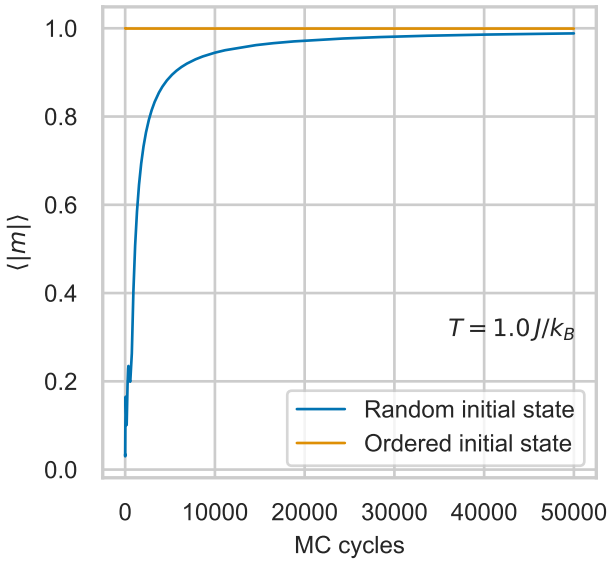

Figure 3: $\langle |m| \rangle$ as a function of MC cycles for a 20×20-lattice with temperature $T = 1.0\ J/k_B$. The results from grids with both random and ordered initialisation have been plotted.

50.000 MC cycles, over 20 different temperatures. Without any parallelisation, this took approximately 13 seconds. We then parallelised the simulation over different temperatures, testing how the number of threads affected the runtime. Since we ran the experiment on a computer with a maximum of 8 available threads, we only ran the software for up to 9 threads. Note that running the simulation with only a single temperature took approximately 0.757 s. Also note that we compiled with the optimisation flag `-O3` when testing the runtime performance. The results can be seen in Table IV.

E. Exploring phase transitions

We simulated the Ising model for $L \in \{40, 60, 80, 100\}$ at different temperatures, and used the sampled values to estimate the critical temperature $T_c(L)$. We looked at 40 evenly spaced temperatures in the range $T \in [2.1, 2.4]\, J/k_B$. The lattices were initiated in a ran-

Num. threads	Runtime [s]	Speed-up
1	13.881	–
2	9.064	1.5
3	6.231	2.2
4	5.310	2.6
5	5.014	2.8
6	4.515	3.1
7	4.733	2.9
8	4.373	3.2
9	4.343	3.2

Table IV: Runtime of $N = 50\,000$ Monte Carlo cycles for 20 temperatures $T \in [1, 4]\, J/k_B$, using a 20×20-grid with and without parallelization. We compiled our code with the optimisation flag `-O3`. The reported runtime is the average time of three runs, and the speed-up factor is calculated with II D.

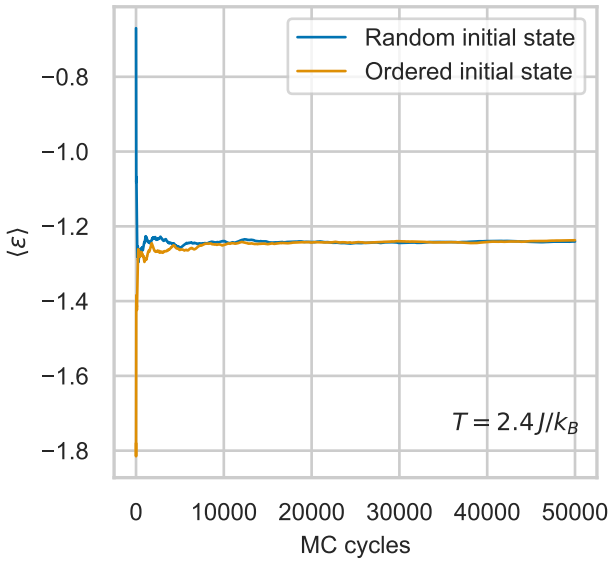

Figure 4: $\langle \epsilon \rangle$ as a function of MC cycles for a 20×20-lattice with temperature $T = 2.4\ J/k_B$. The results from grids with both random and ordered initialisation have been plotted.

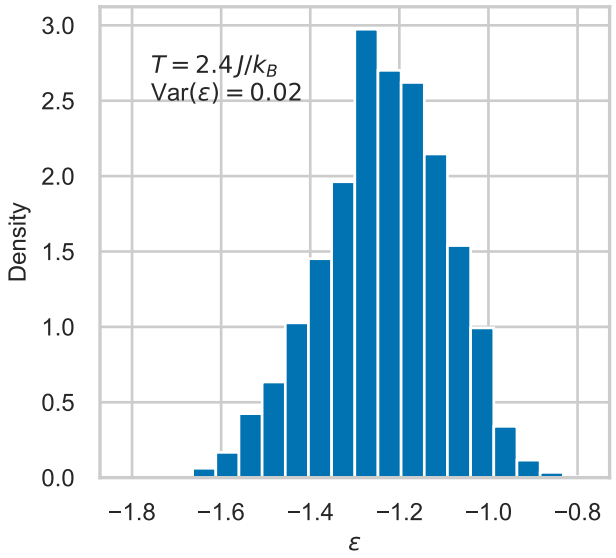

Figure 6: Histogram of the ϵ-values of a 20×20 lattice at $T = 2.4\, J/k_B$. The values were sampled from 100.000 MC cycles, of which the first 10.000 were discarded.

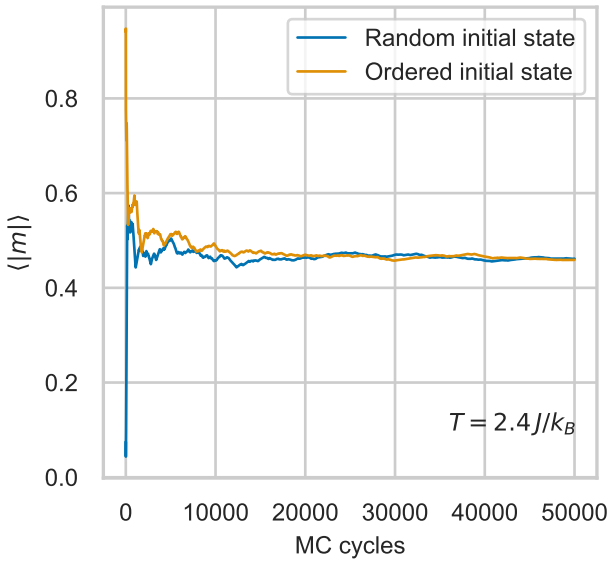

Figure 5: $\langle |m| \rangle$ as a function of MC cycles. The results from using both random and ordered initialisation have been plotted. Here we used $T = 2.4\, J/k_B$.

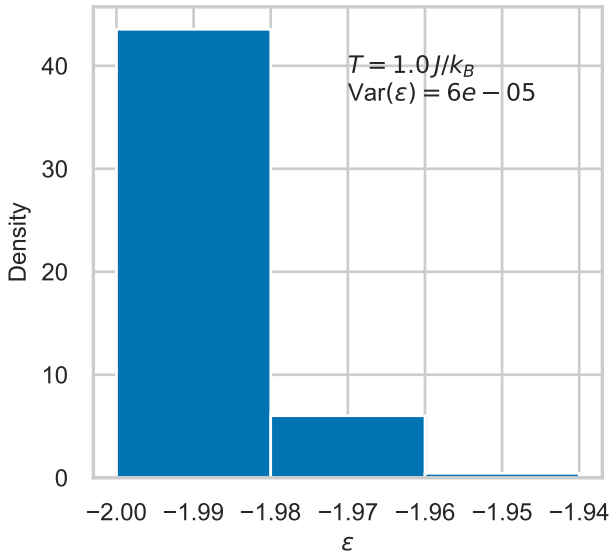

Figure 7: Histogram of the ϵ-values of a 20×20 lattice at $T = 1.0\, J/k_B$. The values were sampled from 100.000 MC cycles, of which the first 10.000 were discarded.

dom configuration. We ran 10^6 MC cycles, of which the first 20.000 cycles made up the burn-in period. We parallelised the simulations over the different temperatures to speed up computations. From the collected samples we computed $\langle \epsilon \rangle$, $\langle |m| \rangle$, C_V, and χ for each temperature and each lattice size. The estimated $\langle \epsilon \rangle$-values as a function of temperature can be seen in Figure 8. In Figure 9 we have plotted $\langle |m| \rangle$ as a function of temperature. Finally, the heat capacity C_V and the susceptibility χ as functions of temperature can be seen in Figure 10 and 11, respectively. We used the temperature that corresponded to the maximum χ-value for each $L \in \{40, 60, 80, 100\}$ as an estimate for $T_c(L)$. The results can be seen in Table V.

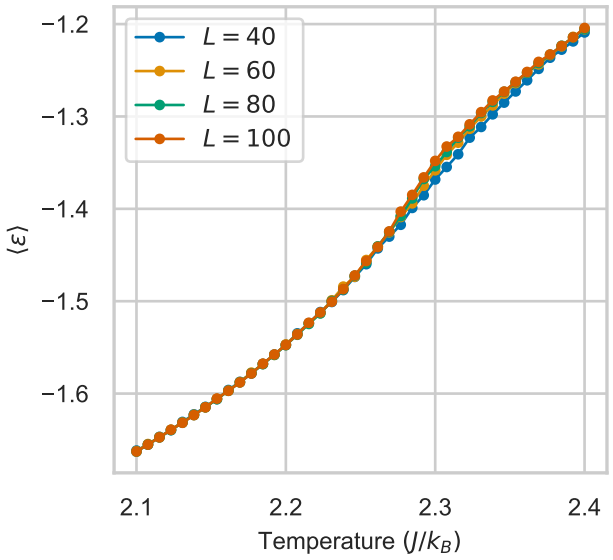

Figure 8: The expected energy as a function of temperature simulated for lattices with lengths $L \in \{40, 60, 80, 100\}$ for 10^6 MC cycles.

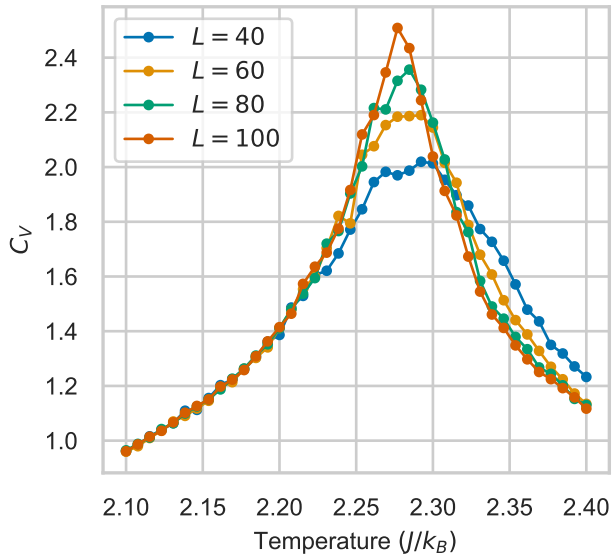

Figure 10: Heat capacity as a function of temperature simulated for lattices with lengths $L \in \{40, 60, 80, 100\}$ for 10^6 MC cycles.

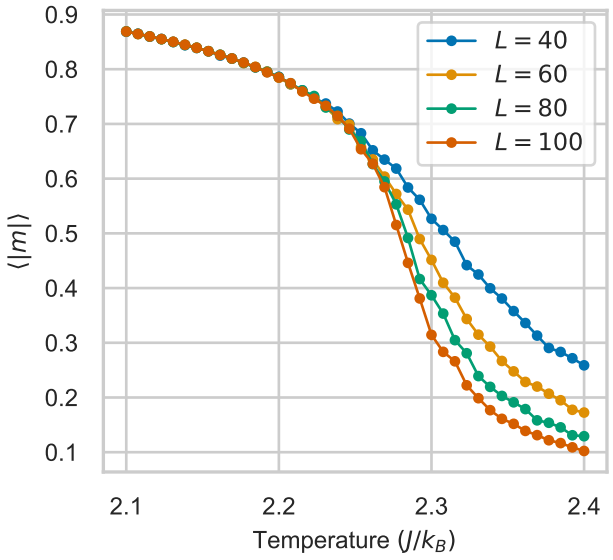

Figure 9: The expected magnetisation as a function of temperature simulated for lattices with lengths $L \in \{40, 60, 80, 100\}$ for 10^6 MC cycles.

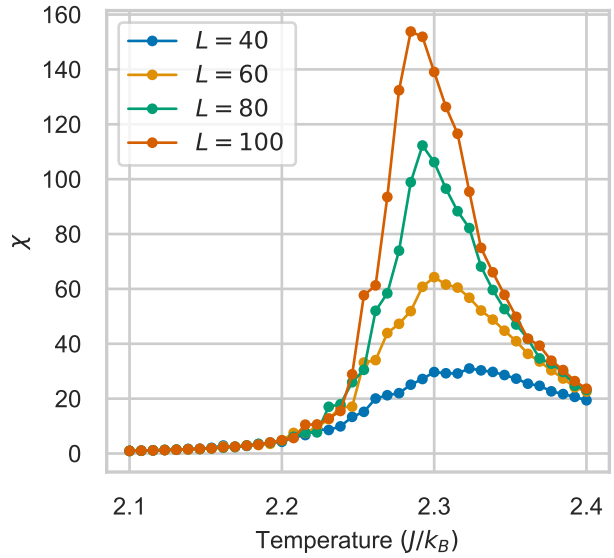

Figure 11: Susceptibility as function of temperature simulated for lattices with lengths $L \in \{40, 60, 80, 100\}$ for 10^6 MC cycles.

F. Estimating the critical temperature

With the results from Table V, we estimated the critical temperature $T_c(L = \infty)$ using the relation in (10). Specifically, we fitted a simple linear model

$$T_c(L) = \beta_0 + \beta_1 L^{-1} + \epsilon, \tag{12}$$

where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is a normally distributed noise term. We applied `LinearRegression` from the `Scikit-learn` Python library [8] to fit the model to the data values. The estimate of β_0 is our estimate of the critical temperature when $L = \infty$. We estimated

$$\beta_0 = 2.260 \pm 0.002.$$

L	$T_c(L)\,[J/k_B]$
40	2.323
60	2.3
80	2.292
100	2.285

Table V: Estimates of $T_c(L)$ for different lattice sizes. The estimates are based on the temperature corresponding to the maximum χ-value for each lattice.

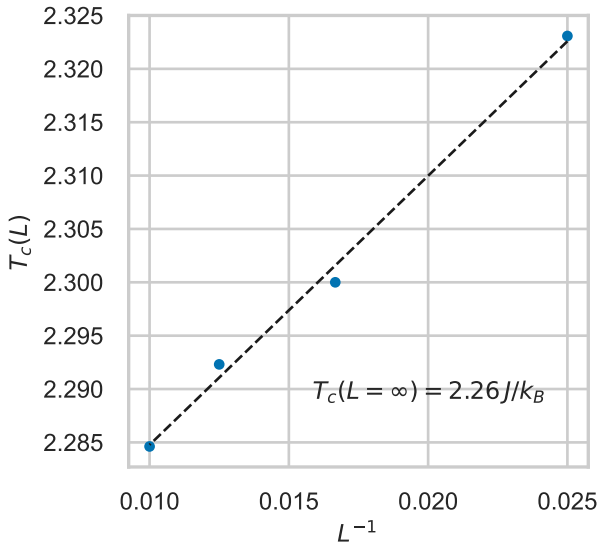

Figure 12: $T_c(L)$ values as a function of L^{-1}, with a fitted regression line. The regression fit estimates $T_c(L=\infty) = 2.260$.

The fitted regression line and the data points are plotted in Figure 12.

IV. DISCUSSION

A. Comparing the analytic and numeric solutions

Table III shows the result of the verification of our program. We see that the relative numerical error for the energy per spin ϵ and the magnetization per spin $|m|$ in all cases is less than or equal to 1%, even for only 5000 MC cycles. The discrepancy is bigger for the total heat capacity and total magnetization, and we see that increasing the amount of MC cycles drastically improves our numerical estimates of the heat capacity and susceptibility. However, even after 500 000 MC cycles, the relative error is still 3% for C_V, and 2% for χ. This error could be reduced further by running even more MC cycles, but it would in turn cost more computational power.

B. Exploring the burn-in time

In Figure 2 and Figure 3 we show the expectation values for the energy and and magnetization per spin as a function of MC cycles, when the temperature of the system is $T = 1\,J/K_B$. These figures serve as a good illustration of the burn-in period of our simulation, and gives us a way to visualize the convergence of the numerical calculation of the observables. We see that the randomly initiated lattice converges toward $\langle\epsilon\rangle = -2$ and $\langle|m|\rangle = 1$, while the lattice with an ordered initial condition, where all spins point in the same direction, begins and remains at the equilibrium values during all the MC cycles. This is in accordance with expecting to observe a fully magnetized lattice at low temperatures.

Figures 4 and 5 show the energy and magnetization in the case where $T = 2.4\ J/k_B$, and here we can clearly see that neither the ordered or unordered lattice begins near the equilibrium value, and in both cases they do not converge sufficiently until after $1\cdot10^4 - 2\cdot10^4$ MC cycles.

C. Estimating the probability distribution

When estimating the probability distributions for ϵ, we observe that the variance is significantly smaller when $T = 1.0\,J/k_B$ compared to $T = 2.4\,J/k_B$. This is expected, because $T = 2.4\,J/k_B$ is closer to the critical temperature, and we know that the heat capacity, i.e. the variance of the energy, sharply increases near the critical temperature. At low temperatures it is highly unlikely to observe spins pointing in opposite directions, causing the model to only slightly fluctuate from $\epsilon = -2$. As we move closer to the critical temperature, spins will flip at a higher frequency due to a higher temperature, increasing the variance of ϵ. We also note that for $T = 1\,J/k_B$, the sample mean is close to what we expect, namely that it is close to the value of a fully magnetised lattice.

D. Computation time with parallelization

When we parallelize our code, we get the values as shown in table IV. We see that running the Monte Carlo cycles on several threads dramatically decreases the runtime of the code. Our code was run on a machine with 4 "performance" cores which can support 2 threads simultaneously, meaning that 8 is the maximum amount of threads said computer can run. (Requesting more threads than are actually available is possible when using `OpenMP`, in which case the program will simply run on the available threads.) We see that when running for more than 6 threads, the runtime begins to decline slower, and the rest of the runtimes have a value of around ~ 4.5 s. Since we are creating a significant set of data files each time we run the program, we could speculate that the lower boundary for the runtime is limited by the operation of creating and writing the data files to the memory

of the computer, and this might be why running the algorithm with more than 6 threads yields almost no increase in efficiency, as compared to running with only 6 threads.

E. Exploring phase transitions

Below the critical temperature, we expect the lattice to be magnetized. We can see that this is the case for every lattice size shown in Figure 9. The magnetization decreases rapidly near the critical temperature, but does not reach zero as expected. Temperatures beyond the critical temperature show some magnetization, although it is lower for larger lattices.

We expect to observe a sharp increase in both the heat capacity and the susceptibility close to the critical temperature. This is related to the power law behaviour near the critical temperature, as discussed in Section II B. We also expect this increase to be sharper for larger lattices. If we look at Figure 10 and Figure 11, we see that both sharply increase as the temperature approaches the critical value. We also see that the maximum is larger with larger lattices, as expected.

Because of the phase transition, we expect to observe a somewhat sharper transition going from low to high energy as temperature increases, with larger lattices. When we look at Figure 8, we can see that the energy increases a little more sharply for $L = 100$ near the critical temperature, but overall the energies tend to overlap.

We did not compute the *autocorrelation time* when running the simulations for different temperatures, and this is could be an interesting aspect to consider in the future. The autocorrelation time is related to the correlation length, and tells us how many MC cycles we need to "skip" in order to get independent samples. In other words, taking the the autocorrelation time into account could have helped us get more accurate results. This is especially relevant for larger lattices near the critical temperature, as the correlation increases with these parameters.

F. Estimating the critical temperature

From the finite-scaling relation in (10) we applied a linear regression model and estimated the critical temperature to be $T_c(L = \infty) = 2.260 \pm 0.002$. Note that the error of this estimate is based on the error of the linear model, so the error of the simulated values has not been propagated to this estimate. The estimate misses Onsager's analytic solution by approximately 0.4%, in terms of the relative error. We consider this to be a satisfyingly accurate numerical result, although we could have improved the accuracy further by collecting even more samples. Especially close to the critical temperature, the Metropolis algorithm requires many MC cycles in order to get a large collection of independent samples. Another approach that could be interesting to explore in the future, is to implement algorithms that can more efficiently simulate the lattice near the critical temperature, such as the Wolff cluster algorithm [6]. The cluster algorithm flips clusters of spins, rather than a single spin at a time. We could also have studied the behaviour of even larger lattices.

We observed that the estimated critical temperatures $T_c(L)$ got closer to Onsager's solution in (9) as the lattice size increased, as expected. We note that our estimate of the critical temperature $T_c(L = \infty)$ is based on the maximal χ-value and the corresponding temperature for each lattice size. We also looked at the temperatures corresponding to the largest C_V-values, but using these values to fit the linear model in (12) gave a less accurate estimate. This was probably because, when using maximal heat capacity for estimating $T_c(L)$, the same $T_c(L)$ estimate was selected for $L = 40$ and $L = 60$. This was not alleviated when we simulated the lattices over a more fine-grained selection of temperature values in the interval $[2.1, 2.4]\, J/k_B$. This indicates that the estimated values of our model could be further improved.

V. CONCLUSION

We have simulated the phase transitions in a ferromagnetic Ising model using the Metropolis algorithm, and studied how simulating with different parameters affects our results in terms of numerical accuracy and computational cost. We found that for bigger lattice sizes, the estimated critical temperature of the system approached the analytic solution. From estimations of the critical temperature for lattices $L \in \{40, 60, 80, 100\}$, we fitted a linear regression model based on finite-size scaling relations. We were able to approximate the critical temperature for an infinitely large lattice as 2.260 ± 0.002, with a relative error of about 0.4%. Our estimate could be improved by running the simulations for a larger amount of Monte Carlo cycles, which would increase the accuracy of our results, but also increase the computational cost of the simulation software.

[1] S. Yu, D. Huang, W. Singer, and D. Nikolić, Cerebral Cortex **18**, 2891 (2008).
[2] W. Yap and H. Saroff, Journal of Theoretical Biology **30**, 35 (1971).
[3] L. Onsager, Phys. Rev. **65**, 117 (1944).
[4] M. Hjorth-Jensen, *Computational Physics: Lecture Notes Fall 2015* (Department of physics, University of Oslo, 2015).
[5] J. O. Haerter and K. Sneppen, *Complex Physics* (Copenhagen University, 2020).
[6] H. G. Katzgraber, Introduction to monte carlo methods (2009).
[7] OpenMP Architecture Review Board, OpenMP application program interface version 5.2 (2022).
[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, Journal of Machine Learning Research **12**, 2825 (2011).

Appendix A: Calculations and raw code

Analytical expressions

Consider a 2×2-lattice containing four spin states s_i and periodic boundary conditions. To summarize all the possible states of the system without double-counting, we multiply the different states as illustrated to the left in Figure 13. The values we need to know for our analytical calculations are the total energy, total magnetization and degeneracy of the different states. We keep track of the different states by counting how many points with spin +1 there are within the lattice. One special case is when we have two states with +1 spins. Illustrated in Figure 14 to the right, we see that the degeneracy of this configuration is four, as there are two ways of putting the spins horizontally and two ways of putting the spins vertically. The total energy and total magnetization are both 0. But, for the configuration to the left there are only two possible configurations, meaning the degeneracy is only two, the total energy is 8J and the total magnetization is 0. The possible states are given in Table II.

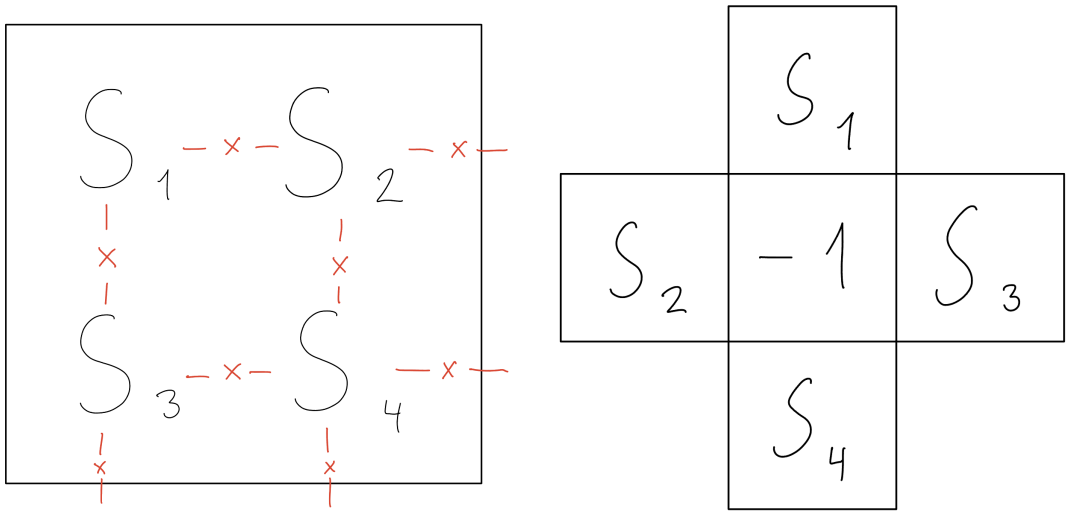

Figure 13: **To the left** a sketch of how we do multiplication between the different spin states s_n, without double-counting the state pairs. **To the right** a sketch of a lattice with an arbitrary size used in calculation of the possible states of ΔE.

If we look at the Boltzmann factor, given as $e^{-\beta \Delta E}$, the shift in energy due to neighbouring spins is always finite, since we are only dealing with spin-up and spin-down. Looking at a lattice with arbitrary size as illustrated to the right in Figure 13, we look at the case where the center value is constant while we change the spins around. Since we are looking at the change in energy, ΔE will only change when one spin state changes. The different configurations of spins will be when all states around the central point are -1, all states are 1, one state is in -1, two states are in -1 or three states are in -1. This gives us only five possible values for ΔE.

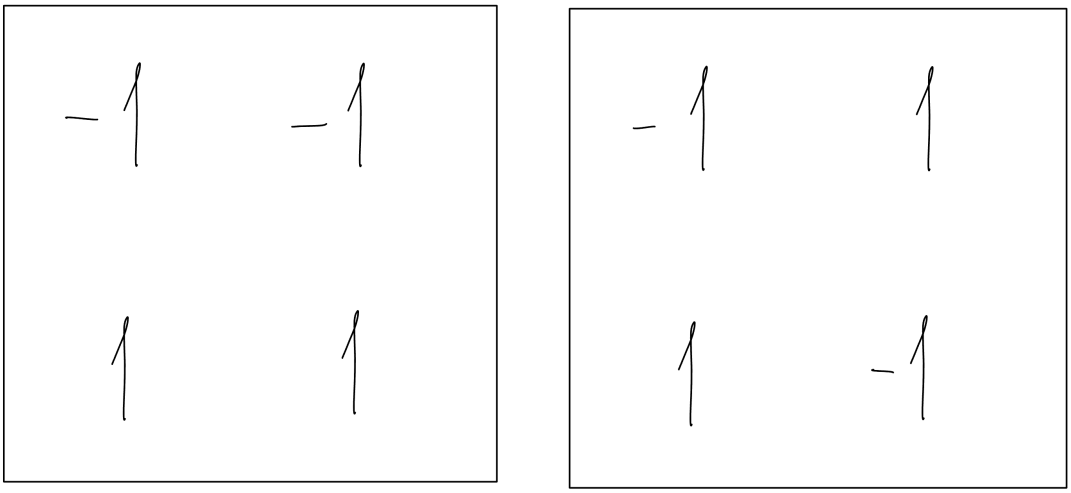

Figure 14: **To the right** we can see the configuration with two +1 states that gives a total energy of 8J. **To the left** we can see the configuration with two +1 states that gives a total energy of 0J.

With the given case of a 2×2 lattice we can analytically solve the partition function, the expected value of energy per spin, the expected value of magnetization per spin, and the specific heat capacity and susceptibility normalized to number of spins. We start with the partition function for $N = 4$, where we insert the values from the possible states in Table II into Equation (6) and solve for every possible state by

$$\begin{aligned} Z &= \sum_{\mathbf{s}} e^{-\beta E(\mathbf{s})} = e^{-\beta \cdot -8J} + 4e^{-\beta \cdot 0} + 4e^{-\beta \cdot 0} + 2e^{-\beta \cdot 8J} + 4e^{-\beta \cdot 0} + e^{-\beta \cdot -8J} \\ &= 2\left(e^{\beta 8J} + e^{-\beta 8J}\right) + 12 \quad \text{or} \quad 4\left(\cosh \beta 8J + 3\right), \end{aligned}$$

where we multiply every state with its degeneracy to get the correct values. Next, we want to calculate the expected value of energy per spin by finding the sum of Equation (3) multiplied with Equation (5). We solve for every possible state by

$$\begin{aligned} \langle \epsilon \rangle &= \sum_{\mathbf{s}} \epsilon(\mathbf{s}) p(\mathbf{s}; T) = \sum_{\mathbf{s}} \frac{E(\mathbf{s})}{N} \frac{1}{Z} e^{-\beta E(\mathbf{s})} \\ &= \frac{-8J}{4} \frac{1}{Z} e^{-\beta \cdot -8J} + 2\frac{8J}{4} \frac{1}{Z} e^{-\beta \cdot 8J} + \frac{-8J}{4} \frac{1}{Z} e^{-\beta \cdot -8J} \\ &= \frac{16J}{4Z}\left(e^{-\beta 8J} - e^{\beta 8J}\right) \\ &= \frac{4J\left(e^{-\beta 8J} - e^{\beta 8J}\right)}{4\left(\cosh \beta 8J + 3\right)} \\ &= -\frac{2J \sinh \beta 8J}{\cosh \beta 8J + 3}, \end{aligned} \tag{A1}$$

where we have extracted a part of the result and replaced it with $\sinh x$, which is given as

$$\sinh x = \frac{e^x - e^{-x}}{2}.$$

We now want the expected value of the energy per spin, where energy per spin is squared. This is done precisely the

same as for Equation (A1), only we square Equation (3). This can be solved by

$$\begin{aligned}
\langle \epsilon^2 \rangle &= \sum_{\mathbf{s}} (\epsilon(\mathbf{s}))^2 p(\mathbf{s};T) = \sum_{\mathbf{s}} \left(\frac{E(\mathbf{s})}{N}\right)^2 \frac{1}{Z} e^{-\beta E(\mathbf{s})} \\
&= \left(\frac{-8J}{4}\right)^2 \frac{1}{Z} e^{-\beta \cdot -8J} + 2\left(\frac{8J}{4}\right)^2 \frac{1}{Z} e^{-\beta \cdot 8J} + \left(\frac{-8J}{4}\right)^2 \frac{1}{Z} e^{-\beta \cdot -8J} \\
&= \frac{16J^2}{Z}\left(e^{\beta 8J} + e^{-\beta 8J}\right) \\
&= \frac{16J^2\left(e^{\beta 8J} + e^{-\beta 8J}\right)}{4\left(\cosh \beta 8J + 3\right)} \\
&= \frac{8J^2 \cosh \beta 8J}{\cosh \beta 8J + 3},
\end{aligned} \tag{A2}$$

where we have extracted a part of the result and replaced it with $\cosh x$, which is given as

$$\cosh x = \frac{e^x + e^{-x}}{2}.$$

The next solution we want is the expected value of the magnetization per spin. Here, we sum the multiplication between the absolute value of the magnetization per spin given in Equation (4) with Equation (5) for all possible states. This can be solved by

$$\begin{aligned}
\langle |m| \rangle &= \sum_{\mathbf{s}} |m(\mathbf{s})| p(\mathbf{s};T) = \sum_{\mathbf{s}} \left|\frac{M(\mathbf{s})}{N}\right| \frac{1}{Z} e^{-\beta E(\mathbf{s})} \\
&= \left|\frac{-4}{4}\right| \frac{1}{Z} e^{-\beta \cdot -8J} + 4\left|\frac{-2}{4}\right| \frac{1}{Z} e^{-\beta \cdot 0} + 4\left|\frac{-2}{4}\right| \frac{1}{Z} e^{-\beta \cdot 0} + \left|\frac{4}{4}\right| \frac{1}{Z} e^{-\beta \cdot -8J} \\
&= \frac{2}{Z}\left(e^{\beta 8J} + 2\right) \\
&= \frac{2\left(e^{\beta 8J} + 2\right)}{4\left(\cosh \beta 8J + 3\right)} \\
&= \frac{\left(e^{\beta 8J} + 2\right)}{2\left(\cosh \beta 8J + 3\right)}.
\end{aligned}$$

We also want the expected value of the magnetization per spin, where the magnetization per spin is squared. We will do this in the same way as we did when calculating Equation (A2). This can be solved by

$$\begin{aligned}
\langle m^2 \rangle &= \sum_{\mathbf{s}} (m(\mathbf{s}))^2 p(\mathbf{s};T) = \sum_{\mathbf{s}} \left(\frac{M(\mathbf{s})}{N}\right)^2 \frac{1}{Z} e^{-\beta E(\mathbf{s})} \\
&= \left(\frac{-4}{4}\right)^2 \frac{1}{Z} e^{-\beta \cdot -8J} + 4\left(\frac{-2}{4}\right)^2 \frac{1}{Z} e^{-\beta \cdot 0} + 4\left(\frac{-2}{4}\right)^2 \frac{1}{Z} e^{-\beta \cdot 0} + \left(\frac{4}{4}\right)^2 \frac{1}{Z} e^{-\beta \cdot -8J} \\
&= \frac{2}{Z}\left(e^{\beta 8J} + 1\right) \\
&= \frac{2\left(e^{\beta 8J} + 1\right)}{4\left(\cosh \beta 8J + 3\right)} \\
&= \frac{\left(e^{\beta 8J} + 1\right)}{2\left(\cosh \beta 8J + 3\right)}.
\end{aligned}$$

We want to calculate the expected value of the total energy and total magnetization for the 2×2-lattice along with the solution where they are squared as with the solutions above. The expected value for the total energy can be found

by the sum of Equation (1) multiplied with Equation (5) and solved by

$$\begin{aligned}\langle E\rangle &= \sum_{\mathbf{s}} E(\mathbf{s})\frac{1}{Z}e^{-\beta E(\mathbf{s})}\\ &= \frac{-8J}{Z}e^{-\beta\cdot -8J} + 2\frac{8J}{Z}e^{-\beta\cdot 8J} + \frac{-8J}{Z}e^{-\beta\cdot -8J}\\ &= \frac{16J}{Z}\left(e^{-\beta 8J} - e^{\beta 8J}\right)\\ &= \frac{16J\left(e^{-\beta 8J} - e^{\beta 8J}\right)}{4\left(\cosh\beta 8J + 3\right)}\\ &= -\frac{8J\sinh\beta 8J}{\cosh\beta 8J + 3},\end{aligned}$$

where the squared of this is given as

$$\langle E\rangle^2 = \left(-\frac{8J\sinh\beta 8J}{\cosh\beta 8J + 3}\right)^2 = \frac{64J^2\sinh^2\beta 8J}{(\cosh\beta 8J + 3)^2}.$$

The expected value for the total energy, where the total energy is squared can be solved by

$$\begin{aligned}\langle E^2\rangle &= \sum_{\mathbf{s}} \left(E(\mathbf{s})\right)^2\frac{1}{Z}e^{-\beta E(\mathbf{s})}\\ &= \frac{64J^2}{Z}e^{-\beta\cdot -8J} + 2\frac{64J^2}{Z}e^{-\beta\cdot 8J} + \frac{64J^2}{Z}e^{-\beta\cdot -8J}\\ &= \frac{128J^2}{Z}\left(e^{\beta 8J} + e^{-\beta 8J}\right)\\ &= \frac{128J^2\left(e^{\beta 8J} + e^{-\beta 8J}\right)}{4\left(\cosh\beta 8J + 3\right)}\\ &= \frac{64J^2\cosh\beta 8J}{\cosh\beta 8J + 3}.\end{aligned}$$

The expected value for the total magnetization can be found by taking the sum of Equation (2) multiplied by Equation (5) for each state. This is solved by

$$\begin{aligned}\langle |M|\rangle &= \sum_{\mathbf{s}} |M(\mathbf{s})|\frac{1}{Z}e^{-\beta E(\mathbf{s})}\\ &= \frac{|4|}{Z}e^{-\beta\cdot -8J} + 4\frac{|-2|}{Z}e^{-\beta\cdot 0} + 4\frac{|-2|}{Z}e^{-\beta\cdot 0} + \frac{|4|}{Z}e^{-\beta\cdot -8J}\\ &= \frac{8}{Z}\left(e^{\beta 8J} + 2\right)\\ &= \frac{8\left(e^{\beta 8J} + 2\right)}{4\left(\cosh\beta 8J + 3\right)}\\ &= \frac{2\left(e^{\beta 8J} + 2\right)}{\cosh\beta 8J + 3},\end{aligned}$$

where the squared of this expected value is given as

$$\langle |M|\rangle^2 = \left(\frac{2\left(e^{\beta 8J} + 2\right)}{\cosh\beta 8J + 3}\right)^2 = \frac{4\left(e^{\beta 8J} + 2\right)^2}{(\cosh\beta 8J + 3)^2}.$$

The expected value for the total magnetization, where the total magnetization is squared can be solved by

$$\begin{aligned}
\langle M^2 \rangle &= \sum_{\mathbf{s}} \left(M(\mathbf{s})\right)^2 \frac{1}{Z} e^{-\beta E(\mathbf{s})} \\
&= \frac{16}{Z} e^{-\beta \cdot -8J} + 4\frac{4}{Z} e^{-\beta \cdot 0} + 4\frac{4}{Z} e^{-\beta \cdot 0} + \frac{16}{Z} e^{-\beta \cdot -8J} \\
&= \frac{32}{Z} \left(e^{\beta 8J} + 1\right) \\
&= \frac{32 \left(e^{\beta 8J} + 1\right)}{4 \left(\cosh \beta 8J + 3\right)} \\
&= \frac{8 \left(e^{\beta 8J} + 1\right)}{\cosh \beta 8J + 3}.
\end{aligned}$$

We can now, with all the analytical solutions to the expected values, find the specific heat capacity normalized to number of spins with Equation (7). This is solved by

$$\begin{aligned}
C_V &= \frac{1}{N} \frac{1}{k_B T^2} (\langle E^2 \rangle - \langle E \rangle^2) \\
&= \frac{1}{N k_B T^2} \left(\frac{64J^2 \cosh \beta 8J}{\cosh \beta 8J + 3} - \frac{64J^2 \sinh^2 \beta 8J}{(\cosh \beta 8J + 3)^2} \right) \\
&= \frac{64J^2}{N k_B T^2} \left(\frac{\cosh \beta 8J}{\cosh \beta 8J + 3} - \frac{\sinh^2 \beta 8J}{(\cosh \beta 8J + 3)^2} \right) \\
&= \frac{64J^2}{N k_B T^2} \left(\frac{\cosh \beta 8J (\cosh \beta 8J + 3)}{(\cosh \beta 8J + 3)(\cosh \beta 8J + 3)} - \frac{\sinh^2 \beta 8J}{(\cosh \beta 8J + 3)^2} \right) \\
&= \frac{64J^2}{N k_B T^2} \left(\frac{\cosh^2 \beta 8J + 3 \cosh \beta 8J - \sinh^2 \beta 8J}{(\cosh \beta 8J + 3)^2} \right) \\
&= \frac{64J^2}{N k_B T^2} \left(\frac{3 \cosh \beta 8J + 1}{(\cosh \beta 8J + 3)^2} \right).
\end{aligned}$$

Finally, we want to find the magnetic susceptibility normalized to number of spins with Equation (8). This can be solved by

$$\begin{aligned}
\chi &= \frac{1}{N} \frac{1}{k_B T} (\langle M^2 \rangle - \langle |M| \rangle^2) \\
&= \frac{1}{N k_B T^2} \left(\frac{8 \left(e^{\beta 8J} + 1\right)}{\cosh \beta 8J + 3} - \frac{4 \left(e^{\beta 8J} + 2\right)^2}{(\cosh \beta 8J + 3)^2} \right) \\
&= \frac{4}{N k_B T^2} \left(\frac{2 \left(e^{\beta 8J} + 1\right)}{\cosh \beta 8J + 3} - \frac{\left(e^{\beta 8J} + 2\right)^2}{(\cosh \beta 8J + 3)^2} \right) \\
&= \frac{4}{N k_B T^2} \left(\frac{\left(2e^{\beta 8J} + 2\right) (\cosh \beta 8J + 3)}{(\cosh \beta 8J + 3)(\cosh \beta 8J + 3)} - \frac{\left(e^{\beta 8J} + 2\right)^2}{(\cosh \beta 8J + 3)^2} \right) \\
&= \frac{4}{N k_B T^2} \left(\frac{e^{2(\beta 8J)} + 5e^{\beta 8J} + e^{-\beta 8J} + 7 - e^{2(\beta 8J)} - 4e^{\beta 8J} - 4}{(\cosh \beta 8J + 3)^2} \right) \\
&= \frac{4}{N k_B T^2} \left(\frac{e^{-\beta 8J} + 3e^{\beta 8J} + 3}{(\cosh \beta 8J + 3)^2} \right).
\end{aligned}$$