Estimating the critical temperature of a 2D Ising model through the Markov Chain Monte Carlo method

Einar Skoglund, Emil J. Kvernevik
(Dated: November 21, 2023)

In this study, we apply the Markov Chain Monte Carlo method (MCMC) using the Metropolis algorithm to investigate the temperature-dependent behavior of the two-dimensional Ising model. Our focus is approximating the quantities that describe the state of the system, namely the average energy and magnetization, specific heat capacity, magnetic susceptibility, along with an approximation of the critical temperature. Our computed results are compared to an analytical baseline of the 2×2 model, giving a relative difference of < 4 %. Then the numerically estimated critical temperature for lattices of infinite size is compared to Onsager's analytical result, where we find that they agree with a relative error of 0.5%.

The GitHub repository for this article can be found at https://github.uio.no/emiljk/FYS3150.git.

I. INTRODUCTION

The Ising model, or Ising-Lenz model, is a mathematical model used to simulate the behavior of ferromagnetic materials. In addition to paving the way for a profound understanding of magnetic phase transitions, the Ising model has also found applications in other areas of physics; e.g. analysis of effects on alloy evolution from irradiation [1].
The model was originally proposed as a problem by Wilhelm Lenz for his student Ernst Ising, who would go on to solve it in one dimension in 1925 - revealing an absence of phase transition in this case.
The solution for the two-dimensional Ising model would be emerged much later in time, almost 20 years after Ising's solution, in 1944. This breakthrough was achieved by a physicist called Lars Onsager, and revealed the presence of a phase transition in the two-dimensional Ising model - confirming the existence of a critical temperature. This critical temperature is a characteristic temperature below which a ferromagnet undergoes a spontaneous magnetization, and above which a transition to a paramagnetic phase.
A material in a ferromagnetic phase aligns its magnetic moments spontaneously, even in the absence of an external magnetic field. In a paramagnetic phase however, each individual magnetic moment only responds and aligns if subjected to an external magnetic field - if any. In the case where there is no external magnetic field, net magnetization becomes zero.

This brings us to the focus of this study. The aim is to numerically estimate the critical temperature of a two-dimensional square-lattice Ising model, and explore the phase behavior of a ferromagnet. For the scope and purposes of this study, the Markov Chain Monte Carlo method stands out as a particularly well-suited method for simulation. This is due to how the method does not require that every possible state is found, as it uses random numbers to draw samples from probability distributions. The resulting samples of which can then be used to approximate probability distributions and expectation values for various quantities of interest. Further, by simulating the model and letting it evolve over a range of temperatures, we can determine an area of interest where the net magnetization tends toward zero - this area would then correspond to the critical temperature where the ferromagnet transitions to a paramagnetic phase. Additionally, we also wish to grasp how accurately our model can derive the systems critical temperature. As such, upon determining it for a finite model, we then compare it to Onsager's analytical result for an infinite model.

The structure of this study is as follows. In section II we provide an overview of the Ising model, and the theoretical framework needed to understand the model. Here we cover analytically derived quantities and the numerical approach employed, namely the Markov Chain Monte Carlo method, along with an extensive explanation on the procedure of simulating the model. In section III the results from the simulations are presented, and then compared to an analytical baseline. After which we discuss how consistent they are with analytical values, and suggest different approaches and future work to increase the accuracy of said results. And lastly, in section IV we summarize our findings and discussion, and conclude the study.

II. METHODS

In this section we introduce the notation and quantities used to describe the Ising model, explain how we implement the MCMC Metropolis algorithm in our simulation, and describe the methods and quantities used to derive our results.

A. The Ising model

To model the ferromagnet we make use of spins s_i, that can take values $s_i = +1$ ($\uparrow$) or $s_i = +1$ ($\downarrow$). These

are contained in a lattice of length L, consisting of a total of $N = L^2$ spins, and we denote the specific spin configuration, or microstate of the system as the matrix

$$\mathbf{s} = \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1L} \\ s_{21} & s_{22} & \cdots & s_{2L} \\ \vdots & \vdots & \ddots & \vdots \\ s_{L1} & s_{L2} & \cdots & s_{LL} \end{bmatrix}. \tag{1}$$

The energy is defined as a state of the interactions between the spins. In the Ising model we limit ourselves to the immediate neighbors (we neglect any long-range interactions that may occur between "clusters" of lattices [3]) as shown in FIG 1. The total energy then becomes

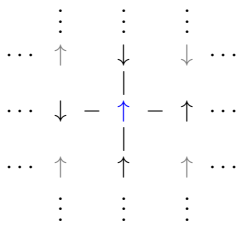


FIG. 1. An illustration of the spin interactions in an example 3×3 domain within a larger lattice. The spin in the middle (drawn in blue) is only affected by the interactions (drawn as horizontal and vertical lines) between the closest neighbors (drawn as black arrows). We do not count the interactions from the spins on the diagonal (the gray arrows).

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

where J is a coupling constant related to the interactions, and $\langle kl \rangle$ denotes that we only count the interactions once. Since we are only looking at a tiny finite domain of an actual macroscopic ferromagnet, we model the interactions at the boundary by using periodic boundary conditions, as illustrated for a 2×2 Ising model in FIG 2. Periodic boundary conditions entails that we allow

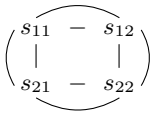


FIG. 2. An illustration of all the interactions in a 2×2 Ising model using periodic boundary conditions. The interactions are shown as horizontal, vertical and curved lines. All spins have an interaction above, beneath, to the left and to the right.

the spins at the boundaries to interact with the spins on the opposite side. This is to make up for the spins we are *missing* at the boundaries, which we would not have in a lattice of infinite size. By using periodic boundary conditions we can in a more accurate way mimic the behavior of an infinite system, even if we are working with lattices of finite sizes.

As we only want to calculate each interaction once, we can calculate the total energy by e.g. only count the interactions to the right and below each spin. This is shown in FIG 3 using a 3×3 lattice where we have projected the first column to the right of the last column, and the first row below the last row.

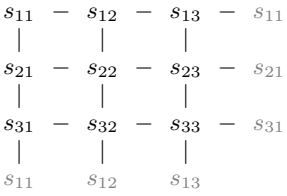


FIG. 3. An illustration of how one can implement periodic boundary conditions without double counting. The figure shows a 3×3 lattice, where we have projected (shown in gray) the first column to the right of the last column, and the first row below the last row. By only counting the interactions to the right and below each spin we avoid double counting the interactions.

To find the total magnetization of the system we simply add all the individual spins.

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

As we are only interested in the amount of spins parallel to each other, we will be using the absolute value $|M|$.

To represent the total energy and magnetization per spin we introduce the normalized energy

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

and normalized absolute magnetization

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

As for a real ferromagnet, we want the model to be able to interact with the surroundings via heat-exchange. This is done by performing spin-flips which change the energy of the system.

By performing many spin-flips we can estimate the expectation value of the energy of the system; this is due to the law of large numbers [3]. This can also be done analytically by using the probability of a given energy-state $p(\mathbf{s})$ in the equation below:

$$\langle E \rangle = \sum_{\mathbf{s}_i} E(\mathbf{s}) p(\mathbf{s}), \tag{6}$$

where we have used that E is a discrete quantity, as seen from (2). The expectation value for M is found in similar way.
From Boltzmann statistics (see for example [3]) we have that the probability for a state $\mathbf{s}$ at a given temperature T follows the Boltzmann distribution:

$$p(\mathbf{s};T) = \frac{1}{Z}e^{-\beta E(\mathbf{s})}, \tag{7}$$

where Z is the partition function given by

$$Z = \sum_{\text{all possible } \mathbf{s}} e^{-\beta E(\mathbf{s})}, \tag{8}$$

where $\beta = \frac{1}{k_B T}$ is the reduced temperature where k_B is the Boltzmann constant. The partition function sums over all 2^N possible states. For the 2×2 lattice these $2^4 = 16$ states are shown in appendix IV, and a summary of the macrostates, i.e. the states with same E and M, are listed in table I. From this table and (6) we show in

TABLE I. The possible total energies and magnetizations along with their degeneracies for the 2×2 Ising model.

Number of ↑	E	M	Degeneracy
0	-8J	-4	1
1	0	-2	4
2	8J	0	2
2	0	0	4
3	0	2	4
4	-8J	4	1

appendix IV that the expectation value for ϵ for the 2×2 Ising model is

$$\langle \epsilon \rangle = -\frac{8J}{Z}\sinh(8J\beta). \tag{9}$$

We find the expectation value for $|m|$ as

$$\langle |m| \rangle = \frac{2e^{8J\beta} + 4}{Z}. \tag{10}$$

We are also interested in two other quantities to describe the state of the Ising model, namely the specific heat capacity (normalized to number of spins):

$$C_V = \frac{1}{N}\frac{1}{k_B T^2}(\langle E^2 \rangle - \langle E \rangle^2), \tag{11}$$

and the magnetic susceptibility (also normalized to number of spins):

$$\chi = \frac{1}{N}\frac{1}{k_B T}(\langle M^2 \rangle - \langle |M| \rangle^2). \tag{12}$$

In appendix IV we show that for the 2×2 Ising model these quantities becomes

$$C_V = \frac{256J^2}{Z^2}\frac{1}{k_B T^2}(3\cosh(8J\beta) + 1), \tag{13}$$

and

$$\chi = \frac{16}{Z^2}\frac{1}{k_B T}(e^{-8J\beta} + 3e^{8J\beta} + 3). \tag{14}$$

As the quantities above all depend on the temperature, we know that as the model eventually explores higher temperatures, they will begin to change [3]. Moreover, for a ferromagnet there exists a critical temperature T_c where $\langle |m| \rangle$ becomes zero. Additionally, we know that near T_c both C_V and χ will exhibit divergent behavior. This is indicative of a phase-transition, where the ferromagnet transitions into a paramagnet. Investigating the evolution of the model over higher temperatures and observing the behavior of the C_V and χ, allows us to then determine T_c for each lattice size. Further, by using our results for the critical temperature and the scaling relation (15),

$$T_c(L) - T_c(L = \infty) = aL^{-1}, \tag{15}$$

a comparison can then be then made between the derived critical temperatures for different lattice sizes and Onsager's analytical result (16).

$$T_c(L = \infty) = \frac{2}{ln(1 + \sqrt{2})} J/k_B \approx 2.269\, J/k_B \tag{16}$$

Where $T_c(L)$ represents the critical temperature of a finite system given a lattice size L, $T_c(L = \infty)$ Onsager's analytical result for an infinite system, and a a scaling constant.
For lattices of size $L > 2$ we will not try to find analytical solutions for the quantities used for estimating the critical temperature. With 2^N number of possible states we will need to find a lot of microstates for lattices with large N's. Therefore we will use a numerical approach when working with larger lattices.

B. Numerical algorithm

The **Monte Carlo** method is a numerical procedure to generate a collection, or sampling, of states using random numbers. From the law of large numbers, [3] we can get the true sample of states by just generating a finite amount of randomly chosen states. This means that it is possible to get a good estimate of all possible states and their degeneracies without explicitly finding all of them, as we did for the 2×2 model.
However, choosing states completely at random turns out to be very inefficient for the Ising model, as we for large lattices need to generate numerous states just to get a fraction of all the possible ones. For a 20×20 lattice there are $2^{400} \approx 2.6 \times 10^{120}$ possible states. We therefore need a more efficient way of choosing states.

As known from the second law of thermodynamics [3], a system will tend towards states with higher

probabilities. From (7) we see that the probability of a given state is determined by the *energy* of that state. Starting from an initial state, we can then determine the probability of moving to a different state by using the difference in energy between the two states. Finding probabilities of evolving to new states by using only the properties of the current state, is known as a **Markov chain**. Choosing new states in the Monte Carlo method based on the current state, is known as the **Markov Chain Monte Carlo** (MCMC) method.

To actually evolve the system in a particular direction, we can change its energy by flipping a random spin. This can be seen as the system interacting with the environment via energy-exchange. However, to evolve the system in way that is realistic according to the laws of thermodynamics [3], we need to use some kind of rule for whether we want to flip a particular spin or not. In our algorithm we will use the **Metropolis-Hastings acceptance rule** as follows: By choosing a random number $r \sim U(0,1)$, where $U(0,1)$ is a uniform distribution between 0 and 1, we will make a change from a state **s** to a proposed state **s'** if

$$r \leq \min\left(\frac{p(\mathbf{s}')T(\mathbf{s}|\mathbf{s}')}{p(\mathbf{s}T(\mathbf{s}')|\mathbf{s})}\right), \tag{17}$$

where $T(i|j)$ is the probability of choosing i when being in j, i.e. the probability of choosing some new state when being in an initial state. We will choose the new state (the spin to flip) from a uniform distribution, meaning that the T's are always the same. We can then simplify (17) as

$$r \leq \min\left(1, \frac{p(\mathbf{s}')}{p(\mathbf{s})}\right). \tag{18}$$

This is known as the **Metropolis** acceptance rule, which when used in the MCMC method becomes the **Metropolis algorithm**.
From equation (18) it is clear that if the probability of the proposed state $p(\mathbf{s}') \geq p(\mathbf{s}_i)$ the proposed state will always be accepted, and the system will move towards a state with higher probability. We note however that if $p(\mathbf{s}') \leq p(\mathbf{s}_i)$ the proposed state will *sometimes* be accepted, allowing us to explore other less probable states even after reaching the most probable. This should result in accepting new states according to the *true* probability distribution of a given lattice.

In (18) we can use the Boltzmann distribution (7) to find the ratio of probabilities as

$$\frac{p(\mathbf{s}')}{p(\mathbf{s}_i)} = \frac{(1/Z)e^{-\beta E(\mathbf{s}')}}{(1/Z)e^{-\beta E(\mathbf{s}_i)}} = e^{-\beta(E(\mathbf{s}')-E(\mathbf{s}_i))} = e^{-\beta\Delta E}, \tag{19}$$

where we have introduced the energy difference $\Delta E = E(\mathbf{s}') - E(\mathbf{s}_i)$.
In appendix IV we show that for lattices of size $L > 2$ the energy difference can only take the five values $\Delta E = (-8, -4, 0, 4, 8)$ J. By creating two vectors $\mathbf{dE} = (-8, -4, 0, 4, 8)$ J and $\mathbf{dp} = e^{-\beta dE}$, we can find the correct probability ratio $\mathbf{dp}(i)$ from

$$\mathbf{dE}(i) = 4(i-2) \qquad i \in \{0, 4\} \tag{20}$$

$$\rightarrow i = \frac{\mathbf{dE}(i)}{4} + 2. \tag{21}$$

Using this we only need to determine $\mathbf{dE}(i)$ to find the probability ratio, meaning that we do not need to calculate the full Boltzmann factor $e^{-\beta\Delta E}$ every time we propose a new state.
To determine $\mathbf{dE}(i)$ for a given spin flip we use the periodic boundary conditions without double counting illustrated in FIG 3. From [2] we can find the row and column of the neighboring spins of a spin $s_{i,j}$ (as in FIG 1) as

$$\text{row} = (L + i + d_{\text{row}}) \mod (L) \quad d_{\text{row}} \in (-1, 1) \tag{22}$$

$$\text{col} = (L + j + d_{\text{col}}) \mod (L) \quad d_{\text{col}} \in (-1, 1) \tag{23}$$

where -1 for d_{row} or d_{col} refers to the neighbor above or left, and +1 for d_{row} or d_{col} refers to the below or right neighbor.

Algorithm 1 MCMC Metropolis

```
s_{i=0} ← [s_{1,1}, s_{1,2}, s_{1,3}, ..., s_{N,N}] ▷ Generate an initial state
for i = 0, 1, ..., N_cycles do
    for i = 0, 1, ..., N_spins do
        s'                          ▷ Generate candidate state by flipping a spin
        r ~ U(0,1)                  ▷ Random number between 0 and 1
        if r ≤ min(1, p(s')/p(s_i)) then
            s_{i+1} = s'            ▷ Accept move
        else
            if r ≥ min(1, p(s')/p(s_i)) then
                s_{i+1} = s_i       ▷ Reject move
    E_tot ← E                       ▷ Compute and store E
    M_tot ← |M|                     ▷ Compute and store |M|
    ...                             ▷ Compute other values of interest.
```

We have listed the full MCMC Metropolis algorithm in 1, and we refer to N possible spin flips as one MCMC **cycle**. After each cycle we measure (i.e. calculate) values that can be used to describe the system, such as $\langle\epsilon\rangle$, $\langle|m|\rangle$, C_V and χ. This is done on average to give each spin a chance to interact with the environment before doing a measurement.

It is important to note however, that the samples generated in the MCMC method are not independent, i.e. they are correlated. This is an effect of the models aforementioned Markov Chain property, with the probability of the next state being dependent on the current. A consequence of this is that the samples generated

during the first cycles of the MCMC simulation could be heavily influenced by the initial state. Generating many samples around the initial state could be a problem if the initial state is highly improbable, causing us to generate states that we should not be able to measure within a few measurements. When calculating expectation values from a finite sample of measurements heavily influenced by the initial state, the result can be a value *pushed* in a less probable direction. One should therefore aim to begin an MCMC simulation in a probable state.
When we do not know which states are more probable, we can avoid the problem of correlated samples by simply ignoring the first measurements during the simulation. From the second law of thermodynamics [3] we expect the system to tend towards states with higher probability. The amount of MCMC cycles needed to reach some kind of equilibrium of states with high probability is referred to as the **burn-in time**. By not using the samples generated during the burn-in time, we should expect the MCMC simulation to give expectation values closer to the *true* expectation values of the system.

C. Simulation

To test our algorithm we compare it to the analytical results for the 2×2 Ising model found in section II A. We run the simulation for $T = 1.0\ J/k_B$ for 10^6 MCMC cycles starting from a randomly initialized spin lattice. By showing how the quantities $\langle\epsilon\rangle$, $\langle|m|\rangle$, C_V and χ changes as a function of MCMC cycles, we estimate how many cycles we need to get good agreement with the analytical result.

To estimate the burn-in time for the algorithm we use a lattice of size $L = 20$, temperatures $T = 1.0\ J/k_B$ and $T = 2.4\ J/k_B$, and a simulation time of 10^6 MCMC cycles per temperature. From this we show how the numerical estimates of $\langle\epsilon\rangle$ and $\langle|m|\rangle$ changes as a function of MCMC cycles, depending on if we start from an ordered or random initial spin lattice. We also compare these results with how the quantity ϵ evolves for the random initial spin configurations for the two temperatures, and discuss how measurements of different quantities can then be used to estimate the burn-in time.

For the same lattice size, temperatures and MCMC cycles we estimate the probability distribution for ϵ by starting from random initial spin configurations. For each temperature we run three simulations where we include different burn-in times. For $T = 1.0\ J/k_B$ we use burn-in times of 0, 600 and 10^4 MCMC cycles, and for $T = 2.4\ J/k_B$ we use burn-in times of 0, 1 and 10^4 MCMC cycles. We comment on the different results for the probability distributions with different burn-in times. We also discuss the variance of the distributions at the two different temperatures.

To be able to run simulations with even larger choices of L, we parallelize our numerical method over the temperature values. Here we make use of parallel threads, which enable us to run MCMC cycles for several temperatures at once, reducing the total run-time for the computation. To see the effect of the parallelization, we use a lattice of size $L = 20$ for 11 temperatures in the range $T = \{2.1, 2.4\}\ J/k_B$ with 10^5 MCMC cycles per temperature, where we compare the computation-time by using 1, 2, 4, 6, 8 and 10 threads. From this we can calculate the speed-up factor per thread:

$$f_{\text{speed-up}} = \frac{t_1}{t_{\text{n threads}}}, \tag{24}$$

where t_1 is the elapsed time for just 1 thread, which corresponds to no parallellization, and $t_{\text{n threads}}$ is the elapsed time for n-threads. We perform three timing tests for each choice of threads on a Macbook Air M1 with 8 available threads.

With the parallelization in order, we are then geared to explore the Ising model with lattice sizes $L = 40, 60, 80, 100$. Plots of the aforementioned quantities of interest are then made, with a focus on the temperatures in the range $T \in [2.1,\ 2.4] J/k_B$.

Important to note is that running enough MCMC cycles over the entirety of the temperature range, in addition to choosing a small enough temperature step dT that ensures enough accuracy, would take a very long time. Moreover, through trial and error we find that only a small part of the temperature range actually holds the points that indicate a phase transition. In other words, running a high number of MCMC cycles over the entire range, while enforcing a small enough dT, is unnecessary. We choose to instead make use of a large scan with fewer MCMC cycles over the entire temperature range to determine an area of interest, and then a fine-grained scan with additional MCMC cycles over the area of interest - which we have set to $T_{\text{interest}} \in [2.26,\ 2.34]$. For the large scan we run 10^5 MCMC cycles with a temperature step $dT = 0.03$, and determine the points that indicate a phase transition. After determining the area of interest, a fine-grained scan is then performed with 10^6 MCMC cycles, with a smaller temperature step $dT = 0.005$. We run both large and fine-grained scans for the different lattice sizes.

To determine the critical temperature, we use the results from the large and fine-grained scans - in particular the magnetic susceptibility, χ. As it boasts having the most distinct peaks, it makes for a particularly well-suited candidate for determining the critical temperatures. With our derived values for T_c, we then attempt to numerically estimate $T_c(L = \infty)$ and compare it to Onsager's analytical result. This is done by performing a linear regression on our set of T_c and L values, an expression for which is found by rewriting

equation (15) to:

$$T_c(L) = aL^{-1} + T_c(L = \infty) \tag{25}$$

where $T_c(L = \infty)$ now corresponds to the intersection point where $1/L = 0 \implies L = \infty$.

D. Tools

We present here the tools we use to perform the simulations and to present our results.

The simulations are made on two computers:

- A Lenovo Yoga Slim 7, with an AMD Ryzen 7 4700U processor, using Windows Subsystem for Linux with 14 GB of assigned RAM and 8 threads.
- A Macbook Air M1, with an Apple M1-chip with 8 GB RAM and 8 threads.

All simulations give the same results, except the time-measurements regarding the parallelization described in section II C. These measurements are done on the Macbook Air M1.
The algorithm is implemented using C++. To generate random numbers we use the `std::mt19937` Mersenne Twister generator from the `random` library. We use the seed 666 for all simulations. We use the `armadillo` library to create the spin lattice-matrix and to save our results in matrices and vectors. To parallelize our numerical algorithm code we use `OpenMP`.
The results from the simulations are processed and visualized using Python. To extract the results from the `armadillo`-matrices we use the `pyarma`-library. We also use the `numpy` library to process data-arrays. All illustrations are made with the library `matplotlib`. The linear fit to the computed critical temperatures, described in section II C, are made using `scipy.stats.linregress`.

III. RESULTS AND DISCUSSION

In this section we present and discuss the results from the simulation of the 2D Ising model using the MCMC Metropolis algorithm. For the case where $L = 2$, we compare the results from the algorithm versus the analytical solutions derived in the section II. We then use the results from lattices of size $L = 20$ to discuss and determine the burn-in, as well as estimate the probability distributions for the normalized energy. We look at the effects of parallelizing the algorithm, after which we use the parallelized code to simulate lattices of sizes up to $L = 100$. The results from these simulations are used to derive an estimate of $T_c(L = \infty)$, which is then compared to Onsager's analytical result (16).

To test our algorithm we start by comparing our numerical simulation for $L = 2$ at $T = 1\ J/k_B$ with the analytical solution presented in section II A. After 10^6 MCMC cycles we get the results presented in table II. From this we see that the relative error of the expectation values of the normalized energy and the normalized magnetization is of order 10^{-5}, while the error for the normalized heat capacity and normalized magnetic susceptibility are greater, being of magnitude 10^{-2}. The reason for this difference can be found by looking at how the numerical estimates change, depending on the number of cycles. In FIG IIIa and FIG IIIb we see that $\langle\epsilon\rangle$ and $\langle|m|\rangle$ are relatively close to the analytic solution during the first 100 cycles. In comparison, we see from FIG IIIa and FIG IIIb that C_V and χ are relatively further away from the analytic solution during the same first 100 cycles. This is explicitly shown in table III where we have listed the numerical estimates and the relative errors after 100 cycles. From this we see that the relative error of C_V and χ after 100 cycles lies around 1350 %, while the relative error of $\langle\epsilon\rangle$ and $\langle|m|\rangle$ are below 3 %. The estimates for $\langle\epsilon\rangle$ and $\langle|m|\rangle$ after 100 cycles are actually better than the final estimate for C_V and χ after 10^6 cycles. However, from FIG III and III we see that all four quantities seems to reach some sort of steady state after 10^4 cycles. As such, a simulation of 10^4 cycles should therefore give results that are consistent with the analytical solutions. But because the estimates of C_V and χ are relatively far away from the analytical solutions during the first cycles of the simulation, the final calculated values after 10^6 cycles seems to have been "pulled" away in a less accurate direction. This is related to the effect of a burn-in time, discussed in section II C.

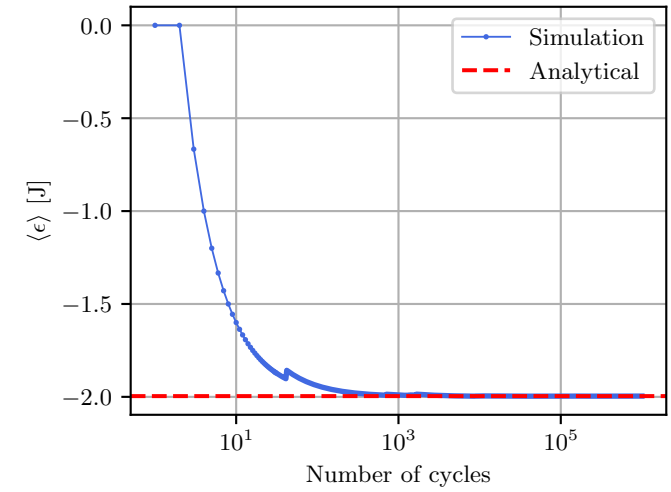
TABLE II. Comparison of the numerical and analytic results for the 2×2 Ising model at $T = 1\ J/k_B$ after 10^6 MCMC cycles starting from a randomized spin lattice. The analytic solutions are found from (9), (10), (13) and (14).

Quantity	Analytic	Simulation	Relative error
$\langle\epsilon\rangle$	-1.9960 J/k_B	-1.9959 J/k_B	0.006 %
$\langle\|m\|\rangle$	0.9987	0.9986	0.004 %
C_V	0.0321 k_B	0.0331 k_B	3.120 %
χ	0.0040 $1/J$	0.0031 $1/J$	3.307 %

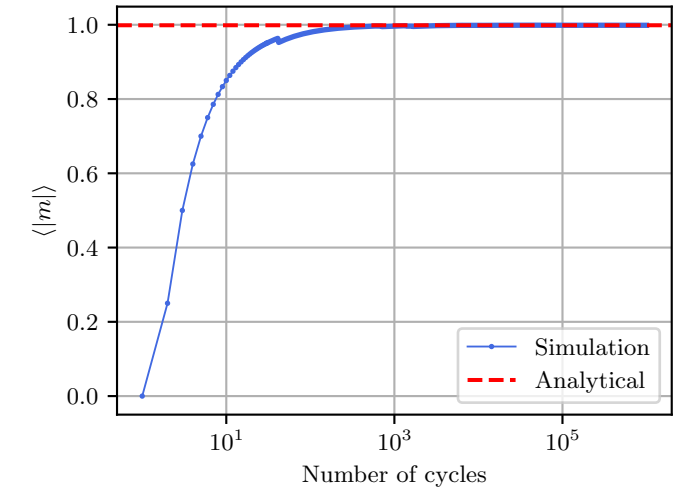
TABLE III. Comparison of the numerical and analytic results for the 2×2 Ising model at $T = 1\ J/k_B$ after 10^2 MCMC cycles starting from a randomized spin lattice. The analytic solutions are found from (9), (10), (13) and (14).

Quantity	Analytic	Simulation	Relative error
$\langle\epsilon\rangle$	-1.9960 J/k_B	-1.9400 J/k_B	2.8 %
$\langle\|m\|\rangle$	0.9987	0.9800	1.9 %
C_V	0.0321 k_B	0.4656 k_B	1351.3 %
χ	0.0040 $1/J$	0.0584 $1/J$	1356.1 %

To further investigate the effect of burn-in, we look at

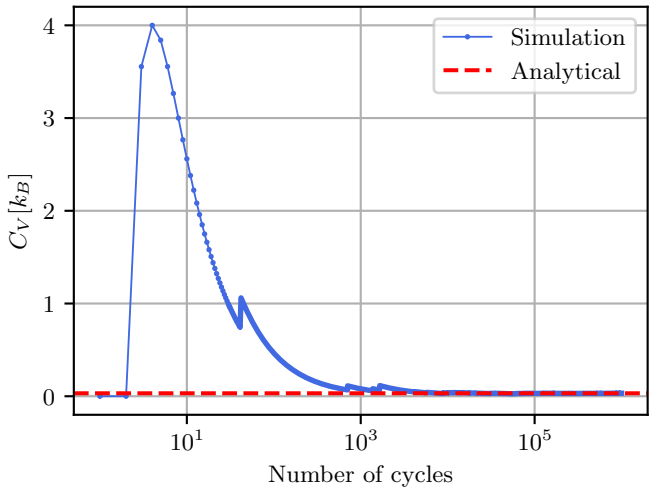


(a)

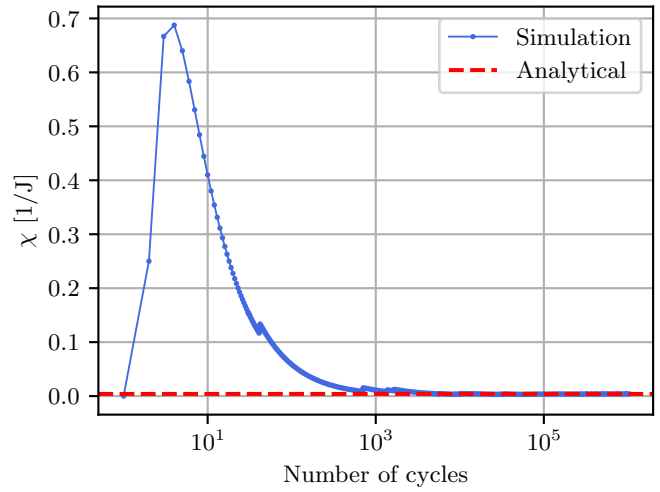


(b)

FIG. 4. The evolution of the expectation value of normalized energy (a) and the expectation value of normalized magnetization (b) during 10^6 MCMC cycles for a lattice with size $L = 2$ at $T = 1.0\ J/k_B$ starting from a random spin configuration. The simulation is compared with the analytical solution found from section II A and listed in TABLE II.



(a)



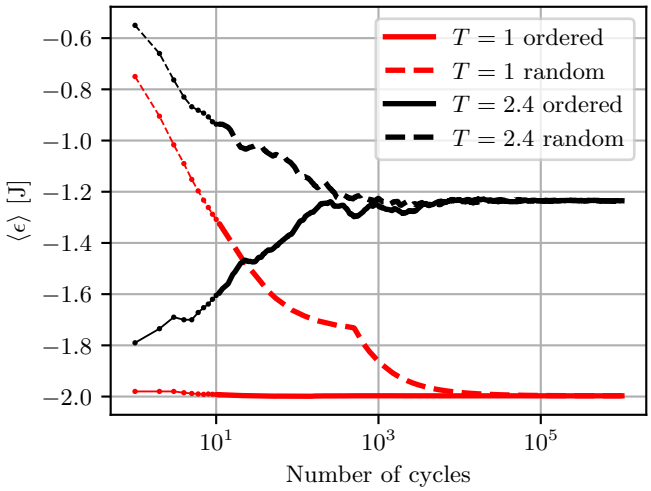
(b)

FIG. 5. The evolution of the normalized heat capacity (a) and normalized magnetic susceptibility (b) during 10^6 MCMC cycles for a lattice with size $L = 2$ at $T = 1.0\ J/k_B$ starting from a random spin configuration. The simulation is compared with the analytical solution found from section II A and listed in TABLE II.

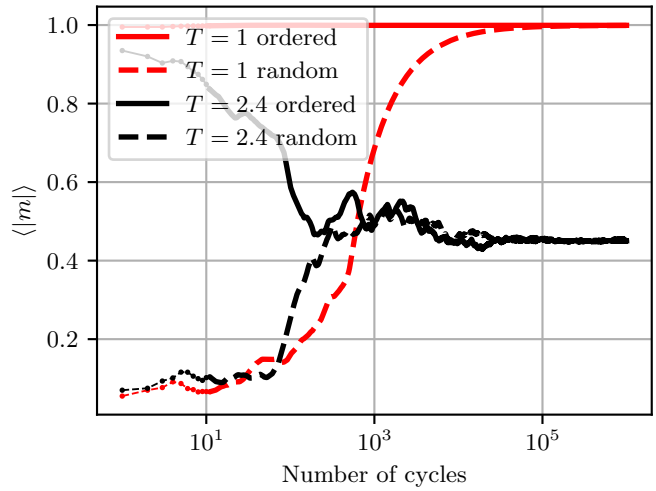
how the estimates of $\langle\epsilon\rangle$ and $\langle|m|\rangle$ evolves with the number of MCMC cycles for a lattice with size $L = 20$. In FIG III and FIG III we show how these quantities evolve for two different temperatures, starting from either an ordered or random initial state. From this we see that the lattice at $T = 1.0\ J/k_B$ starting from an ordered state seems to be stabilizing after only about 10 cycles, while the random initial state needs about 10^4 cycles to stabilize. As the ordered initial state is already quite close to the stabilized values before any MCMC cycles are made, it would be more favorable to use the ordered state when calculating quantities at this temperature, as it requires fewer cycles.

For $T = 2.4\ J/k_B$ both the ordered and random initial states are quite far away from the stabilized values, and they both need about 10^4 cycles to stabilize. In this particular case the choice of an ordered or random state is rather arbitrary.

For these two states at $T = 2.4\ J/k_B$ and the random state at $T = 1.0\ J/k_B$, it could be tempting to set the burn-in time at 10^4 cycles. However, as FIG III only shows how the *expectation values* evolve with the number of cycles, it is still unsure whether the actual states evolve during the simulation.



(a)



(b)

FIG. 6. The evolution of the expectation value of the normalized energy (a) and normalized magnetization (b) during 10^6 MCMC cycles from four different simulations of a lattice with size $L = 20$. For two temperatures $T = 1.0\ J/k_B$ and $T = 2.4\ J/k_B$ we have started from both an ordered (all spins are $\uparrow$) and a random spin configuration.

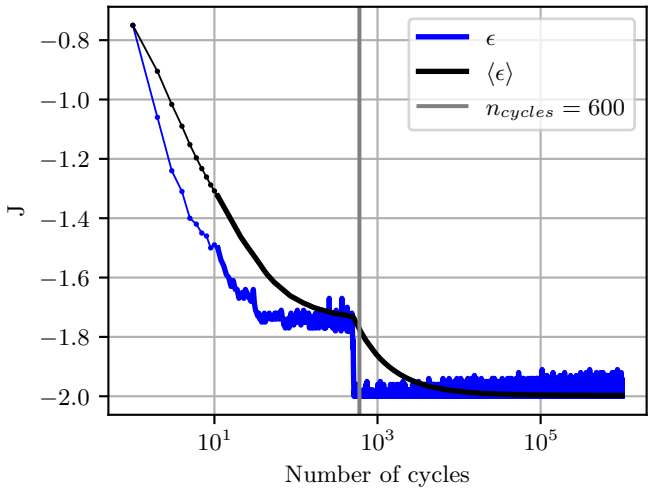


FIG. 7. A comparison of the evolution of the normalized energy and the expectation value of the normalized energy during 10^6 MCMC cycles of a lattice with size $L = 20$ at $T = 1.0\ J/k_B$ starting from a random spin configuration. We have also included a line at 600 cycles which suggest the burn-in time.

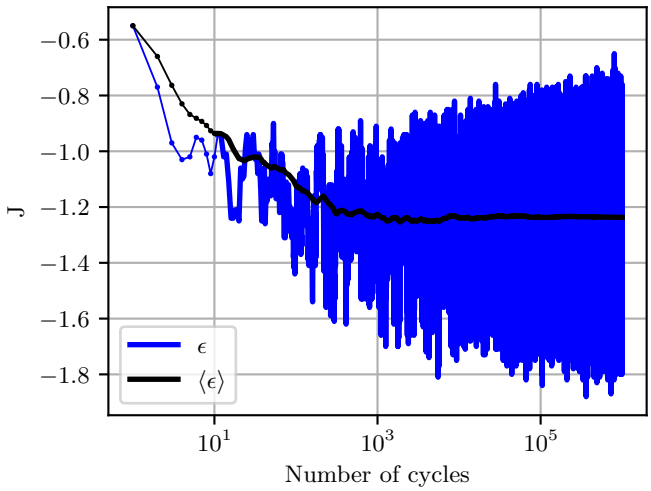


FIG. 8. A comparison of the evolution of the normalized energy and the expectation value of the normalized energy during 10^6 MCMC cycles of a lattice with size $L = 20$ at $T = 2.4\ J/k_B$ starting from a random spin configuration.

In FIG 7 we show the normalized energy at each cycle for the random initial state at $T = 1.0\ J/k_B$. From this we see that the *actual* time for the simulation to stabilize is about 600 cycles, and not 10^4 as one could think from form FIG III. Because ϵ takes values far away from the stabilized value before 600 cycles, the average $\langle\epsilon\rangle$ needs more cycles to stabilize.

In FIG 8 we now show how ϵ changes for the random initial state at $T = 2.4\ J/k_B$. Here it does not seem that we reach an equilibrium far away from the first number of cycles. In fact, it only seems like the first value is not among the values we would expect during the simulation with 10^6 cycles. But with this big of variance in values of ϵ we will expect to need many samples to get a good estimate for $\langle\epsilon\rangle$, which agrees with the evolution in FIG III.

As $|m|$ (5) in similar way as ϵ only depends on the current state of the lattice, we would expect to see similar results when studying this quantity as a function of cycles. This could be an aspect that should be investigated in further studies.

From the analysis above, we can therefore say that the burn-in time for these lattices seems to be a lot less than 10^4. However, using too large of a burn-in time should not be too big of a problem, as not including a burn-in time would leave us in an equally bad starting position. Throwing away 10^4 perfectly good samples does however increase the total run-time for the algorithm to reach good values. But in the case of running 10^6 MCMC cycles, as we do for most of our simulations, doing an extra 1 % of calculations to go through the burn-in time, should not increase the run-time by much. For the purposes of this study, using a burn-in time of 10^4 should therefore be adequate.

By using the energies shown in FIG 7 and 8, we create normalized histograms to show the probability distribution $p_\epsilon(\epsilon;T)$ for the two temperatures. In FIG IVa these two histograms are made by using a burn-in time of 10^4. From this we see that the distributions are centered around the expected values $\langle\epsilon\rangle \approx -2.0\ J/k_B$ for $T = 1.0\ J/k_B$ seen from FIG 7, and $\langle\epsilon\rangle \approx -1.2\ J/k_B$ for $T = 2.4\ J/k_B$ seen from FIG 8.
The variances of ϵ between FIG 7 and 8 also predicts the difference in variance between the two histograms. In FIG IVa we can only see energy values in the range $\epsilon \approx \{-2.0, -1.9\}$ for $T = 1.0\ J/k_B$, while for $T = 2.4\ J/k_B$ we see values in the range $\epsilon \approx \{-1.8, -0.6\}$. The values of ϵ are also more spread out for $T = 2.4\ J/k_B$ compared to the sharp peak for $T = 1.0\ J/k_B$. This is as expected from the theory of phase-transitions, which is briefly explained in section II A. For $T = 1.0\ J/k_B$ the system should be well below the critical temperature (16), which means that we should be seeing ferromagnetic behavior, i.e. parallel aligning spins. This means that there is a clearly preferred energy associated with the ordered spin-configuration, causing energies from less ordered spins to be less likely.
For $T = 2.4\ J/k_B$, we expect the system to be around the critical temperature. Here the system loses its ferromagnetic properties, and the aligned spin states are now longer the only preferred states. With an increased temperature, thermal fluctuations increase [3] causing the system to faster experience change in energy. In the Metropolis algorithm this effect is seen from the acceptance rule 18 and the Boltzmann factor 7, where a higher temperature causes the Boltzmann factor to increase, which again raises the probability of the state to change. More changes of states with different energy gives a greater variance in the probability distribution $p_\epsilon(\epsilon;T)$ for $T = 2.4\ J/k_B$ than $T = 1.0\ J/k_B$.
This difference in variance could cause different burn-in times for the two temperatures, as discussed previously in section III. From the results shown in FIG 7 and 8, we estimated a burn in time of about 600 MCMC cycles for $T = 1.0\ J/k_B$ and 1 for $T = 2.4\ J/k_B$. A histogram made with these burn-in times is shown in FIG IVb. Using these burn-in times gives similar results as for using the burn-in time of 10^4 MCMC in FIG IVa, as we predicted earlier in this section during the discussion of estimating the burn-in. To emphasize how similar the results are when using a burn-in time of 10^4 MCMC cycles or 1 and 600 MCMC cycles, we show in FIG IVc the resulting histogram when using zero burn-in time. Here we see a quite different probability distribution. For $T = 1.0\ J/k_B$ we recognize the higher energy values from FIG 7 which illustrates the transition from the higher energy states towards the stabilization in the lower energy states. For $T = 2.4\ J/k_B$ we recognize the highest energy-value just below -5 J from FIG 8.
These results makes it clear that using a burn-in time of 10^4 seems well beyond what is necessary for a lattice of size $L = 20$. Further studies should investigate the burn-in times for lattices of greater sizes. For now, we use a burn-in time of 10^4 MCMC cycles before running 10^6 MCMC cycles for all lattice-sizes, as this is not substantially more computationally demanding than not including the burn-in.

When applying MCMC to an Ising model consisting of large lattice sizes L, the duration of the run-times can become unreasonably long rather quickly, as one MCMC cycle corresponds to N spin flips. As such, the necessity of a method to shorten down the run-times, i.e. parallelization comes into play. Our approach when it came to the parallelization of the code was to parallelize the loops over the temperatures, as this was the least complicated way of quickening the run-times. The approach was applied using the `OpenMP`-library, after which a series of timing tests were conducted to determine the resulting speed-up factor from parallellization.
Plotting the data from the timing tests, yielded FIG 9.

TABLE IV. Results from timing-tests after parallelization, for $n_{threads} = 1, 2, 4, 6, 8, 10$. For the timing-tests, a lattice of size $L = 20$ was used, for 11 temperatures in the range $T \in [2.1, 2.4]\ J/kB$ with 10^5 MCMC cycles per temperature. Tests were performed on an M1 mac.

Number of threads	Time measurements [s]			$\bar{t}$ [s]
1	31.73	31.06	31.03	31.27
2	16.45	16.40	16.45	16.43
4	9.72	9.71	9.94	9.79
6	7.43	7.24	7.55	7.41
8	7.02	6.89	7.71	7.21
10	7.43	7.51	7.40	7.45

A noticeable decrease in the elapsed run-time is seen for the first three increases of the number of threads, where the run-time almost seems to halve each time. We then turn our focus to the speed-up factor, see equation (24). In an ideal scenario

$$t_{\text{n threads}} \Leftrightarrow 1/t_{\text{n threads}} \Leftrightarrow f_{\text{speed up}} = n_{\text{threads}},$$

however in most cases ideal speed-up is not achieved - as is made apparent by the resulting speed-up factors presented in TABLE V. Further, we can also see that

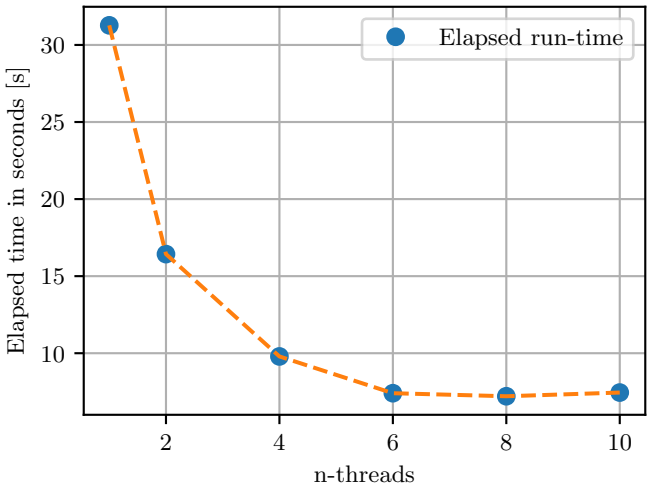


FIG. 9. Timing test for the parallelization over temperature values of our MCMC Metropolis algorithm for different choices of number of threads. The simulations have been done for 11 temperatures with 10^5 MCMC cycles per temperature in a lattice with size $L = 20$, using an M1 mac.

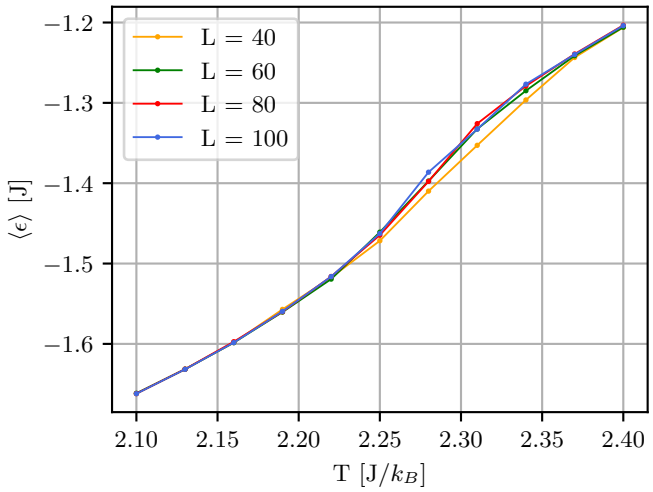


FIG. 10. The expectation value of the normalized energy for different lattice sizes in a large temperature domain after simulations of 10^6 MCMC cycles starting from a random spin configuration. We have used a temperature-step of $\Delta T = 0.03$ J/k_B.

speed-up factors only increase with the number of threads spawned up until certain point, where it seems to taper off. A suspicion here is that our approach's upper bound for speed-up has been met, and that any more threads spawned will only contribute to the amount of overhead in our procedure; i.e. computational work required by our system beyond what is directly necessary. Parallel threads often need to synchronize their execution to avoid any errors or data conflicts, increasing the number of threads can subsequently cause it to need synchronization more frequently. As such, 8 threads are used for all cases where the code needs parallelization.

With the parallelization in order, we then seek to

TABLE V. Resulting speed-up factors, $f_{\text{speed-up}}$, for $n_{threads} = 1, 2, 4, 6, 8, 10$. Equation (24) was used, with the averaged times in TABLE IV $t_1 = 31.27$, i.e. elapsed run-time with no parallellization.

Number of threads	$f_{\text{speed-up}}$
1	1.0
2	1.90
4	3.20
6	4.22
8	4.32
10	4.20

investigate phase transitions and the behavior of the model for larges lattice sizes. Performing a large scan over the entire temperature range $T_{\text{interest}} \in [2.1,\ 2.4]$, yields the results shown in FIG 10, 11, 12 and 13. Upon observing the behavior of $\langle |m| \rangle$ in FIG 11, we

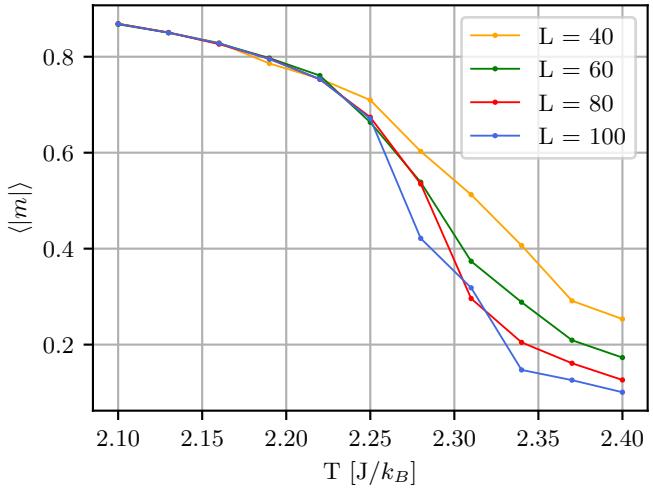


FIG. 11. The expectation value of the normalized magnetization for different lattice sizes in a large temperature domain after simulations of 10^6 MCMC cycles starting from a random spin configuration. We have used a temperature-step of $\Delta T = 0.03$ J/k_B.

see that it quickly tends toward 0 - which indicates that the ferromagnet is transitioning to a paramagnet. This is also evident from the behavior of C_V and χ, see subsection II A, as they near the area of interest. Further, by performing a fine-grained scan of the area of interest and using the results for χ seen in FIG 14, we determine T_c for each of the lattice sizes - see TABLE VI.

Now we turn our focus towards numerically estimating $T_c(L = \infty)$. A linear regression is first performed on our

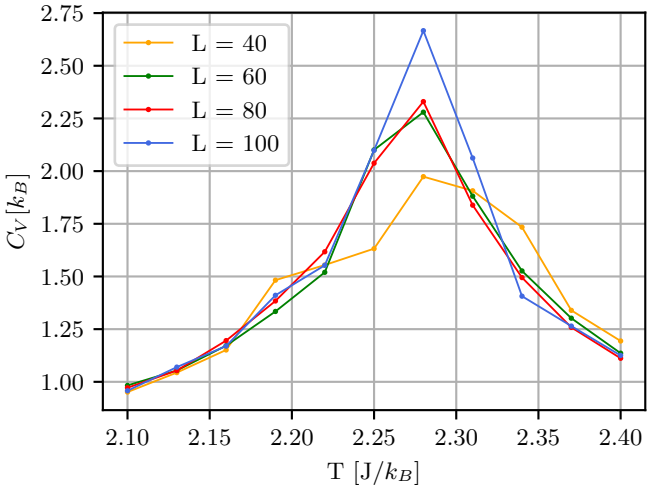


FIG. 12. The expectation value of the normalized heat capacity for different lattice sizes in a large temperature domain after simulations of 10^6 MCMC cycles starting from a random spin configuration. We have used a temperature-step of $\Delta T = 0.03\ J/k_B$.

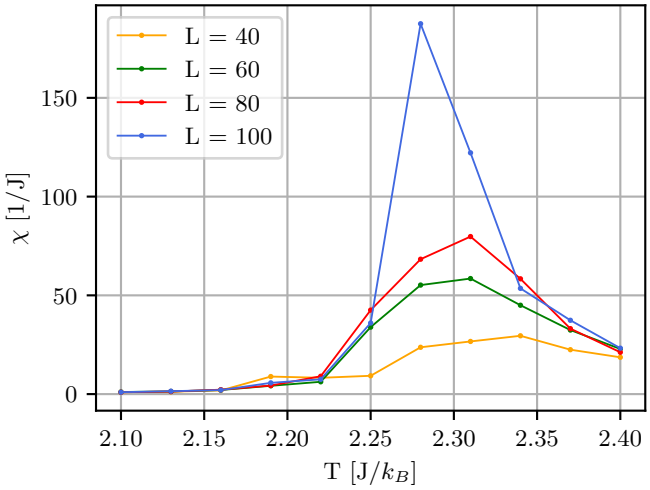


FIG. 13. The expectation value of the normalized magnetic susceptibility for different lattice sizes in a large temperature domain after simulations of 10^6 MCMC cycles starting from a random spin configuration. We have used a temperature-step of $\Delta T = 0.03\ J/k_B$.

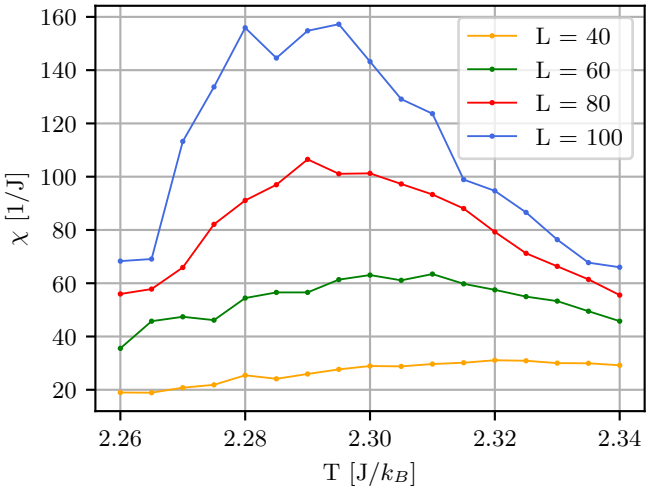


FIG. 14. The expectation value of the normalized magnetic susceptibility for different lattice sizes in a smaller temperature domain after simulations of 10^6 MCMC cycles starting from a random spin configuration. We have used a temperature-step of $\Delta T = 0.005\ J/k_B$.

TABLE VI. Resulting T_c for $L = 40, 60, 80, 100$, from a fine-grained scan of $T \in [2.26, 2.34]$ with 10^6 MCMC cycles, and $dT = 0.005$.

L	$T_c\ [J/k_b]$
40	2.32
60	2.31
80	2.29
100	2.28

TABLE VII. Resulting estimates for $T_c(L = \infty)$ from linear regression, along with relative error (RE) and standard error (SE), when compared to Onsager's analytical result. $T_c(L = \infty)$ estimate for 4 points is noticeably accurate, at the cost of SE. While for 3 points, SE is noticeably reduced at the cost of accuracy and RE.

N points	Est. $T_c(L = \infty)\ [J/k_B]$	RE [%]	SE [%]
4	2.258	0.5	62
3	2.234	1.5	21

entire set of L and T_c, and for the set without the entry for $L = 40$. The fitted line from the linear regression is given by equation (25) and illustrated in FIG 15. The resulting estimates for each linear regression is found in TABLE VII. Here we see that the linear regression for the entire set of T_c boasts a rather accurate numerical estimate, $T_c(L = \infty) \approx 2.258\ J/k_B$. Compared to Onsager's analytical result of $T_c(L = \infty) \approx 2.269\ J/k_B$, it yields a RE of 0.5%. However, it is also exhibits a much larger SE. In other words, the average deviation of the observed data points from the predicted values on the fitted line is quite high. For the case where we discard the entry for $L = 40$, as seen in FIG 16, we see that the exhibited SE is considerably reduced. However, this reduction comes at the expense of a decrease in the accuracy of the numerical estimate, and an increase in the RE. This suggests, of course, that there is some degree of inaccuracy in our measurements of $T_c(L)$ for the different lattice sizes. Additionally, the Metropolis algorithm seems to lose its efficiency close to the T_c [2]. A solution to this would be to employ a different

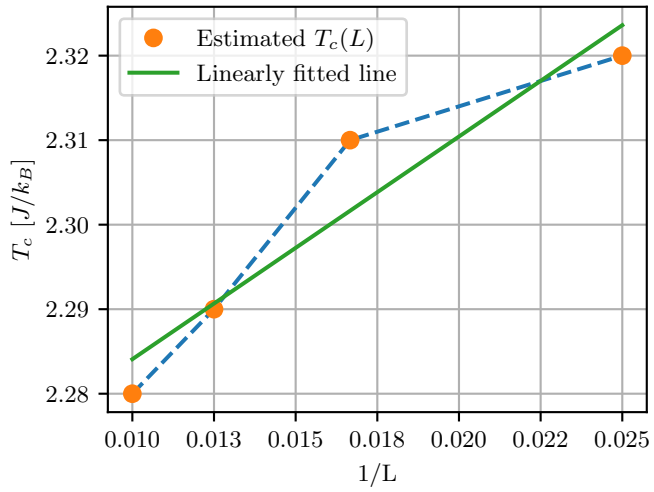


FIG. 15. The resulting linear model made using `scipy.stats.linregress` and the critical temperatures listed in table VI. This model results in a critical temperature $T_c(L = \infty) = 258\ J/k_B$.

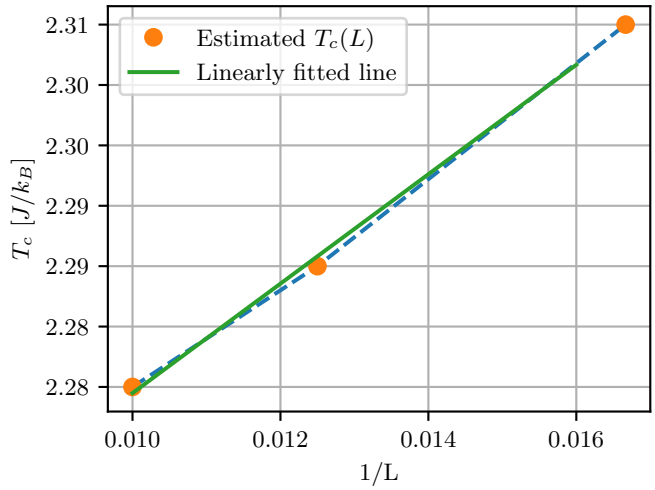


FIG. 16. The resulting linear model made using `scipy.stats.linregress` and three of the critical temperatures listed in table VI. This model results in a critical temperature $T_c(L = \infty) = 234\ J/k_B$.

algorithm that offer a more robust simulation near T_c, e.g. the heat bath algorithm, the Wolff algorithm, etc. [4][5].

IV. CONCLUSION

Having simulated the two-dimensional Ising model using the MCMC method, we have gained valuable insights into the temperature-dependent behavior of ferromagnets. Numerous MCMC cycles were performed for different temperatures, and various thermodynamic quantities of interest were computed. For a temperature of $T = 1\ J/k_B$, different choices of MCMC cycles were ran to determine the necessary amount of cycles for the computed results to be consistent with analytical values. We found that for 10^4 MCMC cycles, the computed results converged toward the analytical values. After 10^4 cycles the RE for our results for $\langle\epsilon\rangle$ and $\langle|m|\rangle$, were 0.006% and 0.004% respectively. While for our results for C_V and χ, they were 3.120% and 3.307%. We then moved on to a larger lattice size $L = 20$, and studied the importance of a burn-in time to improve the accuracy of our computed results. Eventually it was determined that a suitable burn-in time of the simulation of the Ising model would be 10^4.

After testing our model for a larger lattice size, we then went beyond expectation values to estimating the full PDF for ϵ. A normalized histogram was made for two temperatures, $T = 1.0\ J/k_B$ and $T = 2.4\ J/k_B$, where we observed that the variance was higher for the higher temperature. This meant that there was a preferred energy associated with the ordered spin-configuration, causing the energies from less ordered spins to be less likely. Additionally this difference in variance also necessitated the need for different burn-in times for the two temperatures; these were set to 660 MCMC cycles for $T = 1\ J/k_B$, and 1 MCMC cycle for $T = 2.4\ J/k_B$. Upon comparison with the previous burn-in time, we found that 10^4 MCMC cycles is well beyond what is necessary for a lattice of size $L = 20$.

Further, we also wanted to explore the effects of larger choices of lattice size L, and also finally investigate the phase behavior of the model. Since simulating the model for larger choices of L can take unreasonably long, the code was parallelized. A series of timing-tests were then ran to determine the speed-up factor from the parallelization, where the highest speed-up factor achieved was $f_{\text{speed-up}} = 4.32$. We also concluded that any more than 8 threads would have an adverse effect, subsequently reducing $f_{\text{speed-up}}$.

Finally, we investigated the phase transitions in the system. By letting the model evolve over a set temperature range for $L = 40, 60, 80, 100$ we were able to determine an area of interest, which in this case we set to $T \in [2.26, 2.34]\ J/k_B$. Having found the point where the ferromagnet transitioned into its paramagnetic phase, we then determined the critical temperatures $T_c(L)$ for each lattice size. We found that for $L = 40$, the model yielded a $T_c(L) = 2.32\ J/k_B$, for $L = 60$ a $T_c(L) = 2.31\ J/k_B$, for $L = 80$ a $T_c(L) = 2.29\ J/k_B$, and for $L = 100$ a $T_c(L) = 2.28\ J/k_B$. Using our set of $T_c(L)$ and L values, we then derived a numerical estimate of $T_c(L = \infty)$ using linear regression, and compared it to Onsager's analytical result. We found that when using the entire data set our estimate was consistent with

Onsager's result, with $RE = 0.5\%$, but the data points deviated from the predicted values of the fitted line. When instead only using the the first three points of our data set, we found that the deviation of the data points from the fitted line was far less, at the cost of the estimated value being farther away from Onsager's analytical result. We concluded that our method for finding the $T_c(L = \infty)$ is reliable, but that our method for determining the critical temperatures for each lattice size introduced inaccuracies.

To improve the accuracy of the computations, several things can be investigated or implemented. If we were to run the simulations on an even larger scale, for more MCMC cycles and a smaller temperature step dT, this would enable us to more accurately determine the critical temperatures for each lattice size. Adding a more robust algorithm that handle simulations near the critical temperature of the model could also contribute towards the accuracy of computed results.

APPENDIX A - DERIVING THE PARTITION FUNCTION Z AND EXPECTATION VALUES FOR TWO-DIMENSIONAL ISING MODEL WITH $L = 2$

$$\begin{aligned} Z &= \sum_{\text{all possible } \mathbf{s}} e^{-\beta E(\mathbf{s})} \\ &= e^{8J\beta} + 4 + 2e^{-8J\beta} + 4 + 4 + e^{8J\beta} \\ &= 2e^{8J\beta} + 2e^{-8J\beta} + 12 \\ &= 4\cosh(8J\beta) + 12 \end{aligned}$$

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

$$\begin{aligned} \langle \epsilon^2 \rangle &= \sum_{\mathbf{s}} \epsilon^2(\mathbf{s}) p(\mathbf{s}) \\ &= \sum_{\mathbf{s}} \frac{E^2(\mathbf{s})}{N^2} \frac{e^{-\beta E(\mathbf{s})}}{Z} \\ &= \frac{J^2}{16Z} \left(64e^{8J\beta} + 2 \cdot 64e^{-8J\beta} + 64e^{8J\beta} \right) \\ &= \frac{8J^2}{Z} \left(e^{8J\beta} + e^{-8J\beta} \right) \\ &= \frac{16J^2}{Z} \cosh(8J\beta) \end{aligned}$$

$$
\begin{aligned}
\sigma_E^2 &= \langle E^2\rangle - \langle E\rangle^2 \\
&= N^2\left(\langle \epsilon^2\rangle - \langle \epsilon\rangle^2\right) \\
&= N^2\left(\frac{16J^2}{Z}\cosh(8J\beta) - \frac{64J^2}{Z^2}\sinh^2(8J\beta)\right) \\
&= \frac{16N^2J^2}{Z^2}\left(Z\cosh(8J\beta) - 4\sinh^2(8J\beta)\right) \\
&= \frac{16N^2J^2}{Z^2}\left(12\cosh(8J\beta) + 4\cosh^2(8J\beta) - 4\sinh^2(8J\beta)\right) \\
&= \frac{64N^2J^2}{Z^2}\left(3\cosh(8J\beta) + 1\right)
\end{aligned}
$$

$$
\begin{aligned}
C_V &= \frac{1}{N}\frac{1}{k_BT^2}(\langle E^2\rangle - \langle E\rangle^2) \\
&= \frac{256J^2}{Z^2}\frac{1}{k_BT^2}(3\cosh(8J\beta) + 1)
\end{aligned}
$$

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

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

$$
\begin{aligned}
\sigma_M^2 &= \langle M^2\rangle - \langle |M|\rangle^2 \\
&= N^2\left(\langle m^2\rangle - \langle |m|\rangle^2\right) \\
&= N^2\left(\frac{2e^{8J\beta}+2}{Z} - \left(\frac{2e^{8J\beta}+4}{Z}\right)^2\right) \\
&= \frac{N^2}{Z^2}\Big(Z(2e^{8J\beta}+2) - (4e^{16J\beta} + 16e^{8J\beta} + 16)\Big) \\
&= \frac{N^2}{Z^2}\Big((2e^{8J\beta} + 2e^{-8J\beta} + 12)(2e^{8J\beta}+2) - 4e^{16J\beta} - 16e^{8J\beta} - 16\Big) \\
&= \frac{N^2}{Z^2}\left(4e^{-8J\beta} + 26e^{8J\beta} + 4e^{16J\beta} + 28 - 4e^{16J\beta} - 16e^{8J\beta} - 16\right) \\
&= \frac{4N^2}{Z^2}(e^{-8J\beta} + 3e^{8J\beta} + 3)
\end{aligned}
$$

$$
\begin{aligned}
\chi &= \frac{1}{N}\frac{1}{k_BT}(\langle M^2\rangle - \langle |M|\rangle^2) \\
&= \frac{16}{Z^2}\frac{1}{k_BT}(e^{-8J\beta} + 3e^{8J\beta} + 3)
\end{aligned}
$$

APPENDIX B - OVERVIEW OF ACCESSIBLE MICROSTATES FOR 2×2 GRID, AND CORRESPONDING ENERGY AND MAGNETIZATION

$$\epsilon = -8\,J, M = -4$$

$$\begin{bmatrix} \downarrow & \downarrow \\ \downarrow & \downarrow \end{bmatrix}$$

$$\epsilon = 0\,J, M = -2$$

$$\begin{bmatrix} \uparrow & \downarrow \\ \downarrow & \downarrow \end{bmatrix} \begin{bmatrix} \downarrow & \uparrow \\ \downarrow & \downarrow \end{bmatrix} \begin{bmatrix} \downarrow & \downarrow \\ \uparrow & \downarrow \end{bmatrix} \begin{bmatrix} \downarrow & \downarrow \\ \downarrow & \uparrow \end{bmatrix}$$

$$\epsilon = 8\,J, M = 0$$

$$\begin{bmatrix} \uparrow & \downarrow \\ \downarrow & \uparrow \end{bmatrix} \begin{bmatrix} \downarrow & \uparrow \\ \uparrow & \downarrow \end{bmatrix}$$

$$\epsilon = 0\,J, M = 0$$

$$\begin{bmatrix} \downarrow & \uparrow \\ \downarrow & \uparrow \end{bmatrix} \begin{bmatrix} \downarrow & \downarrow \\ \uparrow & \uparrow \end{bmatrix} \begin{bmatrix} \uparrow & \uparrow \\ \downarrow & \downarrow \end{bmatrix} \begin{bmatrix} \uparrow & \downarrow \\ \uparrow & \downarrow \end{bmatrix}$$

$$\epsilon = 0\,J, M = 2$$

$$\begin{bmatrix} \uparrow & \uparrow \\ \downarrow & \uparrow \end{bmatrix} \begin{bmatrix} \downarrow & \uparrow \\ \uparrow & \uparrow \end{bmatrix} \begin{bmatrix} \uparrow & \downarrow \\ \uparrow & \uparrow \end{bmatrix} \begin{bmatrix} \uparrow & \uparrow \\ \uparrow & \downarrow \end{bmatrix}$$

$$\epsilon = -8\,J, M = 4$$

$$\begin{bmatrix} \uparrow & \uparrow \\ \uparrow & \uparrow \end{bmatrix}$$

APPENDIX C - OVERVIEW OF POSSIBLE VALUES FOR ΔE WHEN FLIPPING A SPIN

When flipping a spin only a limited set of values are available for ΔE. As the effect of flipping as spin in a lattice with size $L > 2$, is the same as flipping a spin in a 3×3 domain, as shown in FIG 1, we can limit ourselves to showing the possible energy-changes in a lattice of this size.

$$E = -4\,J \qquad \begin{matrix} & \uparrow & \\ \uparrow & \uparrow & \uparrow \\ & \uparrow & \end{matrix} \implies E = 4\,J \qquad \begin{matrix} & \uparrow & \\ \uparrow & \downarrow & \uparrow \\ & \uparrow & \end{matrix}$$

For $\Delta E = 4\,J$

$$E = -2\,J \qquad \begin{matrix} & \uparrow & \\ \downarrow & \uparrow & \uparrow \\ & \uparrow & \end{matrix} \implies E = 2\,J \qquad \begin{matrix} & \uparrow & \\ \downarrow & \downarrow & \uparrow \\ & \uparrow & \end{matrix}$$

For $\Delta E = 0\,J$

$$E = 0\,J \qquad \begin{matrix} & \uparrow & \\ \downarrow & \uparrow & \uparrow \\ & \downarrow & \end{matrix} \implies E = 0\,J \qquad \begin{matrix} & \uparrow & \\ \downarrow & \downarrow & \uparrow \\ & \downarrow & \end{matrix}$$

For $\Delta E = -4\,J$

$$E = 2\,J \qquad \begin{array}{ccc} & \downarrow & \\ \downarrow & \uparrow & \uparrow \\ & \downarrow & \end{array} \implies E = -2\,J \qquad \begin{array}{ccc} & \downarrow & \\ \downarrow & \downarrow & \uparrow \\ & \downarrow & \end{array}$$

For $\Delta E = -8\,J$

$$E = 4\,J \qquad \begin{array}{ccc} & \downarrow & \\ \downarrow & \uparrow & \downarrow \\ & \downarrow & \end{array} \implies E = -4\,J \qquad \begin{array}{ccc} & \downarrow & \\ \downarrow & \downarrow & \downarrow \\ & \downarrow & \end{array}$$

---

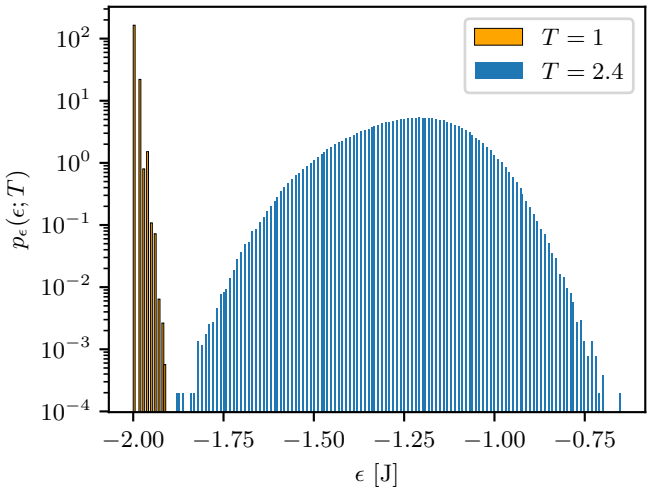
[1] Huang, C., Marian, J. (2016). *A generalized Ising model for studying alloy evolution under irradiation and its use in kinetic Monte Carlo simulations.* Journal of Physics Condensed Matter, 28(42), 425201. [https://escholarship.org/uc/item/6k06m5gtauthor][Visited 18.11.2023]

[2] Hjorth-Jensen, Morten. (August 2015) *Lecture Notes Fall 2015.* Published by Department of Physics, University of Oslo.

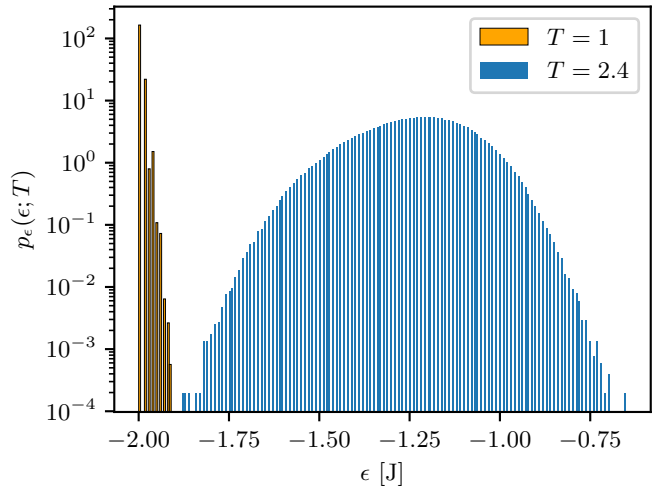
[3] Shroeder, Daniel V. (1995) *An Introduction to Thermal Physics (2021)* chapters 2, 6, 8. Oxford University press

[4] A.M. Ferrenberg, R.H. Swendsen, Phys. Rev. Lett. 61, 2635 (1988). DOI 10.1103/PhysRevLett.61.2635

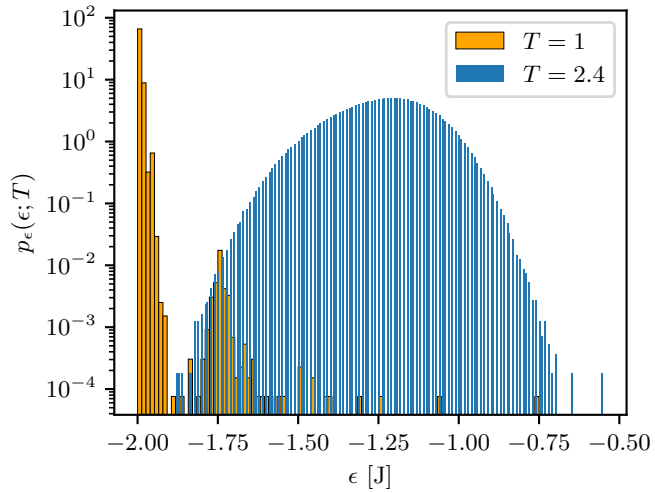
[5] A.M. Ferrenberg, R.H. Swendsen, Phys. Rev. Lett. 63, 1195 (1989). DOI 10.1103/PhysRevLett.63.1195



(a)



(b)



(c)

FIG. 17. The probability distributions of the normalized energies of a lattice with size $L = 20$ for temperatures $T = 1.0$ J/k_B and $T = 2.4$ J/k_B generated from randomized initial spin configuration. In (a) we have used a burn-in time of 10^4 MCMC cycles for both temperatures, in (b) we have used 600 cycles for $T = 1.0$ J/k_B and 1 cycle for $T = 2.4$ J/k_B and in (c) we have not included any burn-in time.