A study of phase transition in the 2D Ising model using Markov Chain Monte Carlo simulation

Alessio Canclini, Filip von der Lippe
(Dated: November 22, 2022)

We investigate phase transition in the widely studied Ising model and find a numerical approximation of the critical temperature at which an infinite lattice Ising model undergoes phase transition. This is done using a Markov Chain Monte Carlo (MCMC) approach. We approximate the expectation value for the mean energy ϵ and mean magnetization m as well as the specific heat capacity C_V and susceptibility χ. The MCMC method is first compared to analytical solutions for a 2×2 lattice and burn-in times are studied for a 20×20 lattice. Based on these results we choose to use 10^6 MCMC cycles for the following analysis since it appears to be the optimal compromise between precision and computational expense. Investigating the probability function $p_\epsilon(\epsilon; T)$ we find a variance of $0.000059J$ for $T = 1.0J/k_B$ and a larger variance of $0.020269J$ for $T = 2.4J/k_B$ indicating a greater probability of jumping to a higher energy state at the higher temperature. The Ising model simulation is then run for lattice sizes $40 \times 40, 60 \times 60, 80 \times 80$ and 100×100. The previously mentioned parameters are plotted against a temperature range of $T \in [2.1, 2.4]J/k_B$ for each lattice size. These results show indications of a phase transition taking place at a critical temperature T_c. These T_c points for different lattice sizes are then used to approximate the infinite lattice case through linear regression. Our final result is $T_{c,tot}(L = \infty) = 2.2686 \pm 0.0129 J/k_B$. Comparing this to Onsager's analytical result of $T_{c\ \text{analytic}}(L = \infty) \approx 2.269 J/k_B$ we see that the error between our numeric and the analytic result lies within our standard error. To be able to run the larger computations for reasonable runtimes the MCMC code has been parallelized using OpenMP achieving a speed-up factor of 3.92 for 8 threads on an 8 thread quad-core CPU.

I. INTRODUCTION

This article will explore temperature-dependent behavior in ferromagnetism using the two-dimensional **Ising model**. The main purpose of our simulations is to determine a numerical estimation of the **critical temperature** at which the system transitions from a magnetized to a non-magnetized phase. This is also called the Curie temperature [1].

With the Ising model we can model a magnetic materials' response to thermal energy. Spins in the model have two states, either up or down. These can be flipped by energy input (such as an increase in temperature) and each individual spin influences the total state of the system.

The Ising model is a widely studied model in statistical physics. Work on the model has shown to be useful for the analysis of several complex systems. Some examples are gasses sticking to solid surfaces or hemoglobin molecules that absorb oxygen [1]. Additionally, when it comes to studies of phase transition, theoretical approaches such as mean field models can actually predict wrong behavior, further motivating the use of the Ising model.[1]

We will simulate the Ising model using a Markov Chain Monte Carlo (MCMC) approach. After validating our numerical method we will finally approximate the critical temperature for an infinite ($L = \infty$) 2D Isisng model. The analytical solution to this problem was found by Lars Onsager in 1944 [2]. Thus, we can use Onsagers analytical result to evaluate the precision of the numerical MCMC method.

To allow for reasonable runtimes for the larger computations our MCMC code is parallelized using OpenMP.

Section II will cover the theoretical background of the Ising model and MCMC method as well as some analytical solutions for code testing. More comprehensive derivations of the analytical values can be found in appendix A and B.

In section III we present results from our Ising model simulations. These entail comparison between analytic and numerical results, an analysis of burn-in time for a larger lattice size of $L = 20$ and finally an investigation of phase transition.

A detailed discussion of the results and methods is then presented in section IV followed by a summary and concluding thoughts in section V.

II. METHODS

A GitHub repository containing code and more information on how to reproduce the results of this report can be found following this link: https://github.com/Fslippe/FYS4150/tree/main/project4

The square 2D lattices for our Ising model will have a length of L containing N spins with the relation $N = L^2$. Each spin s_i will have two possible states of

$$s_i = -1 \text{ or } s_i = +1.$$

The total spin state or **spin configuration** of a lattice will be represented as $\mathbf{s} = (s_1, s_2, ..., s_N)$. In its simplest form the total energy of the system is expressed as

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

Here $\mathscr{B}$ is an external magnetic field. Since we will be looking at the Ising model without an external magnetic field the equation will be simplified to

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

where $\langle kl \rangle$ denotes the sum going over all *neighboring pairs* of spins avoiding double-counting. J is the **coupling constant** simply setting the energy associated with spin interactions. *Periodic boundary conditions* will be implemented allowing all spins to have four neighbors. The magnetization of the entire system is expressed as

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

a sum over all spins. The energy per spin is

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

and the magnetization per spin is given by

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

These values will be used to compare and analyze results. The "inverse temperature" is given by

$$\beta = \frac{1}{k_B T} \tag{5}$$

where T is the systems' temperature and k_B the Boltzmann constant. The partition function is expressed as

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

This, the 'inverse temperature" and the total energy of the system appear in the *Boltzmann distribution*,

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

This will be the probability distribution used for random sampling in our Monte Carlo approach.

For comparison with early numerical implementations we will first consider an analytical solution. The following table I summarizes all sixteen possible **spin configurations** of a 2×2 lattice with *periodic boundary conditions*.

TABLE I. Analytic values for the sixteen **spin configurations** of the 2×2 Ising model lattice.

Nr. of spins in state +1	Degeneracy	Total energy	Total magnetization
0	1	-8J	-4
1	4	0	-2
2	4	0	0
2	2	8J	0
3	4	0	2
4	1	-8J	4

Based on the values in table I we derive the specific analytical expressions for the 2×2 lattice case. The calculations of these analytic solutions can be found in appendix A and are used to verify the numerical solutions.

We will study properties of the system at equilibrium for different lattice sizes as a function of temperature T. The analysis will focus on four main properties. The mean energy (eq. 3), mean magnetization (eq. 4), specific heat capacity normalized to number of spins

$$C_V = \frac{1}{N} \frac{1}{k_B T^2} \left(\langle E^2 \rangle - \langle E \rangle^2 \right), \tag{8}$$

and the susceptibility normalized to number of spins

$$\chi = \frac{1}{N} \frac{1}{k_B T} \left(\langle M^2 \rangle - \langle |M| \rangle^2 \right). \tag{9}$$

Analytical solutions to these can also be found in appendix A.

Markov chain Monte Carlo and the metropolis algorithm

The Markov Chain Monte Carlo method with the metropolis algorithm requires both detailed balance and ergodicity. This means that every transition in the system we are looking at is reversible, aperiodic and positive recurrent. For the Ising model the metropolis algorithm defines a transition probability as a function of energy difference between the previous and new state which ends up fulfilling the detailed balance. This probability is expressed in equation 7 and also fulfills ergodicity since every new change of the system is independent of every other change that has been or will be performed.

A flow chart of the Metropolis algorithm can be found in appendix D, and is described as follows. We start with an initial energy computed from the random or ordered initial configuration of spins. We randomly flip one spin and calculate a temporary energy which we use to calculate ΔE. ΔE can take 5 different values which gives us the possibility to use the returned value as an index that matches with the pre-calculated $w = e^{-\beta \Delta E}$. We then check if w is larger or equal to a random number r from the uniform distribution [0,1] and if so, we accept the new flipped configuration. If not we keep the old configuration which allows us to move towards the energy minimum for the given temperature. The probability of this configuration change is what is earlier described as the transition probability. After this we update the total energy and magnetization of the lattice if the configuration is changed. We repeat this N times before we use the last calculated M and E to add to cumulative sums of E, E^2, M, M^2 and $|M|$. This is one Monte Carlo cycle and is repeated for a chosen number of cycles before expectation values using the cumulative sums are taken. In total the metropolis algorithm together with the Monte

Carlo sampling becomes a Markov Chain. This is how the probability to transition to a new state only is dependent on the current spin configuration and following energy and magnetization.

Random number generation

For our random number r in the interval [0,1] we use a Mersenne Twister generator. This generator has a period of 2^{199937} which defines how long it continues before repeating itself [3]. This means that we will not run into any problems with a random number being repeated, which may compromise our performance.

Critical phenomena

The final goal for the MCMC simulations will be to approximate the critical temperature for an *infinite* ($L = \infty$) 2D Ising model. As mentioned in section I, the analytical solution for this was found by Lars Onsager [2]. His result is

$$T_{c\ \text{analytic}}(L = \infty) = \frac{2}{\ln(1+\sqrt{2})} J/k_B \approx 2.269 J/k_B.$$

For our approximation we will use the scaling relation

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

Here a is constant. Theoretical background for this scaling relation can be found in appendix C. Eq. 10 can be rearranged to the linear function

$$T_c(L) = aL^{-1} + T_c(L = \infty). \tag{11}$$

This relation will be approximated by performing linear regression using the scipy package in python. The data set used will be T_C values for different lattice sizes L from plots of C_V and χ. In other words, the temperature at which C_V and χ reach their peak value for different lattice sizes. The linear regression resulting in

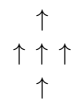
$$y = ax + b \tag{12}$$

will thus supply a value for a and the interception point b yields an approximation for $T_c(L = \infty)$.

Optimization and parallelization

The Ising model simulations we will be implemented using periodic boundary conditions. This way, all spins will have four neighbors as seen in table II, also at the boundaries of the lattice.

TABLE II. An example of a spin and its four neighbors. Here all five spins are in an up state (+1).



Periodic boundaries could be implemented using if-tests, but if-tests within loops can be slow. To make the code slightly more efficient we have chosen to write a `periodic` function using the modulo operator. For an index `idx`, lattice dimension `dim` and `offset` equaling 1 or -1 for neighboring indexes up or right, or down or left respectively we have

```
neighboring index = (idx + dim + offset) % dim.
```

The Monte Carlo method will repeatedly require the Boltzmann factor $e^{-\beta\Delta E}$. The energy shift induced by flipping a single spin

$$\Delta E = E_{\text{after}} - E_{\text{before}} \tag{13}$$

can only take five possible values in a 2D-lattice of arbitrary size($L > 2$) This is shown in appendix B. These values are

$$\Delta E = 8J, 4J, 0, -4J, -8J. \tag{14}$$

To reduce computational cost we will avoid repeatedly calling the exponential function. This is done by precomputing the five possible Boltzmann factors in an array. Furthermore, the code is written such that it can be run parallelized. This allows us to run it on several threads using OpenMP to reduce runtimes. We chose to parallelize at the level of temperature and number of cycles meaning that each thread gets its own set of temperatures or number of cycles to run the MCMC algorithm for. This does not speed up the algorithm when running for one specific temperature or number of MC cycles, but can improve the speed when testing for different values of these parameters.

III. RESULTS

We begin by verifying the numerical implementation by comparing analytical and numerical results for a 2×2 lattice case for $T \in [0.5, 4.0] J/k_B$ in figure 1, 2, 3 and 4.

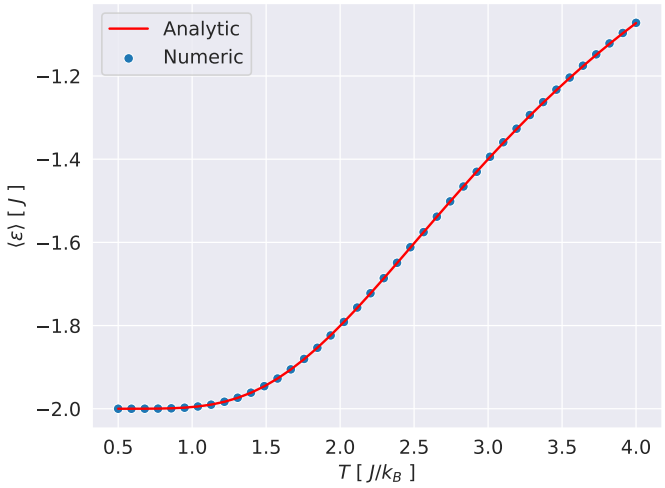


FIG. 1. Comparison of the red analytic and blue numeric results in a 2×2 lattice case. Here one sees the expectation value for the energy per spin ϵ plotted against different temperatures T. Numerical results shown are for 10^6 MC cycles.

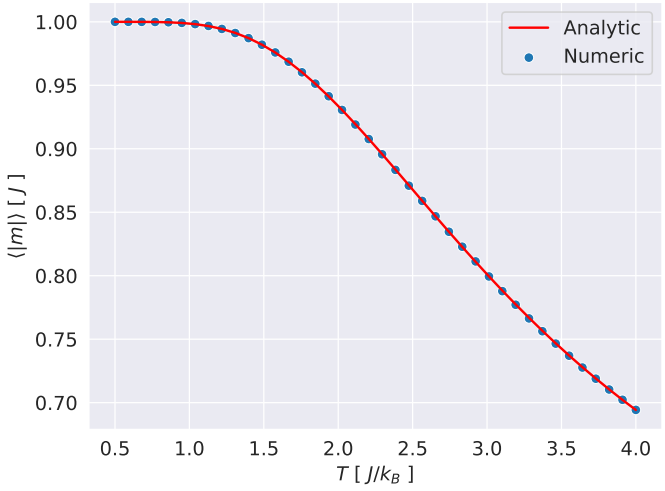


FIG. 2. Comparison of the red analytic and blue numeric results in a 2×2 lattice case. Here one sees the expectation value for the magnetization per spin m plotted against different temperatures T. Numerical results shown are for 10^6 MC cycles.

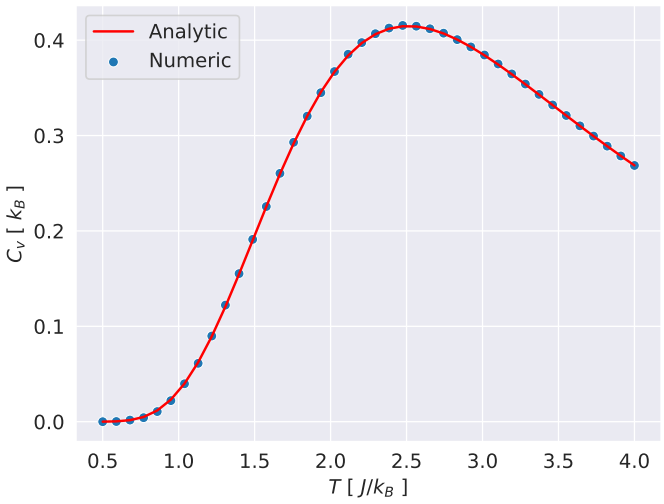


FIG. 3. Comparison of the red analytic and blue numeric results in a 2×2 lattice case. Here one sees the expectation value for the heat capacity C_V (normalized to number of spins) plotted against different temperatures T. Numerical results shown are for 10^6 MC cycles.

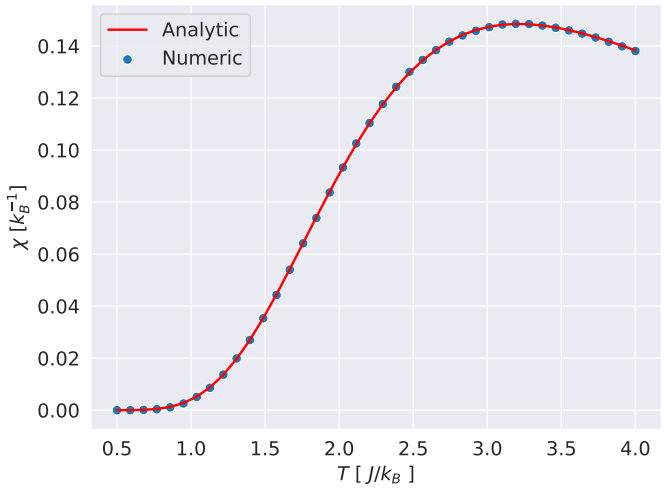


FIG. 4. Comparison of the red analytic and blue numeric results in a 2×2 lattice case. Here one sees the expectation value for the susceptibility χ (normalized to number of spins) plotted against different temperatures T. Numerical results shown are for 10^6 MC cycles.

Figures 1, 2, 3 and 4 show excellent agreement between numerical MCMC results and analytical results for the whole range of temperatures shown.

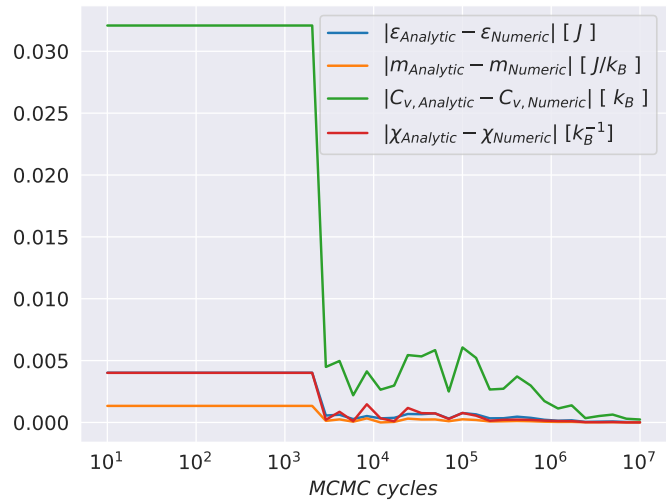


FIG. 5. The blue, yellow, green and red line show the change of the difference between numerical and analytical results as a function of MCMC cycles for $\langle\epsilon\rangle$, $\langle|m|\rangle$, C_V and χ respectively. In this case $L = 2$ and the results are presented on a log scale along the x axis.

In figure 5 one can see how the difference between numerical and analytical results decreases for increasing MCMC cycles. After about 10^4 cycles the numerical results starts to stabilize and are in good agreement with the analytical result for a 2×2 lattice. We see that 10^7 cycles gives the best agreement, but because of limited computational power we find a middle ground for further computations of 10^6 cycles.

This allows us to move on to more complex computations of larger lattice sizes. The next figures show results to study the **burn-in** time for our MCMC method for a 20×20 lattice. Results are shown for both a lattice with random spins and an ordered lattice with all spins starting in an up ($\uparrow$) state.

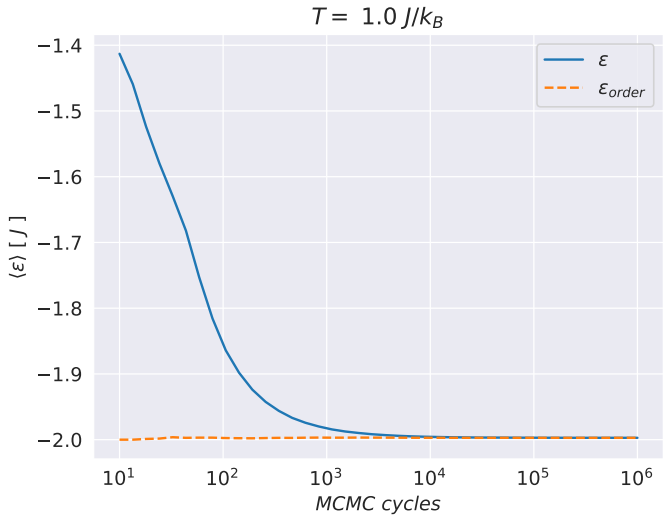


FIG. 6. Numerical approximation values for $\langle\epsilon\rangle$ against number of MCMC cycles for $T = 1.0J/k_B$. The blue line shows the evolution from a random spin configuration whereas the orange line is for the case of an ordered spin configuration. Here $L = 20$ and the results are presented on a log scale along the x axis.

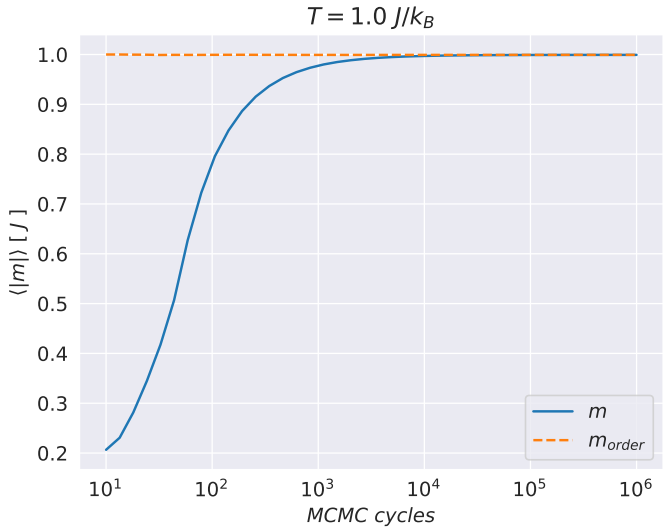


FIG. 7. Numerical approximation values for $\langle|m|\rangle$ against number of MCMC cycles for $T = 1.0J/k_B$. The blue line shows the evolution from a random spin configuration whereas the orange line is for the case of an ordered spin configuration. Here $L = 20$ and the results are presented on a log scale along the x axis.

Figure 6 and 7 show that the MCMC method has a burn-in time of about 10^4 cycles for $T = 1.0J/k_B$ in an unordered spin case. Starting with ordered spins there is no burn-in time at all.

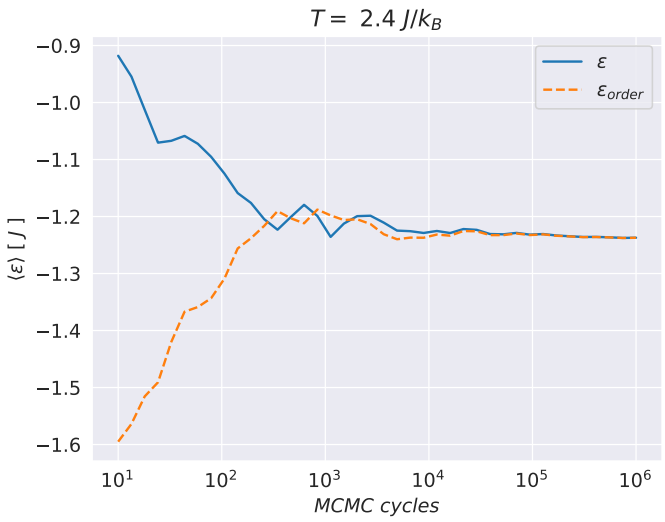


FIG. 8. Numerical approximation values for $\langle\epsilon\rangle$ against number of MCMC cycles for $T = 2.4J/k_B$. The blue line shows the evolution from a random spin configuration whereas the orange line is for the case of an ordered spin configuration. Here $L = 20$ and the results are presented on a log scale along the x axis.

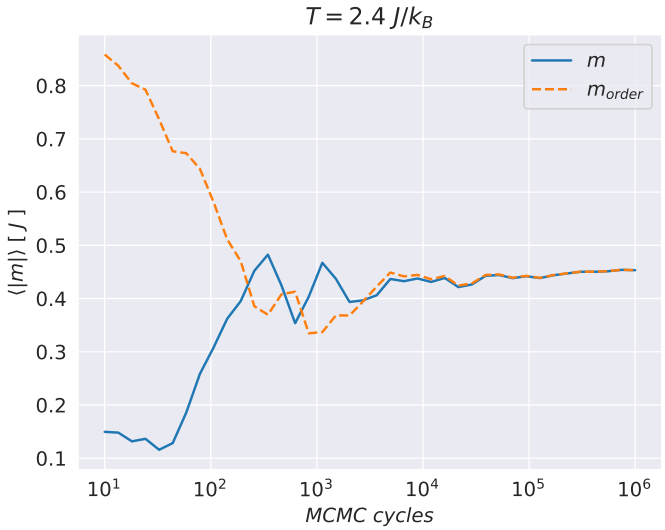


FIG. 9. Numerical approximation values for $\langle|m|\rangle$ against number of MCMC cycles for $T = 2.4J/k_B$. The blue line shows the evolution from a random spin configuration whereas the orange line is for the case of an ordered spin configuration. Here $L = 20$ and the results are presented on a log scale along the x axis.

For $T = 2.4J/k_B$ we see that the values for $\langle\epsilon\rangle$ in figure 8 and equally for $\langle|m|\rangle$ in figure 9 are fairly stable around 10^4. The full burn-in time looks to be around 10^5 cycles for both ordered and unordered spins.

Moving beyond expectation values we approximate the probability function $p_\epsilon(\epsilon; T)$ using MCMC samples. This is done for both $T = 1.0J/k_B$ and $T = 2.4J/k_B$.

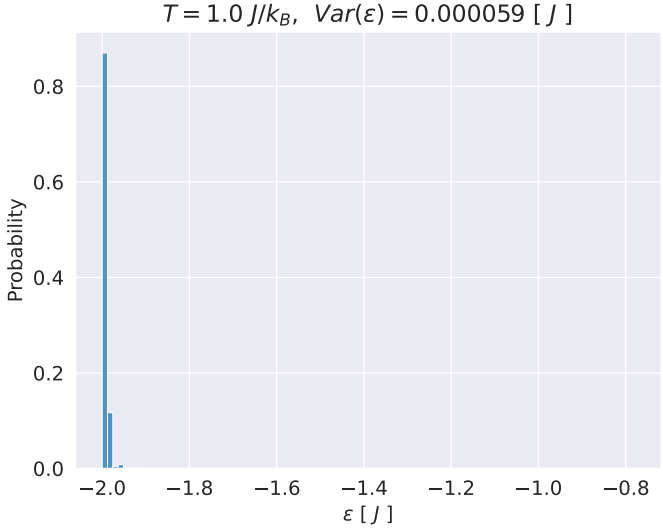


FIG. 10. Approximation of the probability function $p_\epsilon(\epsilon; T)$ for $L = 20$ and $T = 1.0J/k_B$ with random initial.

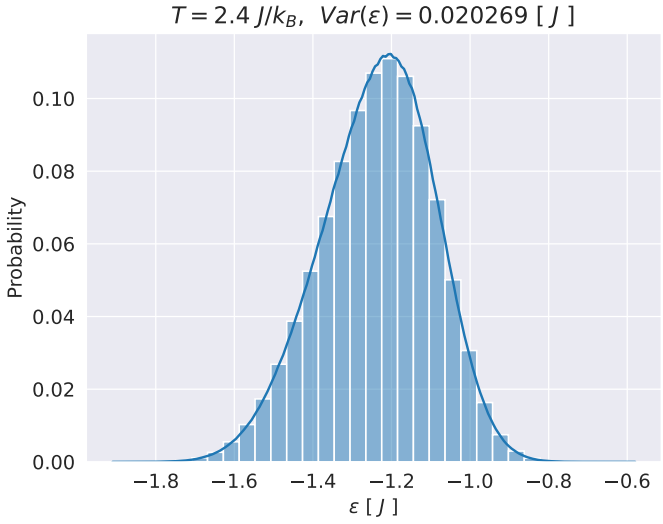


FIG. 11. Approximation of the probability function $p_\epsilon(\epsilon; T)$ for $L = 20$ and $T = 2.4J/k_B$ with random initial spins. The blue line shows the kernel distribution estimation of the histogram.

Comparing figure 10 for $T = 1.0J/k_B$ and figure 11 for $T = 2.4J/k_B$ the most noticeable result is the difference in variance. For $T = 1.0J/k_B$ an $\epsilon \approx -2$ is by far the most likely, whilst relatively highly probable ϵ values for $T = 2.4J/k_B$ cover a larger range of approximately $[-1.7, -0.9]J/k_B$.

In figure 12, 13, 14, and 15 we have computed $\langle\epsilon\rangle$, $\langle|m|\rangle$, C_V and χ respectively for varying temperatures in the range $[2.1, 2.4]J/k_B$.

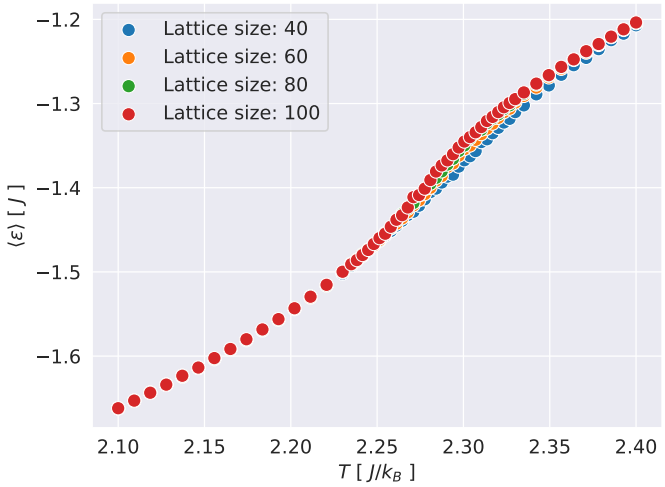


FIG. 12. $\langle\epsilon\rangle$ is shown as a function of temperature T for lattice size $L = 40, 60, 80, 100$.

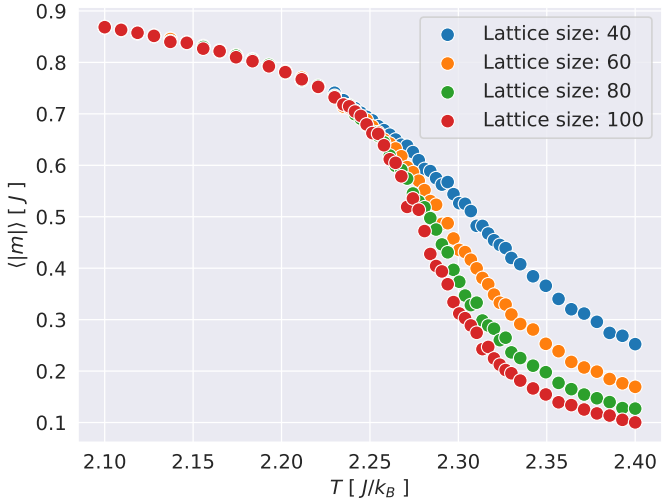


FIG. 13. $\langle|m|\rangle$ is shown as a function of temperature T for lattice size $L = 40, 60, 80, 100$.

Figure 12 shows small variations in mean energy for increasing lattice size L. Looking closely in at $T \in [2.22, 2.4]J/k_B$ one can see that $\langle\epsilon\rangle$ increases slightly with increasing L. In figure 13 the mean magnetization changes more noticeably for different L. $\langle|m|\rangle$ tends toward zero for increasing T and noticeably the graphs slope steepens around $T \in [2.25, 2.30]J/k_B$ for increasing lattice sizes.

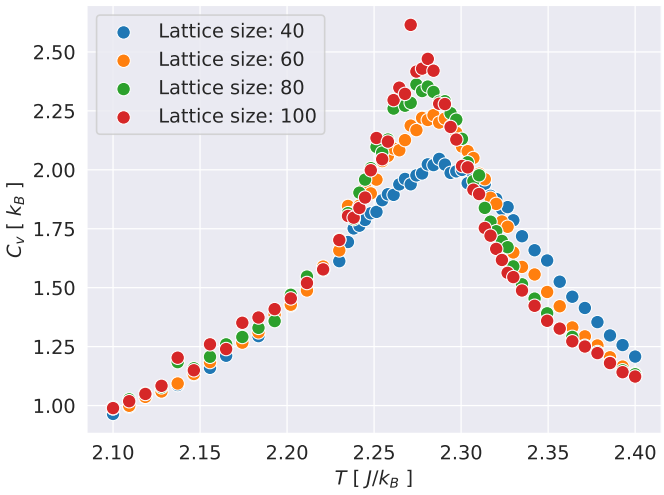


FIG. 14. C_V is shown as a function of temperature T for lattice size $L = 40, 60, 80, 100$.

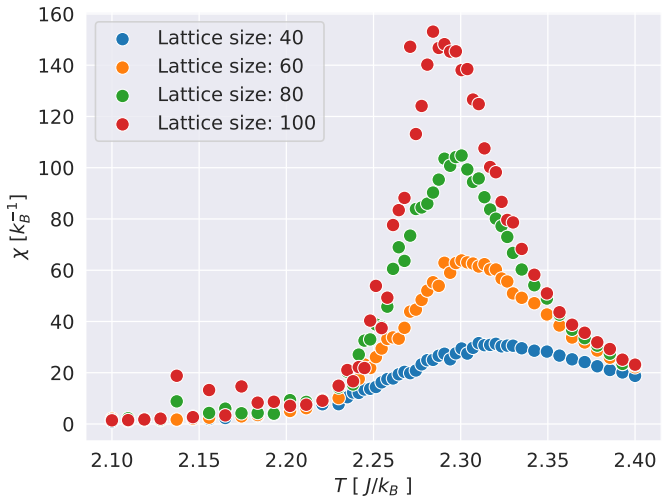


FIG. 15. χ is shown as a function of temperature T for lattice size $L = 40, 60, 80, 100$.

The specific heat capacity in figure 14 increases for $T \in [2.25, 2.30]J/k_B$ with increasing lattice size. In figure 15 the susceptibility shows similar behavior to the heat capacity with an increase of χ for larger lattice sizes in the region of $T \in [2.25, 2.35]J/k_B$. In both cases the peak value for a 40×40 lattice is around $T = 2.30J/k_B$. As the lattice size increases, the peak C_V and χ trend towards a lower temperature. For all four figures above a smaller ΔT has been chosen for $T > 2.235J/k_B$, specifically for $T \in [2.235, 2.33]J/k_B$ we have $\Delta T = 0.003J/k_B$.

TABLE III. Critical temperature as a function of lattice size computed to match the maximum computed value for both χ and C_V.

Lattice sizes	40	60	80	100
T_C using C_V $[J/k_B]$	2.287	2.284	2.274	2.271
T_C using χ $[J/k_B]$	2.310	2.300	2.301	2.284

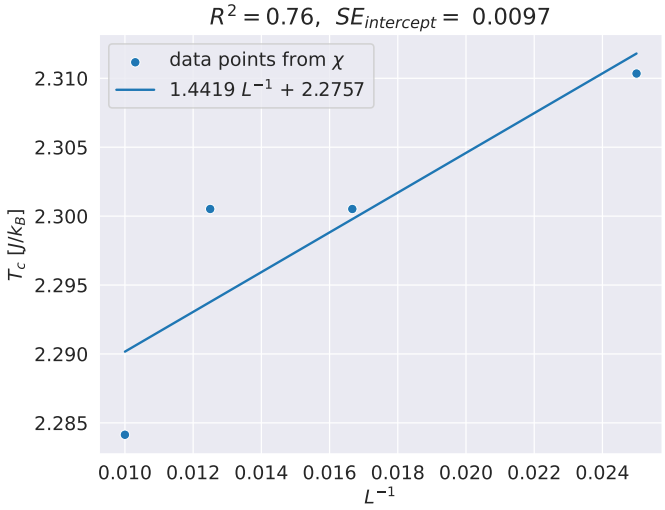


FIG. 16. The blue line shows linear regression using critical temperature points T_c from χ for different lattice sizes L. Here temperature T is plotted against L^{-1}.

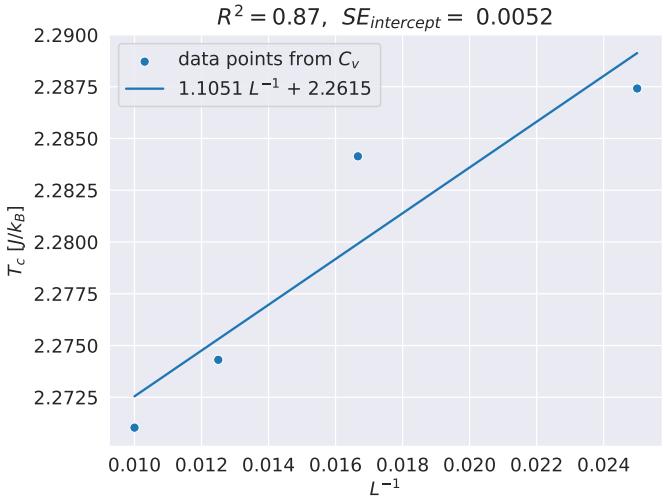


FIG. 17. The blue line shows linear regression using critical temperature points T_c from C_V for different lattice sizes L. Here temperature T is plotted against L^{-1}.

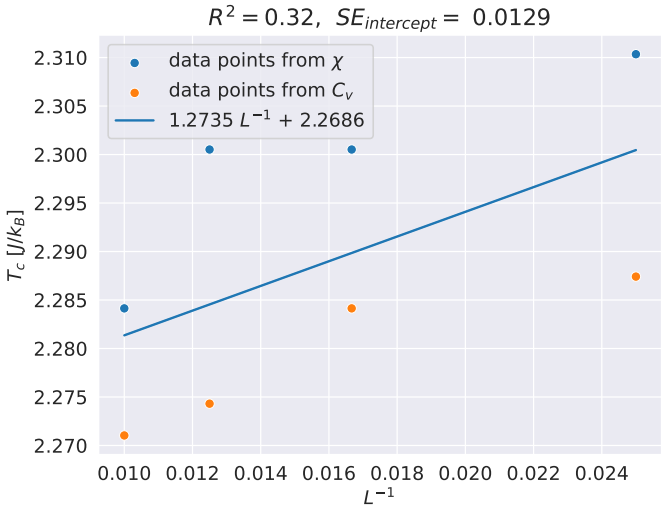


FIG. 18. The blue line shows linear regression using critical temperature points T_c from C_V and χ for different lattice sizes L. Here temperature T is plotted against L^{-1}.

Taking the temperature values at the peak values of the specific heat capacity C_V and the susceptibility χ show in table III and performing linear regression results in the blue line in figure 16 and 17 where we see good linear fits. For χ we have the function

$$T_{c,\chi}(L) = 1.4419L^{-1} + 2.2757 J/k_B. \tag{15}$$

This gives us $a = 1.4419$ and letting $L = \infty$ we have with the uncertainty expressed as the standard error

$$T_{c,\chi}(L = \infty) = 2.2757 \pm 0.0097 J/k_B. \tag{16}$$

For C_V we have:

$$T_{c,C_V}(L) = 1.1051L^{-1} + 2.2615 J/k_B. \tag{17}$$

This gives us $a = 1.1051$ and letting $L = \infty$ we have

$$T_{c,C_V}(L = \infty) = 2.2615 \pm 0.0052 J/k_B. \tag{18}$$

A combination of figure 17 and 16 can be seen in figure 18 and gives us

$$T_{c,tot}(L = \infty) = 2.2686 \pm 0.0129 J/k_B. \tag{19}$$

TABLE IV. Time used for different number of threads in parallelization at temperature level. A quad-core, 8 thread CPU has been used for the ising model initialized with a 5×5 lattice with randomized initial spins and temperature $T = 1$ J/k_B

Number of threads	1	4	8
Time used (s)	29.12	8.42	7.32
speed-up factor	1	3.45	3.92

In table IV we see that parallelizing our code can reduce computation time a factor of 0.25 using 8 threads for an 8 thread CPU. We also see that a 4 thread parallelization performs well with a speed-up factor of 3.45.

IV. DISCUSSION

We start by verifying our numerical Ising model simulation against analytical results for a 2×2 lattice to ensure its functionality. The results in figures 1, 2, 3 and 4 show excellent agreement over $T \in [0.5, 4.0] J/k_B$ for 10^6 MCMC cycles. This shows the MCMC method is a suitable choice for this 2D Ising model problem. Taking a closer look at the amount of cycles necessary for good agreement in figure 5 confirms that at least 10^4 MCMC cycles are necessary for a precise numerical approximation. All further simulations are therefore run for 10^6 cycles. 10^7 would definitely give even better results, but also comes at a large run time cost.

Studying the burn-in time (i.e. the time it takes for the algorithm to stabilize on the correct values) for a 20×20 lattice for $T = 1.0 J/k_B$ in figure 6 and 7 we see that the only noticeable burn-in time is for the unordered spin case. For ϵ in figure 6 this is between 10^3 and 10^4 cycles. The ordered case of all spins being up already matches the expected energy, thus not having a burn-in time. Figure 8 and 9 for $T = 2.4 J/k_B$ on the other hand, show a slightly longer burn-in time closer to 10^5 cycles. In this case the ordered spin configuration does not instantly match the expected values resulting in more similar behavior for the ordered and unordered burn-in time. One can also see some oscillatory behavior matching the almost even distribution to both sides of $-1.2J$ in figure 11. It could be argued that more precise results can be generated if one chooses not to use samples from the burn-in time. This is because a very unlikely initial state can affect the final expectation values since large amount of the samples taken from the burn-in-time are highly unlikely. A solution to this, and what we have chosen to do, is to run as many cycles as is computationally reasonable when considering runtimes. Based on the burn-in time results, running 10^6 cycles seems enough to remove the significance of samples from the burn-in time. At least in the $L = 20$ case. Whether this is actually true for larger lattice sizes as well would require further testing. Another option, and potential future improvement would be to not use the burn-in time data at all and only use samples from after the burn in time to calculate our expectation values.

Studying the histograms approximating the probability function $p_\epsilon(\epsilon; T)$ in figure 10 and 11 there is a clear difference in variance. We have a variance of $0.020269J$ for $T = 2.4 J/k_B$ and a much smaller $0.000059J$ for $T = 1.0 J/k_B$. This can be explained by an increasing probability of jumping to a higher energy state as T increases. This means that we for a higher temperature would expect great oscillation even after the burn in time which for a temperature of $1 J/k_B$ would be more unlikely. We can see this in figure 8 and 6 where $T = 1 J/k_B$ only takes values very close to $-2J$ after 10^4 cycles equaling 99% of the total samples. For $T = 2.4 J/k_B$ we can see that we have fluctuations after cycle 10^4 and even visible different sample values between cycle 10^5 and 10^6. This is what mainly explains the large difference in variance between the two histograms.

Having developed a certain understanding and confidence in our model we now start looking for indications of phase transition. One can see that the mean magnetization drops off more abruptly after $T = 2.25 J/k_B$ in figure 13. This matches the expected phase transition for the Ising model which is a loss of magnetization after a certain critical temperature T_c. Seeing how the graphs slope steepens for increasing L we would expect it to be infinitely steep in an infinite lattice case at T_c. Studying the region $T \in [2.235, 2.33] J/k_B$ in figure 14 and 15 we observe definite peak values for C_V and χ. This abrupt change from the change of the external temperature parameter further indicates a phase transition happening in this region of T. From our linear regressions using the maximum values of χ and c_v in figure 16 and 17 we see a good linear fit with relatively large R^2 meaning that our data follows the expected dependence on L^{-1}. When combining the two results in figure 18 we get a prediction of the critical temperature giving us

$$T_{c,tot}(L = \infty) = 2.2686 \pm 0.0129 J/k_B. \tag{20}$$

This is an excellent approximation of Onsager's analytical result of

$$T_{c\ \text{analytic}}(L = \infty) \approx 2.269 J/k_B \tag{21}$$

confirming the Markov Chain Monte Carlo method's applicability to the modeling of the 2D Ising model without an external magnetic field.

To improve the approximation one could use an even smaller ΔT. This would help us to more accurately find the maximum of both χ and C_v for a chosen lattice size matching its critical temperature. Testing for more lattice sizes would also help us improve our approximation. This would give us more data points and a following lower uncertainty in our linear regression results.

Considering code optimization, we have chosen to parallelize the section of the code lopping over different temperatures and different numbers of cycles. This gives us time savings when we are plotting against these two variables, but not when running a single MCMC run. Another way of parallelizing the code would therefore be at the level of the MCMC "walkers" using several threads to execute multiple individual MCMC runs to finally combine the results. This would improve performance also when running for only one temperature and one number of MCMC cycles, and may also see better performance overall. Time savings from implementing such a parallelization is left as possible research.

From our timing test using parallelization at temperature level we saw a great reduction in computation time when using more threads with a speed-up factor of 3.92 going from utilizing 1 to 8 threads. For an 8 thread CPU we would expect the speed-up factor being closer to 8, which shows that the physical core count of the processor influences the speed-up factor. In our case of

a quad-core CPU we saw that even 4 threaded parallelization got close to the 8 thread performance. In total this means that the expected speed-up factor using all of a processor's threads would be closer to its physical core and not thread count. On the other hand, we do not know how parallelization at the level of the Markov Chain Monte Carlo method would influence the speed-up time. Such a parallelization may see better performance than our time level parallelization overall as well as better scaling when using more threads. This would require further investigation.

V. CONCLUSION

We have simulated the 2D Ising model using a Markov Chain Monte Carlo (MCMC) approach. This was done taking random samples from the Boltzmann distribution using the Metropolis algorithm. We have found the MCMC method to give good approximations of mean energy, magnetization, heat capacity and susceptibility of the Ising model for different temperatures. The chosen 10^6 MCMC cycles offers a good compromise between precision and computational cost. Whether the chosen number of cycles is actually enough in comparison to the burn-in time for lattices larger than 20×20 could be an area of future research. Studying the Ising model parameters' behavior over different temperatures such as in figure 14 shows indications of a phase transition at a certain critical temperature T_c. Using eight data points of T_c for four different lattice sizes we perform a linear regression allowing us to approximate the critical temperature for an infinite lattice. This results in $T_{c,tot}(L = \infty) = 2.2686 \pm 0.0129 J/k_B$. This is an impressive approximation of Onsager's analytical result of $T_{c\ \text{analytic}}(L = \infty) \approx 2.269 J/k_B$. A smaller variance for the probability function $p_\epsilon(\epsilon; T)$ at $T = 1.0 J/k_B$ was found in comparison to $T = 2.4 J/k_B$ indicating a low probability of jumping to a higher energy state at low temperatures. Furthermore, a speed-up factor of 3.92 was achieved when running the parallelized code across 8 threads on an 8 thread quad-core CPU. This was good enough for the chosen simulations, yet it is possible other parallelization implementations are more efficient. This again would require further research. Overall, running the same simulations for a higher number of MCMC cycles, more lattice sizes, smaller choices of ΔT and disregarding burn-in time samples would presumably give better results than presented.

[1] M. Hjorth-Jensen, *Computational Physics* (Department of Physics, University of Oslo, 2015).
[2] UiO, "Project 4," .
[3] M. Shema, in *Seven Deadliest Web Application Attacks*, edited by M. Shema (Syngress, Boston, 2010) pp. 71–90.

Appendix A: Analytical solutions for a 2×2 lattice

For the case of a 2×2 lattice with $L = 2$ and $N = 4$ we have sixteen possible spin configurations show in table I. These values will be used for the analytic solutions. The specific partition function becomes

$$Z = \sum_{\text{all possible } \mathbf{s}} e^{-\beta E(\mathbf{s})} = 2e^{-\beta(-8J)} + 2e^{-\beta 8J} + 12e^{0} = 2e^{\beta 8J} + 2e^{-\beta 8J} + 12 = 4(\cosh(8\beta J) + 3).$$

Additionally, we will calculate a few expectation values for which the general formula is given as

$$\langle A \rangle = \sum_{s} A_s p(s).$$

This is a sum over all spin states s_i. Here $p(s)$ is a chosen probability distribution, in our this case the Boltzmann distribution in eq. 7. The fist and second of the following analytical derivations will use the hyperbolic function definition $\sinh(x) = \frac{e^x - e^{-x}}{2}$ and $\cosh(x) = \frac{e^x + e^{-x}}{2}$ respectively. For the energy we have

$$\begin{aligned}\langle E \rangle &= \sum_{s} E(\mathbf{s}) p(\mathbf{s}; T) = \frac{1}{Z} \sum_{s} E(\mathbf{s}) e^{-\beta E(\mathbf{s})} = \frac{1}{Z} \left(2 \cdot (-8J) e^{\beta 8J} + 2 \cdot 8J e^{-\beta 8J}\right) \\ &= \frac{1}{Z} \left(-16 e^{\beta 8J} + 16 e^{-\beta 8J}\right) = \frac{16J}{Z} \left(e^{-\beta 8J} - e^{\beta 8J}\right) = -\frac{32J \sinh(8J\beta)}{Z},\end{aligned}$$

$$\begin{aligned}\langle E^2 \rangle &= \sum_{s} E(\mathbf{s})^2 p(\mathbf{s}; T) = \frac{1}{Z} \sum_{s} E(\mathbf{s})^2 e^{-\beta E(\mathbf{s})} = \frac{1}{Z} \left(2 \cdot (-8J)^2 e^{\beta 8J} + 2 \cdot (8J)^2 e^{-\beta 8J}\right) \\ &= \frac{128J^2}{Z} \left(e^{-\beta 8J} + e^{\beta 8J}\right) = \frac{256J^2 \cosh(8J\beta)}{Z},\end{aligned}$$

$$\langle \epsilon \rangle = \sum_{s} \epsilon_s p(\mathbf{s}; T) = \sum_{s} \frac{E(\mathbf{s})}{N} p(\mathbf{s}; T) = \frac{1}{N} \sum_{s} E(\mathbf{s}) p(\mathbf{s}; T) = \frac{\langle E \rangle}{4} = -\frac{8J \sinh(8J\beta)}{Z},$$

$$\langle \epsilon^2 \rangle = \sum_{s} \epsilon_s^2 p(\mathbf{s}; T) = \sum_{s} \left(\frac{E(\mathbf{s})}{N}\right)^2 p(\mathbf{s}; T) = \frac{1}{N^2} \sum_{s} E(\mathbf{s})^2 p(\mathbf{s}; T) = \frac{\langle E^2 \rangle}{16} = \frac{16J^2 \cosh(8J\beta)}{Z}.$$

Then for the magnetization we have

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

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

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

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

Finally, we find analytical expressions for the specific heat capacity

$$C_V = \frac{1}{N}\frac{1}{k_B T^2}\left(\langle E^2\rangle - \langle E\rangle^2\right) = \frac{1}{4}\frac{1}{k_B T^2}\left(\frac{256J^2\cosh(8J\beta)}{Z} - \left(-\frac{32J\sinh(8J\beta)}{Z}\right)^2\right),$$

and the susceptibility

$$\chi = \frac{1}{N}\frac{1}{k_B T}\left(\langle M^2\rangle - \langle|M|\rangle^2\right) = \frac{1}{4}\frac{1}{k_B T}\left(\frac{32}{Z}\left(e^{\beta 8J}+1\right) - \left(\frac{8}{Z}\left(e^{\beta 8J}+2\right)\right)^2\right).$$

Appendix B: Possible ΔE values

Considering a 2D lattice of arbitrary size ($L > 2$) and remembering that we are working with periodic boundary conditions, we can show that there are only a few possible values of ΔE. The calculation of ΔE between spin configurations will be limited to the flipping of a single spin. To find the possible energy differences we will look at a spin at a random position. This central spin will start in an up state(+1) and then be flipped (-1). We remind that $E(\mathbf{s}) = -J\sum_{\langle kl\rangle}^{4} s_k s_l$ and $\Delta E = E_{\text{after}} - E_{\text{before}}$.

$\begin{matrix} & \uparrow & \\ \uparrow & \uparrow & \uparrow \\ & \uparrow & \end{matrix}$ has $E = -4J$, now flipping we have $\begin{matrix} & \uparrow & \\ \uparrow & \downarrow & \uparrow \\ & \uparrow & \end{matrix}$ and $E = 4J$, resulting in $\Delta E = 8J$.

This can be show for the four remaining possible starting configurations as well.

$\begin{matrix} & \uparrow & \\ \downarrow & \uparrow & \uparrow \\ & \uparrow & \end{matrix}$ has $E = -2J$. Flipping we have $\begin{matrix} & \uparrow & \\ \downarrow & \downarrow & \uparrow \\ & \uparrow & \end{matrix}$ and $E = 2J$, resulting in $\Delta E = 4J$.

$\begin{matrix} & \uparrow & \\ \downarrow & \uparrow & \uparrow \\ & \downarrow & \end{matrix}$ has $E = 0$. Flipping we have $\begin{matrix} & \uparrow & \\ \downarrow & \downarrow & \uparrow \\ & \downarrow & \end{matrix}$ and $E = 0$, resulting in $\Delta E = 0$.

$\begin{matrix} & \uparrow & \\ \downarrow & \uparrow & \downarrow \\ & \downarrow & \end{matrix}$ has $E = 2J$. Flipping we have $\begin{matrix} & \uparrow & \\ \downarrow & \downarrow & \downarrow \\ & \downarrow & \end{matrix}$ and $E = -2J$, resulting in $\Delta E = -4J$.

$\begin{matrix} & \downarrow & \\ \downarrow & \uparrow & \downarrow \\ & \downarrow & \end{matrix}$ has $E = 4J$. Flipping we have $\begin{matrix} & \downarrow & \\ \downarrow & \downarrow & \downarrow \\ & \downarrow & \end{matrix}$ and $E = -4J$, resulting in $\Delta E = -8J$.

The five possible values of the energy difference are thus, $\Delta E = 8J, 4J, 0, -4J, -8J$.

Appendix C: Theoretical background for critical phenomena

Power laws with so-called critical exponents describe how a physical system behaves when close to its critical point. For the Ising model this is close to its critical temperature. For temperatures T close to T_c the *infinite* 2D Ising model's mean magnetization, heat capacity and susceptibility behave as follows:

$$\begin{aligned} \langle |m| \rangle &\propto |T - T_c(L = \infty)|^{\beta}, \\ C_V &\propto |T - T_c(L = \infty)|^{-\alpha}, \\ \chi &\propto |T - T_c(L = \infty)|^{-\gamma}. \end{aligned}$$

Here $\beta = 1/8$, $\alpha = 0$ and $\gamma = 7/4$ are the critical exponents. We see that C_V and χ diverge close to T_c. The *correlation length*

$$\xi \propto |T - T_c(L = \infty)|^{-\nu} \tag{C1}$$

with $\nu = 1$ also diverges near T_c. For our finite system $\xi = L$ is the largest correlation length. Replacing for ξ in eq. C1 then leads to the scaling equation

$$T_c(L) - T_c(L = \infty) = aL^{-1}. \tag{C2}$$

Appendix D: Metropolis algorithm flow chart

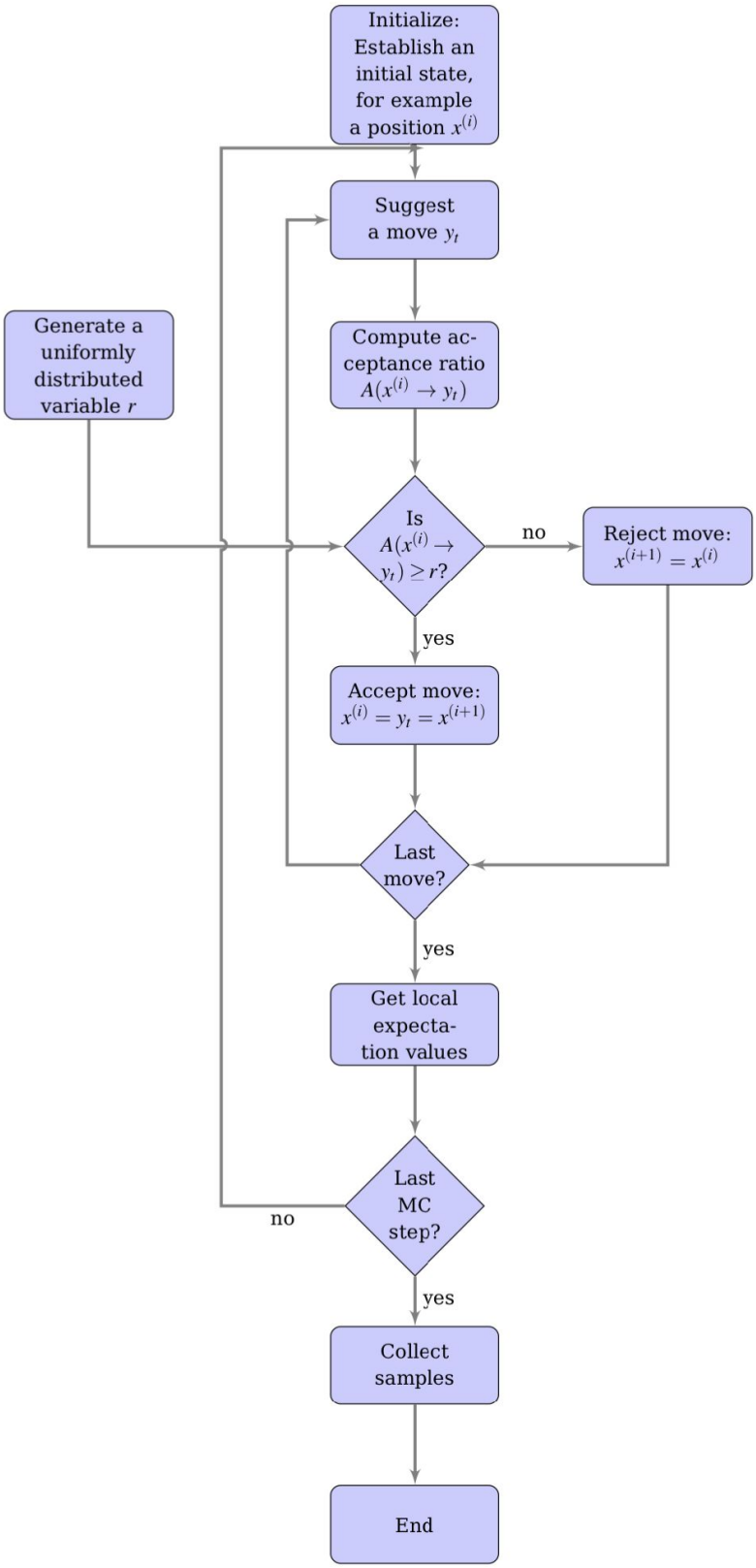


FIG. 19. Flowchart of the Metropolis algorithm taken from Computational Physics lecture notes [1].