The 2D Ising Model

Andrew Quan, Oskar Idland, Hishem Kløvnes, Håvard Skåli

(Dated: November 22, 2023)

This study delves into the temperature-dependent behaviour of spin particle configurations by studying the 2D Ising model, using the Markov Chain Monte Carlo method and OpenMP parallelization for computation efficiency. We were able to estimate that for an $L \times L$ lattice with size up to $L \approx 80$, performing 100 000 Monte Carlo cycles is sufficient for result stability at temperature $T = 1.0\,J/k_{\text{B}}$. We studied equilibration times for a 20×20 lattice at the aforementioned temperature as well as $T = 2.4\ J/k_{\text{B}}$, and estimated these to be approximately 8 000 and 80 000 Monte Carlo cycles respectively. Furthermore, we observed that inconsistencies between numerical and analytical results increased with temperature due to flaws in our numerical algorithms, and took this into consideration when interpreting our results. We investigated the time-dependent phase transitions between ferromagnetism and paramagnetism for lattice sizes $L \in [40, 60, 80, 100]$, and estimated the critical temperatures $T_c(L)$ where these seemingly occured. In spite of our flawed algorithms, we were able to use these results to estimate the critical temperature of an infinite 2D Ising model to be approximately $T_c(L = \infty) \approx 2.259 \pm 0.003\ J/k_{\text{B}}$, deviating from the analytical result derived by Onsager by $0.01\ J/k_{\text{B}}$.

https://github.uio.no/oskarei/CompFys-Project4

CONTENTS

I. Introduction 1

II. Methods 2
A. Theory and Analytical Expressions 2
1. Definitions and Properties 2
2. Expectation Values 3
3. Energy Shifts 4
4. Phase Transitions and Critical Temperature 4
B. Algorithms 4
C. Numerical Calculations 5
1. Specific Approaches 5
2. Program Structure 5
3. Tools 7

III. Results & Discussions 7
A. Effects of Parallellisation 7
B. Expectation Values 7
1. Comparing with Analytical Results 7
2. Equilibration Time 8
C. Probability Functions 8
D. Phase Transitions and Critical Temperature 9
1. Lattices of Increasing Size 9
2. The Infinite Ising Model 10

IV. Conclusion 10

References 11

I. INTRODUCTION

We will be studying the Ising model in two dimensions to explore the temperature-dependent behaviour in ferromagnets. Also called the Lenz-Ising model named after physicists Ernst Ising and Wilhelm Lenz, it is a simple statistical model which lets us investigate phase transitions. The model sees magnetic dipole moments of atoms arranged in a 2D square lattice, which hereafter will simply be called individual spins as it is the property of the atoms we will be focusing on. This lets us treat atoms with equal spins in the lattice as indistuingishable. Furthermore, each individual spin in the lattice can be in one of two discrete states at any moment in time, spin up or spin down, which are described with $s_i = +1$ and $s_i = -1$ respectively. A simple representation of the Ising model made by Sascha Wald is shown in fig. 1.

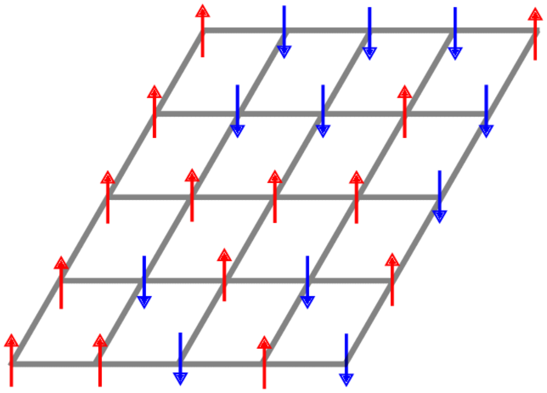

FIG. 1: Schematic representation of a configuration of the 2D Ising model. The individual spins are spaced evenly in a square lattice, some pointing up ($s_i = +1$) and some pointing down ($s_i = -1$). Made by Sascha Wald and taken from ResearchGate [1].

The lattice structure allows each spin to interact with its neighbouring spins, and if neighbours have different spins they will have higher energy than if their spins are aligned. The system will tend to align such that the total energy is as low as possible, but if heat enters the system this tendency will be disturbed. Thus, at different temperatures we may witness different structural phases. A particular goal in this investigation is to numerically estimate the critical temperature at which our 2D system undergoes a phase transition from a magnetised phase to a phase with no net magnetisation. To do this we will take use of the Markov Chain Monte Carlo method, and to increase the efficiency of our program we will parallelise it using OpenMP [2]. The Monte Carlo algorithm and our program structure will be described in further detail in section II B and section II C 2 respectively. In section III we will present, interpret and discuss our results, with section IV summarising our main discoveries and concluding this work.

II. METHODS

A. Theory and Analytical Expressions

1. *Definitions and Properties*

We will consider a square lattice of length L such that the number of spins are $N = L^2$, where the i-th spin is denoted s_i. The spin state of the entire system, which is called a spin configuration or microstate, can then be described in vector notation as

$$\mathbf{s} = (s_1, s_2, ...s_N) \tag{1}$$

In the Ising model we will assume no external magnetic field, such that we can approximate the total energy of the system as being fully determined by the neighbouring spin pairs. If we let J be a coupling constant associated with one such pair, the total energy of a microstate $\mathbf{s}$ is then

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

where $\langle kl \rangle$ means that the sum goes over all neighbouring pairs of spins without double counting. We will be working with unitless spins, which means that J has units of energy, and we will therefore express the energy in units of J. Moreover, we will be using periodic boundary conditions in our model, which means that all spins will have exactly four neighbours in the lattice. The neighbours above and to the left of the spin in the upper left corner will then be the spins in the lower left and upper right corners respectively. Furthermore, since the spin describes the atoms' magnetic dipole moments, the total magnetisation of a microstate can be written as

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

which we see is dimensionless in our choice of units. Consequently, we can define the energy per spin $\epsilon(\mathbf{s})$ and magnetisation per spin $m(\mathbf{s})$ as

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

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

For our lattice of spins we ca define a macrostate as the number of spins $s = +1$ in the lattice, such that there are multiple microstates $\mathbf{s}$ for every macrostate, except for the ones where all spins in the lattice are aligned parallell. Furthermore, a given macrostate can yield different values of the total energy $E(\mathbf{s})$, depending on how the spins are arranged. For example, if we imagine a 2×2 lattice in the macrostate where two of the spins are positive and two of them are negative, there are multiple ways of arranging them, hence multiple microstates. Due to the periodic boundary conditions of the Ising model, two of these microstates will yield the total energy $E(\mathbf{s}) = 8\,J$ while four of them will have zero total energy, since we treat the spins as indistuingishable. The two former microstates are the ones where the equal spins are opposite of each other, such that all the spins in the lattice have antiparallell neighbours, while in the latter case all of them will have one spin on either side that is parallell.

We can now define the degeneracy level as the number of microstates within one macrostate that yield that same energy. To illustrate the connection between the number of spins $s = +1$, the total energy and magnetisation of the system and the degeneracy level, we have given the possible combinations of these values in table I.

No. of $s = +1$	$\boldsymbol{E}(\mathbf{s})$ **[$\boldsymbol{J}$]**	$\boldsymbol{M}(\mathbf{s})$	**Degeneracy**
0	-8	-4	None
1	0	-2	4
2	8	0	2
	0		4
3	0	2	4
4	-8	4	None

TABLE I: Total energy $E(\mathbf{s})$ and magnetisation $M(\mathbf{s})$ for a 2×2 lattice with 0, 1, 2, 3 and 4 spins $s = +1$. Because we use periodic boundary conditions, the total energy can either be 0 J or 8 J when we have two spins $s = +1$ and two spins $s = -1$, depending on if the equal spins are neighbours or not. The last column contains the degeneracy level for the different combinations of number of spins $s = +1$ and total energy.

For a particular statistical ensemble, such as our Ising model, its statistical properties at a given temperature T when in thermodynamic equilibrium can be described by the partition function:

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

Here the sum is over all possible microstates $\mathbf{s}$ for a given a lattice size L, and we define the so-called "inverse temperature" β as

$$\beta = \frac{1}{k_\mathrm{B} T} \tag{7}$$

where k_B is the Boltzmann's constant. Since the exponent in eq. (6) must be dimensionless, we see that the temperature must have units of J/k_B. Furthermore, the probability of our system being in a given spin configuration $\mathbf{s}$ as a function of temperature T is then given by the Boltzmann distrubution:

$$p(\mathbf{s}; T) = \frac{1}{Z} e^{-\beta E(\mathbf{s})} \tag{8}$$

2. *Expectation Values*

Turning back to the 2×2 lattice described above, it is useful to analytically calculate the exact values of some properties that we must estimate numerically for a larger system, such that we can compare our program output in the 2×2 case to test our methods. First, let's calculate the partition function Z. We know from table I that there are two microstates with total energy $E(\mathbf{s}) = -8\ J$ and two with $E(\mathbf{s}) = 8\ J$, while there are twelve with zero total energy. Thus, summing these together and using that $e^0 = 1$ we have

$$Z = 12 + 2\left(e^{8J\beta} + e^{-8J\beta}\right) = 12 + 4\cosh(8J\beta) \tag{9}$$

To investigate how properties of our system change with temperature, magnetisation, energy and lattice size, we will be studying its specific heat capacity at constant volume C_V and magnetic susceptibility χ. When they both are normalised to the number of spins N, these are given by

$$C_V = \frac{N}{k_\mathrm{B} T^2}\Big(\left\langle \epsilon^2 \right\rangle - \langle \epsilon \rangle^2\Big) \tag{10}$$

$$\chi = \frac{N}{k_\mathrm{B} T}\Big(\left\langle m^2 \right\rangle - \langle |m| \rangle^2\Big) \tag{11}$$

with units $[C_V] = k_\mathrm{B}$ and $[\chi] = J^{-1}$. Here $\langle \epsilon \rangle$, $\left\langle \epsilon^2 \right\rangle$, $\langle |m| \rangle$ and $\left\langle m^2 \right\rangle$ are the expectation values of the total energy per spin, squared total energy per spin, absolute value of the total magnetisation per spin and squared total magnetisation per spin of the system respectively.

In general, for a given macrostate these are defined as

$$\langle \epsilon \rangle = \frac{1}{Z} \sum_{\mathbf{s}} \epsilon(\mathbf{s}) e^{-\beta E(\mathbf{s})} \tag{12}$$

$$\left\langle \epsilon^2 \right\rangle = \frac{1}{Z} \sum_{\mathbf{s}} \left(\epsilon(\mathbf{s})\right)^2 e^{-\beta E(\mathbf{s})} \tag{13}$$

$$\langle |m| \rangle = \frac{1}{Z} \sum_{\mathbf{s}} |m(\mathbf{s})| e^{-\beta E(\mathbf{s})} \tag{14}$$

$$\left\langle m^2 \right\rangle = \frac{1}{Z} \sum_{\mathbf{s}} (m(\mathbf{s}))^2 e^{-\beta E(\mathbf{s})} \tag{15}$$

Which, for the 2×2 lattice gives us

$$\begin{aligned}
\langle \epsilon \rangle &= \frac{1}{12 + 4\cosh(8J\beta)} \sum_{\mathbf{s}} \epsilon(\mathbf{s}) e^{-\beta E(\mathbf{s})} \\
&= \frac{1}{12 + 4\cosh(8J\beta)} 2\left(-2Je^{8J\beta} + 2Je^{-8J\beta}\right) \\
&= -\frac{2J\sinh(8J\beta)}{3 + \cosh(8J\beta)}
\end{aligned} \tag{16}$$

and similarly

$$\left\langle \epsilon^2 \right\rangle = \frac{4J^2\cosh(8J\beta)}{3 + \cosh(8J\beta)} \tag{17}$$

$$\langle |m| \rangle = \frac{2 + e^{8J\beta}}{6 + 2\cosh(8J\beta)} \tag{18}$$

$$\left\langle m^2 \right\rangle = \frac{1 + e^{8J\beta}}{6 + 2\cosh(8J\beta)} \tag{19}$$

Inserting this into equations eq. (10) and eq. (11) and introducing the factor $\gamma = 3 + \cosh(8J\beta)$ gives us

$$C_V = \frac{16J^2}{k_\mathrm{B}T^2}\left(\frac{\cosh(8J\beta)}{\gamma} - \frac{\sinh^2(8J\beta)}{\gamma^2}\right) \tag{20}$$

$$\chi = \frac{1}{k_\mathrm{B}T}\left(\frac{2 + 2e^{8J\beta}}{\gamma} - \frac{4 + 4e^{8J\beta} + e^{16J\beta}}{\gamma^2}\right) \tag{21}$$

We will use this 2×2 case to test the validity of our numerical results and to determine how many Monte Carlo cycles we need to perform.

3. *Energy Shifts*

As mentioned in the introduction, we will be using the Markov Chain Monte Carlo method, explained in detail in section II B, for our numerical calculations. We will therefore need to repeatedly use the Boltzmann factor $e^{-\beta \Delta E}$, where

$$\Delta E = E_\mathrm{after} - E_\mathrm{before} \tag{22}$$

is the energy shift due to flipping a single spin. To make our program as efficient as possible we will need to avoid calculating this value every time we use the Boltzmann factor. Let us now imagine an $L \times L$ lattice with $L > 2$, where the spins are denoted $s_{i,j}$ such that the spins at the upper left, upper right, lower left and lower right corners are $s_{0,0}$, $s_{0,L-1}$, $s_{L-1,0}$ and $s_{L-1,L-1}$ respectively. Say we now flip the spin of an arbitrary $s_{i,j}$. From eq. (2) we see that the energy shift will be

$$\Delta E = \pm 2J\big(s_{i+1,j} + s_{i,j+1} + s_{i-1,j} + s_{i,j-1}\big) \tag{23}$$

where the sign corresponds to the spin going from $s_{i,j} = \pm 1$ to $s_{i,j} = \mp 1$. Furthermore, we know from the 2×2 case that the sum of four spins are ± 4, ± 2 or 0. Thus, the possible energy shifts are

$$\Delta E(+-, +4) = \Delta E(-+, -4) = 8\,J \tag{24}$$

$$\Delta E(+-, +2) = \Delta E(-+, -2) = 4\,J \tag{25}$$

$$\Delta E(\mp\pm, 0) = 0\,J \tag{26}$$

$$\Delta E(+-, -2) = \Delta E(-+, +2) = -4\,J \tag{27}$$

$$\Delta E(+-, -4) = \Delta E(-+, +4) = -8\,J \tag{28}$$

where the $+-$ means that the spin flips from $+1$ to -1 and $+4$ means that the sum of the neighbouring spins are 4, etc.

4. *Phase Transitions and Critical Temperature*

The size-dependent critical temperature $T_c(L)$ of a 2D Ising model holds significant importance in the field of statistical physics, and describes the temperature at which the system undergoes a phase transition from an ordered configuration with all or most of the spins aligned (ferromagnetism), to an unordered configuration with spins pointing in more or less random directions (paramagnetism). In the context of this work, we aim to explore further the behaviour of the critical temperature as a function of the lattice size L. To achieve this, we will utilize numerically calculated values of $\langle\epsilon\rangle$, $\langle|m|\rangle$, C_V and χ for increasing lattice sizes and decreasing temperature ranges to determine at which temperature we see rapid changes.

An infinite 2D Ising model is of great interest in statistical physics, and in 1944 Lars Onsager made a groundbreaking discovery when he derived an analytical expression of its critical temperature:

$$T_c(L = \infty) = \frac{2}{ln(1 + \sqrt{2})} J/k_B \approx 2.269 J/k_B \tag{29}$$

We will attempt to use our numerical results to estimate this value, and to do this we will be utilising the following expected relation

$$T_c(L) - T_c(L = \infty) = aL^{-1} \tag{30}$$

$$\Rightarrow \quad T_c(L = \infty) = T_c(L) - aL^{-1} \tag{31}$$

where a is a constant.

B. Algorithms

Monte Carlo (MC) methods are a category of computational techniques that rely on random sampling to solve problems and make numerical estimations. Whether simulating complex physical systems, optimizing processes, or estimating mathematical quantities, MC-methods use probabilistic sampling to obtain results. The strength of MC lies in its ability to handle problems with a large number of variables or intricate interactions, providing approximate solutions through statistical sampling rather than exhaustive computation. Its applications range from numerical approximations of integrals for complex functions to studying the behaviour of particles in a system, such as the (in this case, 2D) Ising model or the behaviour of gases.

We will be using a specific type of Monte Carlo method in our numerical calculations which is very common when working with the 2D Ising model, called the Metropolis algorithm. Briefly summarised in algorithm 1, this is particularly well-suited for simulating the behaviour of physical systems with large numbers of interacting elements. By iteratively proposing changes to the spin configurations and probabilistically accepting or rejecting these changes based on a defined acceptance criterion, such as the Boltzmann factor, the algorithm efficiently explores the equilibrium time of spin configurations.

When calculating the energy of a specific spin configuration, we were able to remove the need for if-statements or the use of the modulus operator in the nested loop over the i and j indices, which is a common approach. We did this by using the fact that a sum can be re-ordered to give

Algorithm 1 The Metropolis Algorithm

```
procedure MONTE CARLO CYCLE($\mathbf{s}, L, \beta$)
    for $i = 0, 1, \dots, L-1$ do
        for $j = 0, 1, \dots, L-1$ do
            ▷ Compute energy difference due to flipping $s_{ij}$
            $\Delta E \leftarrow \Delta E_{\text{function}}(\mathbf{s}, i, j)$

            ▷ Flip if energy difference is negative or zero
            if $\Delta E \leq 0$ then
                $s_{ij} = -s_{ij}$                          ▷ Flip spin

            else                         ▷ Energy difference is positive
                $w = e^{-\beta \Delta E}$          ▷ Probability of flipping spin
                ▷ Generate random number $r \in [0, 1]$
                ▷ Flip spin if $r \leq w$
                if $r \leq w$ then
                    $s_{ij} = -s_{ij}$                      ▷ Flip spin

            ▷ Calculate $E$, $E^2$, $|M|$ and $M^2$ for $\mathbf{s}$
            $E$, $E^2 = E_{\text{function}}(\mathbf{s})$
            $|M|$, $M^2 = M_{\text{function}}(\mathbf{s})$
            ▷ Update expectation values accordingly
```

the same result. We therefore moved the special cases of $j = i$, outside the innermost loop, and the boundary points outside the entire loop. In section II A 3 we also determined that the possible positive energy differences ΔE to be either 4 or 8, and could therefore pre-calculate the Boltzmann factors w_4 and w_8 for these cases.

C. Numerical Calculations

1. *Specific Approaches*

We will begin by validating our numerical calculations by testing our program on a 2×2 lattice. By comparing our analytical solution with the numerical results, we will explore how many Monte Carlo cycles are needed for the program to converge towards the analytical results. We will continue by parallelising our program using OpenMP, and from there attempt to estimate a resulting speed-up factor. We will parallellise the calculations of expectation values of a 20×10 lattice when using 50 000 Monte Carlo cycles with respect to 10 different temperatures in the range $T \in [2.1, 2.4]$ J/k_{B}, where each thread is assigned a temperature.

Once we know our program performs as expected, we will continue working on the 20×20 lattice to study its equilibration time in terms of Monte Carlo cycles as function of temperature. We will do this by first numerically estimating $\langle \epsilon \rangle$ and $\langle |m| \rangle$ for each Monte Carlo cycle and writing the results to files to be plotted in Python. We will do this for four different cases:

1. An ordered initial spin configuration where all spins are positive with temperature $T = 1.0$ J/k_{B}
2. A randomised initial spin configuration with temperature $T = 1.0$ J/k_{B}
3. An ordered initial spin configuration where all spins are positive initial spin configuration with temperature $T = 2.4 J/k_B$
4. A randomised initial spin configuration with temperature $T = 2.4 J/k_B$

Furthermore, we will not just study the expectation value, but also the probability distribution of ϵ as a function of temperature. We will estimate this by sampling the energy per spin of every spin configuration we encounter during each step in each Monte Carlo cycle we perform for the randomised cases, and writing the samples to files. We will then visualise the results as normalised histograms where each bin corresponds to distinct ϵ-values sampled.

Finally, we study the critical temperatures $T_c(L)$ for increasing lattice sizes, and attempt to estimate $T_c(L = \text{inf})$ ourselves. We do this by plotting estimates of $\langle \epsilon \rangle$, $\langle |m| \rangle$, C_V and χ for lattice sizes $L \in [40, 60, 80, 100]$ and temperature values in the range $T \in [2.1, 2.4]$ J/k_B, and from there estimate a narrow temperature range in which the critical temperatures might lie. After determining this range we estimate $\langle \epsilon \rangle$, $\langle |m| \rangle$, C_V and χ again with a smaller gap from temperature to temperature, and then plot these results. We expect C_V and χ to spike at T_c, and we therefore calculate the mean of the two temperatures where we observe these spikes. We then use eq. (31) to estimate $T_c(L = \text{inf})$ for each lattice, and set out estimate to the mean of these values. Lastly we wish to compare our estimate to the analytical result by Onsager.

2. *Program Structure*

Our program is divided into the files `main.cpp`, `energy.cpp`, `heatcap_suscept.cpp`, `anal.cpp`, `test.cpp` and `MCMC.cpp`. The first of these is the main program, where we use the functions from the other files to perform various calculations. A summery of their respective contents are given below:

`energy.cpp`:

> The `energy.cpp` file contains the energy function which calculates the total energy of a given spin configuration. The input arguments to this function is s and L. L is the number of rows and columns in our s lattice, containing $L \times L$ spin particles. The `energy(s,L)` function calculates the total energy of the lattice, by looking at the spin of each particle. Each particle is either in a state of spin-up or down, which is represented by either a 1 or -1 in the lattice.

`heatcap_suscept.cpp`:

The `heatcap_suscept.cpp` file contains two simple functions for χ and C_v. The input arguments for C_v is `n_spins`, which is the number of particles, `T`, the temperature and `exp_`ϵ and `exp_`ϵ`sq` which are the energy expectation values. For χ it is the same, but magnetisation expectation values instead of energy. `heatcap_suscept.cpp` simply uses equation 10 and 11 to solve for χ and C_v.

`anal.cpp`:

This file contains four functions with their only input-arguments being β, the inverse temperature. From this there are four function for analytically calculating each of the values $\langle\epsilon\rangle$, one for $\langle|m|\rangle$, C_v and χ.

`test.cpp`:

To be sure that the numerical work stands correct, there is a test function. This function, when called, compares the energy calulated both numerically and analytically of all the 2×2 lattices shown in table I. To make sure it worked for larger lattices, we added a test for a 4×4 lattice as well. If the results match the function return true and otherwise it returns false.

`MCMC.cpp`:

The `monte_carlo_cycle` function, in the `MCMC.cpp` file, executes a single Monte Carlo cycle for a spin system. It processes a square matrix `s` representing the spin configuration with lattice size `L`, utilizing weights `w4` and `w8` for energy differences. The function iterates through spins, calculates energy and magnetization changes upon potential spin flips, and applies the Metropolis criterion to decide flips probabilistically. It also inputs a random number generator, with a specific seed so the outcome is random, but regenerative. Tracking temporary and cumulative values for energy and magnetization, it stores energy values in a matrix ϵ. Optionally, the function appends stored energy values to a specified file (`filename`). This function is vital for simulating spin system behavior in the context of investigating phase transitions and thermodynamic properties.

(see section II B)

Moreover, the file `main.cpp` is structured as follows:

1. First we use the test function `test_energy` to check that the energy function correctly calculates the energies of all the configurations in table I, to see if the numerical values match the analytical ones.

2. We move on to initialising a randomised 2×2 spin configuration matrix with a specific seed so that we are able to compare the results between runs. We perform 1 000 Monte Carlo cycles using `monte_carlo_cycle` from `MCMC.cpp` over a range of temperatures, and for every 100th cycle we update and print the estimated expectation values $\langle E\rangle$, $\langle E^2\rangle$, $\langle\epsilon\rangle$, $\langle\epsilon^2\rangle$, $\langle|M|\rangle$, $\langle M^2\rangle$, $\langle|m|\rangle$ and $\langle m^2\rangle$, as well as the resulting estimates of C_V and χ. At the end we calculate the analytical expectation values, heat capacity and susceptibility using the functions from `anal.cpp` and print these as well for comparison.

3. Next we study the burn-in time and probability function $p_\epsilon(\epsilon;T)$ by initialising both ordered and unordered 20×20 spin configurations. We also studied the burn-in time for both unordered and ordered spin configurations for two temperatures, $T_1 = 1.0$ and $T_2 = 2.4$. To study the probability function $p_\epsilon(\epsilon;T)$ we performed 100 000 Monte Carlo cycles, for both temperatures with unordered configurations. We chose to investigate the unordered configuration, because it is the most natural configuration. We then plotted the probability function $p_\epsilon(\epsilon;T)$ for both temperatures, using `matplotlib.pyplot`.

4. In all instances where Monte Carlo simulations were employed, parallelization was implemented using `OpenMP`. We systematically timed all numerical calculations, comparing the performance of single-threaded computations against parallelized ones. This allowed us to assess the time efficiency of our simulations.

5. We explored the potential presence of a phase transition by expanding the lattice, while investigating a greater temperature range. Parallelisation proved useful in significantly reducing computation time for these extensive calculations. We recorded the values of C_v and χ, saving them to file for plotting.

6. We wanted to determine the critical temperature of an infinite ($L = \infty$) 2D Ising model. Starting with known critical temperatures from prior work, we reconsidered equation 29, reformulating it as equation 30. This allowed us to approximate a, and by averaging the critical temperatures, we obtained the critical temperature for the infinite Ising model.

Listed below are the four different Python scripts we have used to visualise our results:

`numerical_analytical.py`:

The purpose of this script is to plot numerical estimates for $\langle\epsilon\rangle$, $\langle|m|\rangle$, C_V and χ as functions of temperature for the 2×2 lattice and an increasing number of Monte Carlo cycles, along with the corresponding analytical values. The data is read out from a specified file, where the first column contains temperature values and the last four contain the values for the four physical quantities specified above.

`equilibration.py`:

Here we plot the estimates of $\langle\epsilon\rangle$ and $\langle|m|\rangle$ for the 20×20 lattice that are calculated in our main program as functions of Monte Carlo cycles. The estimates from ordered and unordered initial configurations are plotted together, with subplots seperating energy and magnetisation in their respective rows, and the two columns representing the two temperatures $T = 1.0\ J/k_{\mathrm{B}}$ and $T = 2.4\ J/k_{\mathrm{B}}$ respectively.

`probability.py`:

In this program we read the `.txt`-files with ϵ- and m-samples generated in our main program, and create normalised histograms for these to approximate the probability functions $p_\epsilon(\epsilon; T)$ and $p_m(m; T)$ for the same temperatures as in `equilibration.py`.

`phase_transition.py`:

Here we plot numerical estimates of $\langle\epsilon\rangle$, $\langle|m|\rangle$, C_V and χ as functions of temperature similarly to in `numerical_analytical.py`, but this time for different lattice sizes and a fixed number of Monte Carlo cycles. The data is read from a specified file here as well, but here the lists of values are sorted to be in order of increasing temperature, as the orders are not perfect due to different threads writing to file simultaneously.

In this script we also estimate the critical temperatures $T_c(L)$ by finding the temperatures where C_V and χ have their peaks for the respective lattice size, and computing the mean of these. Relation (31) is then used to estimate the temperature of an infinite Ising model.

3. *Tools*

C++ [3] will be used for the numerically heavy computations. The standard library of C++ will be used for random number generation, and trigonometric functions. As containers of data and for vectorisation, we will implement Armadillo's [4] `arma::vec` class, together with other functions the library offer. To paralellise our code we used the OpenMP [2] library. Python [5] will be used to for lighter computations and all plots. To vectorise our code we use `numpy` [6], and for visualisation we use `matplotlib.pyplot` [7]. To create a linear fit of data and calculate errors, we use `scipy.linregress` [8]. Code completion and debugging is done in Visual Studio Code [9] with additional assistance of GitHub Copilot [10]. We use `git` [11] for version control, and `GitHub` [12] for remote storage of our code.

III. RESULTS & DISCUSSIONS

A. Effects of Parallellisation

When executing our program across various lattice sizes and over a range of temperatures, we observed a notable reduction in runtime by employing OpenMP. The time elapsed for different lattice sizes and thread configurations is presented in **??**, considering a temperature range $T \in [2.1, 2.4], J/k_{\mathrm{B}}$.

$L \times L$	1 Thread [s]	4 Threads [s]
40×40	34.4708	25.3691
60×60	70.725	59.4863
80×80	150.161	111.417
100×100	232.406	161.98

TABLE II: Time elapsed for different lattice sizes and thread configurations, with a temperature range $T \in [2.1, 2.4]J/k_{\mathrm{B}}$, with 50 000 Monte Carlo cycles per temperature.

To determine the speed-up factor, we added the times for one-thread computations and divided them by the times for four-thread computations, resulting in an estimated speed-up factor of 1.36. The number of available threads on a computer impacts the speed-up factor. A computer with only four available threads can parallelize the computation into four threads, whereas a computer with eight available threads can parallelize it into eight threads. Consequently, the speed-up factor should be higher for a computer with eight available threads compared to a computer with four available threads.

B. Expectation Values

1. *Comparing with Analytical Results*

When running our program with the 2×2 lattice at temperature $T = 1.0\ J/k_{\mathrm{B}}$ we were able identify that we would need to perform at least 10 000 Monte Carlo cycles before our results stabilised properly. We also noticed that our numerical results stabilised on values that differed from the analytically calculated values, although their magnitudes seemed reasonable. This is evident in table III, which shows our numerical estimates of $\langle\epsilon\rangle$, $\langle|m|\rangle$, C_V and χ for the 2×2 lattice after differing numbers of Monte Carlo cycles. For example, the susceptibility normalised to the number of spins would converge to around $0.0021 - 0.0023\ J^{-1}$, while the numerical result was $0.0040\ J^{-1}$, which is of the same order of magnitude but yields a relative error of about 50%.

No. of cycles	$\langle\epsilon\rangle$ $[J]$	$\langle\lvert m\rvert\rangle$	C_V $[k_\mathrm{B}]$	χ $[J^{-1}]$
10	−1.8000	0.9375	1.4400	0.1594
20	−1.9000	0.9688	0.7600	0.0836
50	−1.9600	0.9875	0.3136	0.0344
100	−1.9800	0.9938	0.1584	0.0173
200	−1.9775	0.9931	0.1780	0.0186
500	−1.9880	0.9962	0.0954	0.0104
1000	−1.9940	0.9981	0.0479	0.0052
5000	−1.9960	0.9988	0.0319	0.0033
10000	−1.9974	0.9992	0.0204	0.0022
100000	−1.9975	0.9992	0.0197	0.0022
1000000	−1.9973	0.9992	0.0215	0.0023
Analytical	−1.9960	0.9987	0.0321	0.0040

TABLE III: Numerical estimates of $\langle\epsilon\rangle$, $\langle|m|\rangle$, C_V and χ for $T = 1.0\ J/k_\mathrm{B}$ after increasing numbers of Monte Carlo cycles are performed. The last row contains the analytical values.

To further investigate how the discrepancies between the numerical and analytical results would evolve with temperature we also calculated and logged our numerical estimates of $\langle\epsilon\rangle$, $\langle|m|\rangle$, C_V and χ for 100 temperature values in the interval $T \in [1.0, 2.4]\ J/k_\mathrm{B}$ after 100, 1 000, 10 000 and 100 000 Monte Carlo cycles. The result is plotted along with the analytical values in fig. 2. We see clearly that the numerical values follow a consistent temperature-dependent trend after 10 000 − 100 000 cycles, but that these deviate from the analytical values. It is also obvious that these deviations increase with increasing temperature, which we need to take into consideration when estimating the critical temperatures at different lattice sizes, as the reason for the discrepancies most likely lie within our numerical methods.

2. Equilibration Time

When estimating the equilibration time of a 20 × 20 lattice we found that this was noticeabley larger at $T = 2.4\ J/k_\mathrm{B}$ than at $T = 1.0\ J/k_\mathrm{B}$, as evident in fig. 3. For the latter it appeared to be only a few Monte Carlo cycles compared to how many we plotted for, while for the former 80000 − 100000 cycles seemed more realistic. This is shown more clearly in **??**, where we have zoomed in on the y-axes around the converging values, and focused on a reasonable number of Monte Carlo cycles for the respective temperatures. We see that for $T = 1.0\,J/k_\mathrm{B}$ the expectation values for the initially ordered and unordered configurations meet and stabilise after about 8000 cycles, while for $T = 2.4\,J/k_\mathrm{B}$ this isn't the case until after cirka 80 000 cycles are completed.

Not surprisingly, the estimates made from the initially unordered configurations start at values that diverge radically from the expectation values in the low temperature case. This is because the expectation values of this case align closely with the lowest energy configurations, which are quite few, even for the 2 × 2 lattice as we saw in table I. Since the ordered initial configuration had all positive spins, which yields $\epsilon = -2.0\ J$ and $|m| = 1$, the expectation value was naturally estimated almost immediately. Moreover, for the high temperature case both configurations were far off at the beginning, fluctuating quite a lot before eventually stabilising on the estimates $\langle\epsilon\rangle = -1.236\ J$ and $\langle|m|\rangle = 0.452$.

C. Probability Functions

The upper part of fig. 5 shows the estimated normalised probability functions $p_\epsilon(\epsilon; T = 1.0\ J/k_\mathrm{B})$ (left) and $p_\epsilon(\epsilon; T = 2.4\ J/k_\mathrm{B})$ (right) for the energy per spin ϵ after generating 40 000 000 samples for each. These are in good agreement with the amount of fluctuations observed in fig. 4, as we see a narrow spike at one of the lowest energies on the left, and a broader Gaussian-like distribution on the right, both with their peaks seemingly centered on the estimated expectation values. The latter may also indicate the presence of some numerical miscalculations, as the bell curve is slightly tilted, falling more quickly to ~ 0 on the right side than on the left. We attempted to run our program multiple times, collecting as few as 40 000 samples to as many as 400 000 000, but the tilted curve was consistently present.

In an attempt to investigate the reason for the consequent shape of $p_\epsilon(\epsilon; T = 2.4\ J/k_\mathrm{B})$ we decided to store the magnetisations per spin for all the same configurations, and similarly estimate the probability functions for the magnetisation per spin to see if these also would yield peculiar results. The resulting estimates of $p_m(m; T = 1.0\ J/k_\mathrm{B})$ and $p_m(m; T = 2.4\ J/k_\mathrm{B})$ are presented in the bottom left and bottom right plots in fig. 5 respectively. We see that the latter has an expected shape, with peaks at seemingly equal distance from $m = 0$ on either side, and a through at this value. However, the negative peak is higher than the positive peak, which could be connected to the shape of the energy probability function at same temperature. Nevertheless, this shape is easier to explain. It is most likely caused by the fact as the number of performed Monte Carlo cycles increases, the probability of a spin flip minimising the energy of the system decreases. This consequently leads to a decrease in the probability of the magnetisation per spin changing, let alone go from being negative to positive. This is also confirmed by that the low temperature probability function has an equally sharp spike as $p_\epsilon(\epsilon; T = 1.0\,J/k_\mathrm{B})$, but at around $m = 1$, meaning that we never end up flipping so many spins that

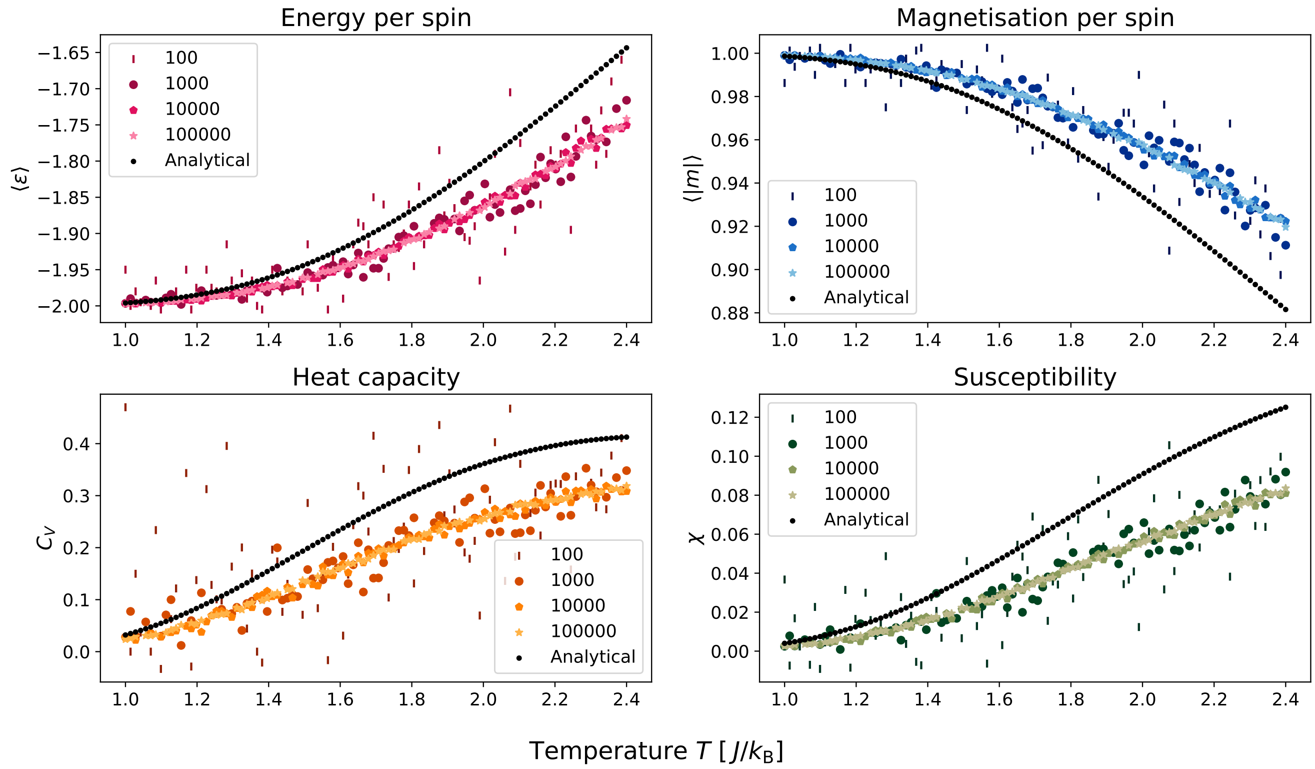

FIG. 2: Numerical estimates of $\langle\epsilon\rangle$, $\langle|m|\rangle$, C_V and χ in the temperature range $T \in [1.0, 2.4]$ J/k_B] for the 2×2 lattice after 100, 1 000, 10 000 and 100 000 Monte Carlo cycles. Lighter values correspond to a larger number of cycles, and the black dots are the corresponding analytical values.

m goes from being positive to negative after surpassing a certain number of Monte Carlo cycles.

D. Phase Transitions and Critical Temperature

1. *Lattices of Increasing Size*

When running 100 000 Monte Carlo cycles at temperatures in the range $T \in [2.1, 2.4]$ J/k_B for lattice sizes $L \in [40, 60, 80, 100]$ we estimated the values of $\langle\epsilon\rangle$, $\langle|m|\rangle$, C_V and χ plotted in fig. 6. In the upper right plot of $\langle|m|\rangle$-values we see that these suddenly decrease rapidly at around $T = 2.25\,J/k_\mathrm{B}$ for all our Ising models. This is a clear indication of the transition from the ferromagnetic phase to the paramagnetic phase. Furthermore, we see that the decrease is steeper for the larger lattices, which we would expect since there are more spins interacting with each other in a large lattice and therefore more spin interactions collectively going from ordered to disordered structures, leading to more dramatic changes in the net magnetisation.

Moreover, the graphs for the heat capacity C_V and susceptibility χ follow the expected trend, although there are some seemingly random fluctuations in C_V and χ for the largest lattice in the low temperature range. This could be because we should have performed more Monte Carlo cycles, as the larger the lattice the more cycles are necessary to properly estimate the expectation values. However, there is no confusion where the interesting parts of the graphs are, as we see peaks around $T \in [2.20, 2.35]$ J/k_B for all four lattices, which is a clear indicator of the critical temperatures $T_c(L)$ being within this range. We therefore chose to study this range in more detail, and thus performed 100 000 Monte Carlo cycles for 50 temperature values within this slightly narrower range as well. The resulting values are plotted and shown in **??**.

From the plots of C_V and χ in the narrower range we see that the critical temperatures all reside somewhere between $2.26 - 2.30$ J/k_B, being somewhat skewed towards smaller values for the larger lattice. This makes sense considering the critical temperature of the infinite Ising model. Furthermore, by calculating the mean of the temperatures where we observe spikes in C_V and χ for the respective lattice sizes, we were able to estimate the following critical temperatures:

$$T_c(40) \approx 2.287\ J/k_\mathrm{B} \tag{32}$$

$$T_c(60) \approx 2.272\ J/k_\mathrm{B} \tag{33}$$

$$T_c(80) \approx 2.275\ J/k_\mathrm{B} \tag{34}$$

$$T_c(100) \approx 2.266\ J/k_\mathrm{B} \tag{35}$$

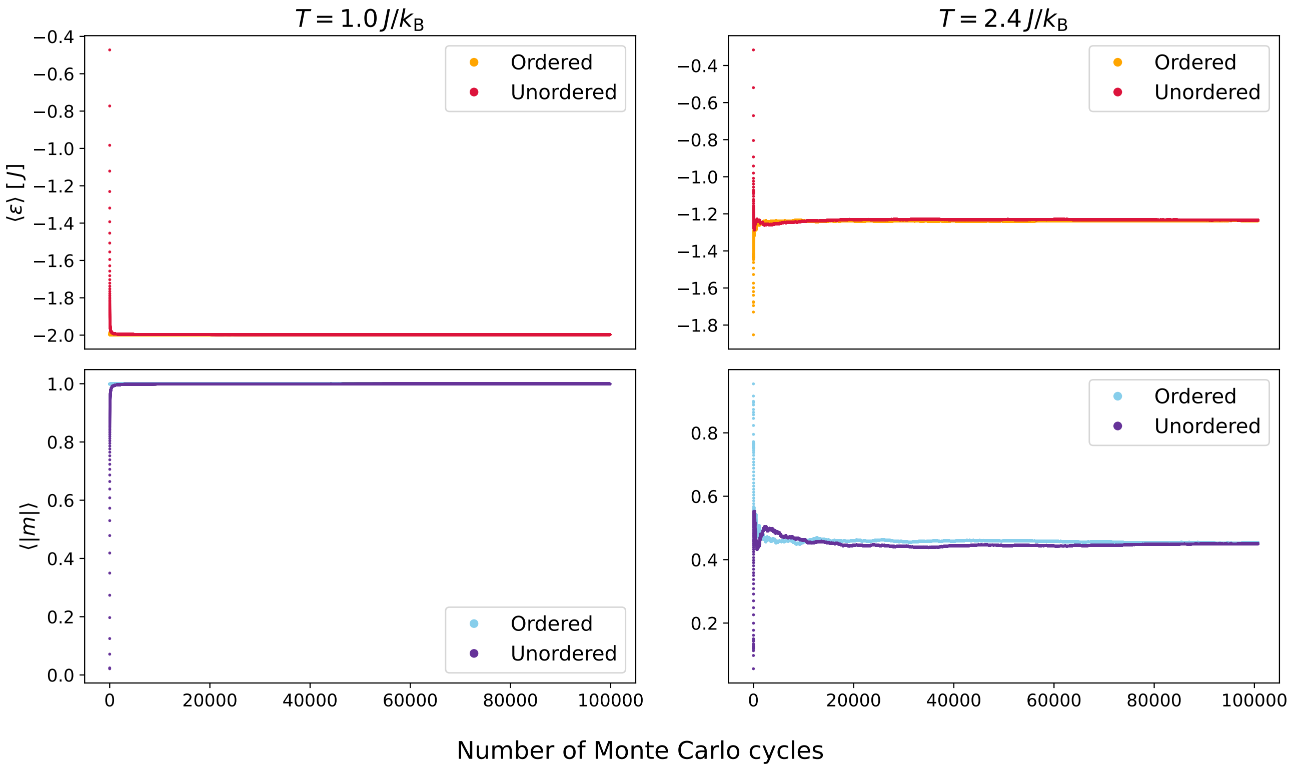

FIG. 3: Numerical estimates of $\langle\epsilon\rangle$ (upper row) and $\langle|m|\rangle$ (lower row) for a 20×20 lattice as functions of number of Monte Carlo cycles performed for $T = 1.0\ J/k_\mathrm{B}$ (left) and $T = 2.4\ J/k_\mathrm{B}$ (right). Estimates calculated from ordered initial spin configurations with all spins being positive are plotted in light color, while the estimates calculated from unordered initial configurations are plotted with dark colors.

2. *The Infinite Ising Model*

When estimating the critical temperature of the infinite 2D Ising model, we used eq. (31) to calculate four values from the critical temperatures of the finite lattices, and computed the mean and standard deviation of these to get

$$T_c(L = \infty) \approx 2.259 \pm 0.003\ J/k_\mathrm{B} \tag{36}$$

It is important to note that we would expect the critical temperatures to decrease towards Onsager's result as the lattice size increases towards infinity, but never approach it. However, we notice that the estimate of the critical temperature of the 100×100 lattice actually is smaller than the analytical result for the infinite Ising model, and it therefore makes sense that our estimate is too low. We could have gone further into estimating the constant a and maybe achieve a more accurate result, but this seemed redundant as changing the magnitude of this constant also significantly increased the uncertainty in our estimate of $T_c(L = \infty)$. Additionally, there was a clear sign of numerical miscalculations in our results presented in section III B 1, which definitely transferred to these estimates as well. There is definitely a high chance that these discrepancies are the reasons for our inaccurate estimate.

IV. CONCLUSION

We have studied the 2D Ising model to investigate the temperature-dependent magnetic behaviour of spin particle configurations. Through the implementation of the Markov Chain Monte Carlo method and the parallelization of our program using OpenMP, we were able to perform numerical estimates of expectation values for energy $\langle\epsilon\rangle$, magnetisation $\langle|m|\rangle$, heat capacity C_V and susceptibility χ of $L \times L$ lattices with sizes ranging from $L = 2$ to $L = 100$, the former of which were compared to analytical results. These results deviated from the analytical values, especially at larger temperatures, indicating flaws in our numerical algorithms. Nevertheless, we were able to decipher that we needed approximately of 100 000 Monte Carlo cycles for overall result stability, as equilibrium time analyses showed clear temperature dependencies, specifically longer stabilisation periods observed at higher temperatures.

Though flawed, our numerical results supplied us with qualitative understanding of the phase transition from ferromagnetism to paramagnetism. By studying the temperature dependencies of $\langle\epsilon\rangle$, $\langle|m|\rangle$, C_V and χ for lattice sizes $L \in [40, 60, 80, 100]$, we were able to estimate the critical temperatures $T_c(L)$ at which the re-

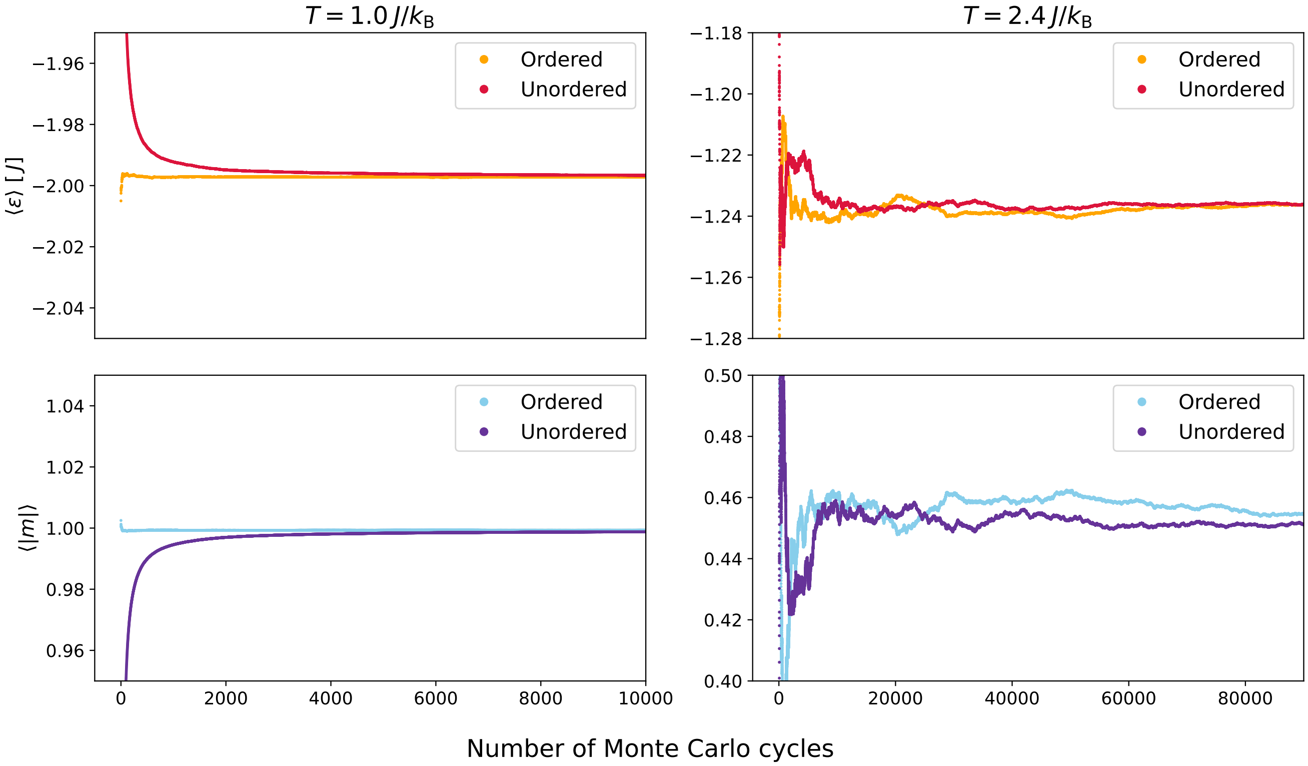

FIG. 4: Numerical estimates of $\langle\epsilon\rangle$ (upper row) and $\langle|m|\rangle$ (lower row) for a 20×20 lattice as functions of number of Monte Carlo cycles performed for $T = 1.0\,J/k_\mathrm{B}$ (left) and $T = 2.4\,J/k_\mathrm{B}$ (right), zoomed in on both the x- and y-axes to properly study where the values stabilise. Estimates calculated from ordered initial spin configurations with all spins being positive are plotted in light color, while the estimates calculated from unordered initial configurations are plotted with dark colors.

spective models underwent phase transitions. We were also able to use these results to estimate the critical temperature of an infinite 2D Ising model to be $T_c(L = \infty) \approx 2.259 \pm 0.003\ J/k_\mathrm{B}$, although this value deviated more than we had hoped from the analytical result $T_c(L = \infty) \approx 2.269\,J/k_\mathrm{B}$ derived by Onsager, most likely due to the instabilities of our numerical algorithms.

REFERENCES

[1] S. Wald, "Schematic representation of a configuration of the 2d ising model on a square lattice," (Accessed: October 2023).

[2] O. A. R. Board, "Openmp v.4.5," (Accessed: October 2023).

[3] T. F. S. Foundation, "C++, gnu 12.3," (Accessed: October 2023).

[4] D. C. Sanderson and D. R. Curtin, "Armadillo c++ library," (Accessed: October 2023).

[5] P. S. Foundation, "Python 3.11.4 documentation," (Accessed: October 2023).

[6] N. Developers, "Numpy v1.22 manual," (Accessed: October 2023).

[7] M. D. Team, "Matplotlib: Visualization with python," (Accessed: October 2023).

[8] S. Community, "Scipy v1.8.0 reference guide," (Accessed: October 2023).

[9] Microsoft, "Visual studio code," (Accessed: October 2023).

[10] Microsoft, "Github copilot," (Accessed: October 2023).

[11] J. Hamano, "Git," (Accessed: October 2023).

[12] Microsoft, "Github," (Accessed: October 2023).

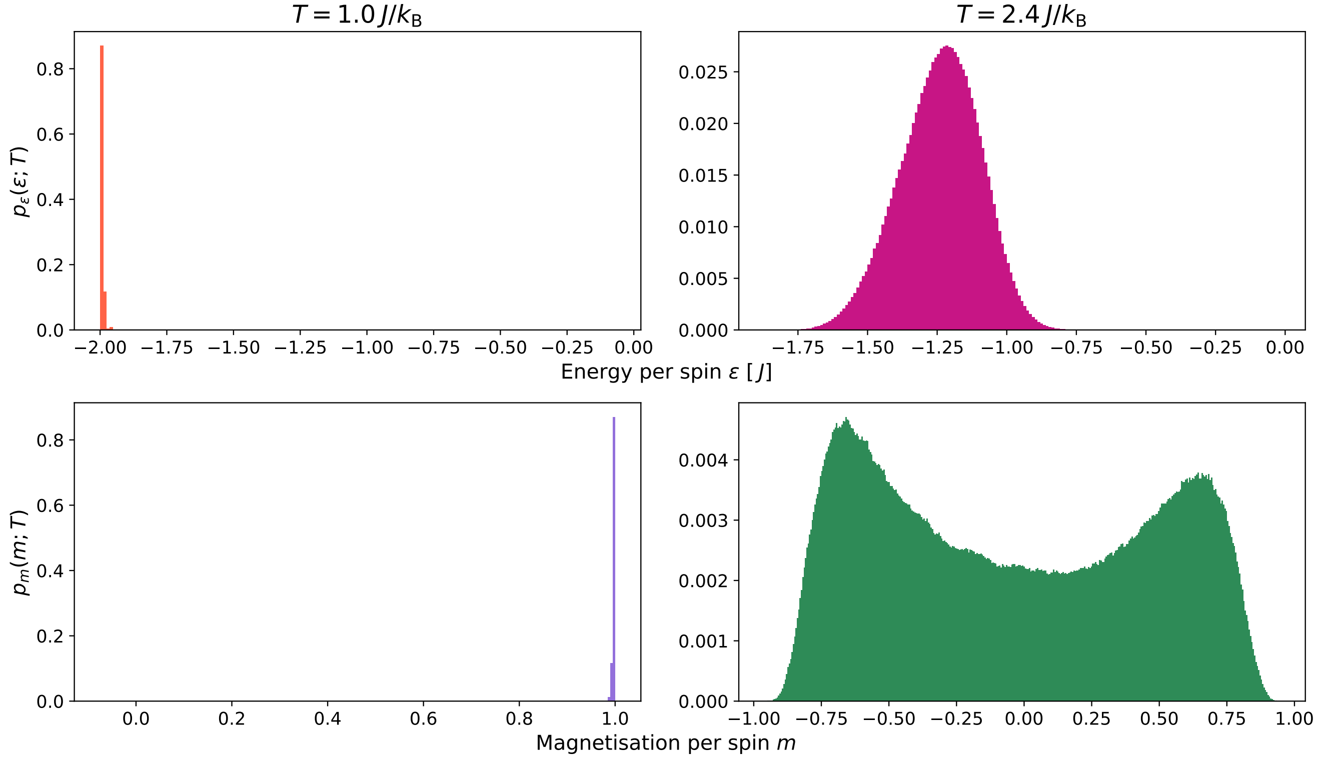

FIG. 5: Estimated probability functions $p_\epsilon(\epsilon; T)$ (upper row) and $p_m(m; T)$ (lower row) for temperatures $T = 1.0\ J/k_\mathrm{B}$ (left) and $T = 2.4\ J/k_\mathrm{B}$ (right) generated by creating normalised histograms from ϵ- and m-samples of a 20×20 lattice.

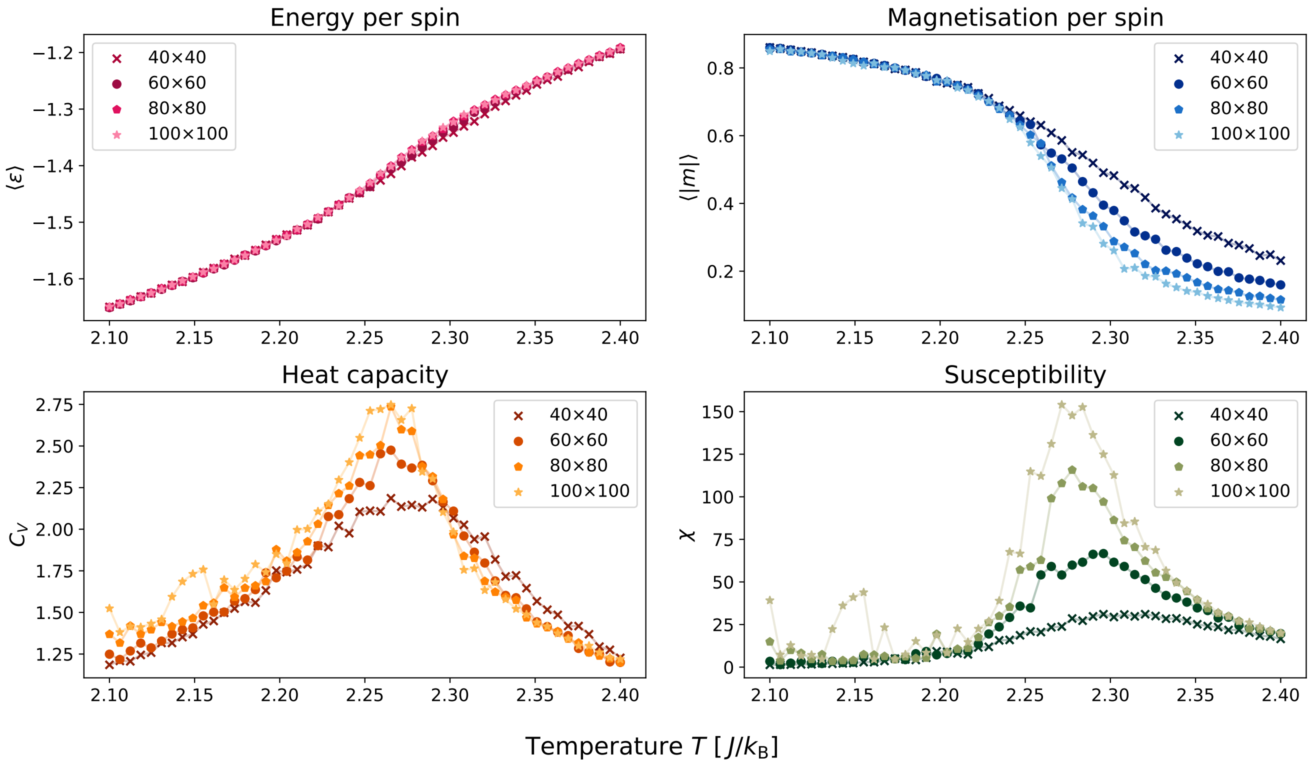

FIG. 6: Numerical estimates of $\langle\epsilon\rangle$, $\langle|m|\rangle$, C_V and χ for lattice sizes $L \in [40, 60, 80, 100]$ and temperatures $T \in [2.1, 2.4]$ J/k_B] after performing 100 000 Monte Carlo cycles. Lighter values correspond to larger lattice sizes.

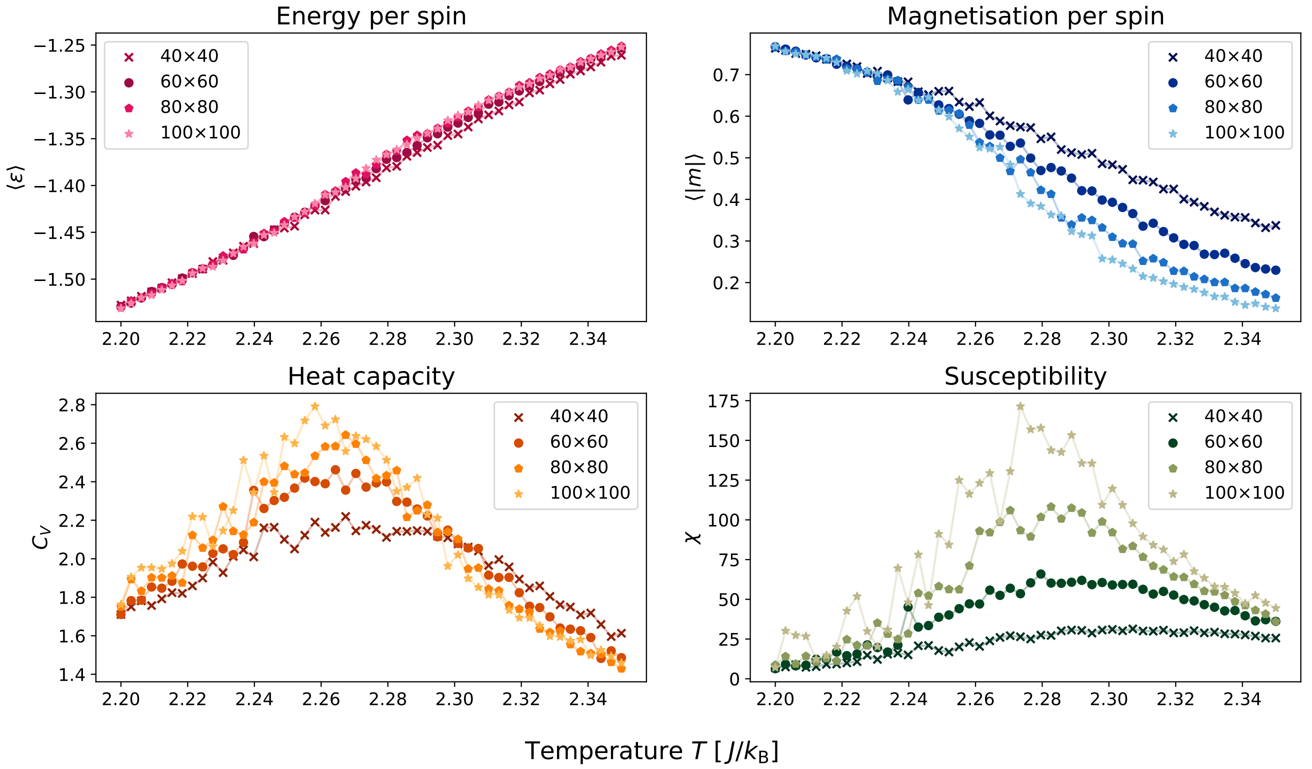

FIG. 7: Numerical estimates of $\langle\epsilon\rangle$, $\langle|m|\rangle$, C_V and χ for lattice sizes $L \in [40, 60, 80, 100]$ and temperatures $T \in [2.20, 2.35]$ J/k_B] after performing 100 000 Monte Carlo cycles. Lighter values correspond to larger lattice sizes.