# Simulating the Ising model using the Metropolis algorithm

FYS3150

Written by:

*Sigurd Kjelsbøl Huse and Odin Johansen*

Department of Physics UiO

UiO : University of Oslo

March 8, 2023

# Contents

## 1. Abstract

We use the Monte Carlo Metropolis method to model the phase transitions of the Ising model in a 2-dimensional grid with periodic boundary conditions. The first thing we do is produce analytical results with a simple $2 \times 2$-grid, this is a safety procedure to make sure the algorithm works. Doing this we get the following numerical approximations: $\langle\epsilon\rangle_{\text{num}} = -1.996[J], \langle|m|\rangle_{\text{num}} = 0.999[1], C_{V\,\text{num}} = 0.032[k_B], \chi_{\text{num}} = 0.004[1/J]$. With the analytical values being: $\langle\epsilon\rangle = -1.996[J], \langle|m|\rangle = 0.999[1], C_V = 0.032[k_B], chi = 0.004[1/J]$. Then we implement a $20 \times 20$-grid and examine the equilibration time for temperatures [1, 2.4]$J/k_b$ with random and ordered initial conditions. We count the number of times each energy occurs to find the probability distribution function, for different values of the energy per spin. Observing a smaller variance for lower temperatures and a higher variance for higher temperatures. Next, we simulate a $40 \times 40$, $60 \times 60$, $80 \times 80$ and $100 \times 100$ grid with temperatures in the range [2.1K-2.4K], and run everything through $2.0 \cdot 10^6$ MC cycles. To optimize the code, we parallelized it across 10 cores making the simulations approximately 6.5 times faster. For the phase transition, we find the critical temperature to be $2.2689[J/K_B]$ for an infinite grid using linear regression. With an analytical value of $2.269[J/K_B]$, our estimate of the critical temperature for an infinite grid is within uncertainties. Every code referenced in this paper can be found in the repository linked at the top right corner of every page.

## 2. Introduction

Magnetism is a phenomenon that has been perplexing humans for thousands of years. People in ancient Greece noticed that lodestones naturally attracted iron, they named it magnētis lithos meaning "The magnesian stone" and used it as a way of extracting arrowheads from inside the bodies of wounded soldiers. Much later, we discovered various types of magnetic properties such as paramagnetism, ferromagnetism, and diamagnetism. In this paper, we are only looking at ferromagnetism and statistical methods for modelling such a material. The Ising model is a model well suited for this. It consists of a static square grid filled with "particles", which can only interact with the particles to the: left, right, up and down of it. The model is used to describe ferromagnetic properties and uses quantized spin. Following the famous mathematicians Arnold Ross's words "Think deeply about simple things", which is especially true when exploring the Ising model. It is quite a simple model yet it reveals profound physical properties.

While being a good model, it also has its downsides. For example, the total amount of states becomes quite large very fast, meaning that simulating it becomes computationally expensive even though every particle can only occupy one out of two spin-states, spin up or spin down. Since this method requires a lot of computation it limits us to only calculating numerical properties. The problem of computational expense becomes especially clear when we discover the difficulties of calculating the probability distribution function. The number of states grows exponentially, so to solve this problem we implement the Monte Carlo Metropolis algorithm. This algorithm lets us avoid the most computationally expensive calculations while not losing much accuracy and still being able to calculate physical observables.

The numerical values from our simulations can be verified by comparing them to analytical results calculated by Lars Onsager in 1944 [1]. We start by analytically exploring a $2 \times 2$ grid as a basis for our algorithm. Later we increase this to a $20 \times 20$ grid, where we investigate the acceptance of states for temperatures [1K, 2.4K]. While also estimating the probability distribution function for expected energy per spin for the same temperatures. We continue by studying the critical temperature by assuming a power law around it and looking at phase transitions. Estimating the critical temperature for an infinite grid using linear regression.

This paper gives only a brief insight into the perplexing world of magnetism and it shows how understanding their inner workings is important fundamental research. Today magnets are used in different fields such as medicine, industry and technology. A better understanding of magnets allows for better research in technology and gives us fundamental insight into condensed matter. It is therefore important to understand models for magnetism, and one step along the way is the Ising model.

## 3. Theory

### i. Units

This paper's equations are scaled by units of $J$ and $k_B$. Where $J$ is the coupling constant, and $k_B$ is the Boltzmann constant. This is primarily to avoid under or overflow problems when running simulations, and makes comparing results for different grid sizes easier. The units used are $[J]$ for energy, $[k_B]$ for specific heat capacity, $[J/k_B]$ for temperature, $[1/J]$ for magnetic susceptibility and magnetization is unitless. Our results are written in these units, and every unit is written per particle.

### ii. Statistical Mechanics and analytical expressions for a simplified system.

Working on highly complex systems with many degrees of freedom, one can't know every particle's exact position and velocity. The Ising model consists of a $L \times L$-grid. Each position in the grid is occupied by one particle, which has either positive $+1$ or negative $-1$ spin. Although this might seem like a simple system, the number of possible states grows exponentially when the grid size is increased making the system hard to work with. A way around this is working with the system statistically making it possible to get good estimates for large grids. Before starting the analysis, we note that the probability of observing a state $i$ is given by

$$p_i = \frac{1}{Z} e^{-\beta E_i}. \tag{1}$$

In equation (1), $\beta = \frac{1}{k_B T}$, $E_i$ is the energy in microstate $i$ and $k_B$ is the Boltzmann constant.

### i. *Partition function*

We start by looking at a canonical and discrete partition function $Z$ because it describes the statistical properties of a system in thermodynamical equilibrium.

$$Z = \sum_{\mathrm{i}} e^{-\beta E_i}. \tag{2}$$

Wanting the mean energy and magnetization values to be on a per-spin basis, we calculate it with the partition function as a dividend seeing that it's dimensionless. Finding the energy $E(\mathbf{s})$ for a microstate $\mathbf{s}$ can be done by taking the sum over all neighboring pairs, without double-counting, meaning we only evaluate the path between different connected particles once when taking the sum. Where each microstate $\mathbf{s}$ is a vector $\mathbf{s} = (s_1, s_2, \ldots, s_N)$, representing the spin state of the entire system. In a $L \times L$-system, $N$ is defined as $N \equiv L^2$.

$$E(\mathbf{s}) = -J \sum_{\langle kl \rangle}^{N} s_k s_l. \tag{3}$$

In equation (3), $J$ is the coupling constant, representing the strength of the bonds between neighboring particles. $\langle kl \rangle$ indicates that we sum over all the nearest neighbors, and $s_k$ and $s_l$ are the adjacent particle's spin. Inserting equation (3) into equation(2) gives us the partition function for an Ising model.

$$Z = \sum_{\text{all possible } \mathbf{s}} e^{-\beta E(\mathbf{s})}. \tag{4}$$

In equation (4) the total number of microstates, given that we have $N$ spins is $2^N$, as each spin state is either $+1$ or $-1$. Calculating $E(\mathbf{s})$ for our 2-dimensional system.

$$E(\mathbf{s}) = -J \sum_{i}^{N} \sum_{j}^{N} (s_{i,j} s_{i,j+1} + s_{i,j} s_{i+1,j}). \tag{5}$$

Computing the analytical values for our 2-dimensional 2-spin system with periodic boundary conditions we get the values shown in table (I). Having a periodic boundary condition, means that we compute the values, pretending the grid is repeated infinitely many times in the directions: up, down, left and right.

Calculating the magnetization of the system $M(\mathbf{s})$ by summing the spins.

$$M(\mathbf{s}) = \sum_{i=1}^{N} s_i. \tag{6}$$

To compute the partition function from equation (4) we insert all the 16 possible $E(\mathbf{s})$ for the $2 \times 2$-system. Multiplying the number of degeneracies with their respective $E(\mathbf{s})$. Thus giving us the expression for Z with a $2 \times 2$ grid.

$$\begin{aligned} Z_{2\times 2} &= e^{-\beta(-8)J} + 2 \cdot e^{-\beta 8J} + e^{-\beta(-8)J} + 12 \\ &= 2 \cdot e^{-8\beta J} + 2 \cdot e^{8\beta J} + 12 \\ &= 4\cosh(8\beta J) + 12. \end{aligned} \tag{7}$$

| Spin-up states | Degeneracy | Total magnetization | Total energy |
|---|---|---|---|
| 4 | 1 | 4 | -8 |
| 3 | 4 | 2 | 0 |
| 2 | 4 | 0 | 0 |
| 2 | 2 | 0 | 8 |
| 1 | 4 | -2 | 0 |
| 0 | 1 | -4 | -8 |

Table I. A table of all the different states with spin-up configurations, total energy, total magnetization and degeneracies for a two-dimensional system.

### *ii. Expectation value of energy per spin*

Finding the expectation value of energy per spin $\epsilon(\mathbf{s})$. We take the sum of all energy states, multiplied by their probability and get $\langle\epsilon\rangle$. Where Z is the partition function, defined in equation (4), and $E(\mathbf{s})$ in equation (3).

$$\epsilon(\mathbf{s}) = \frac{E(\mathbf{s})}{N}. \tag{8}$$

$$\begin{aligned}\langle\epsilon\rangle = \left\langle\frac{E}{N}\right\rangle &= \frac{1}{N}\sum_{\text{all possible }\mathbf{s}} p(\mathbf{s})\,E(\mathbf{s}) \\ &= \frac{1}{NZ}\sum_{\text{all possible }\mathbf{s}} E(\mathbf{s})\,e^{-E(\mathbf{s})\beta}.\end{aligned} \tag{9}$$

Similarly $\langle\epsilon^2\rangle$, is almost the same as equation (9) but we square the values of $\epsilon(\mathbf{s})$.

$$\langle\epsilon^2\rangle = \left\langle\frac{E(\mathbf{s})^2}{N^2}\right\rangle = \frac{1}{N^2Z}\sum_{\text{all possible }\mathbf{s}} E(\mathbf{s})^2\,e^{-E(\mathbf{s})\beta}. \tag{10}$$

Using equations (7, 9, 10) and the values in table (I). We can compute $\langle\epsilon\rangle$ and $\langle\epsilon^2\rangle$ for a $2\times 2$-system.

$$\langle\epsilon\rangle_{2\times2} = \frac{1}{4(4\cosh(\beta 8J)+12)}(16e^{-\beta 8J} - 16e^{\beta 8J}) = -\frac{2\sinh(8\beta J)}{\cosh(8\beta J)+3}. \tag{11}$$

$$\langle\epsilon^2\rangle_{2\times2} = \frac{1}{16(4\cosh(8\beta J)+12)}(2\cdot 8^2 e^{-\beta 8J} + 2\cdot 8^2 e^{\beta 8J}) = \frac{4\cosh(8\beta J)}{\cosh(8\beta J)+3}. \tag{12}$$

### *iii. Expectation value of the absolute value of magnetization per spin*

In studying a ferromagnet, one of the most central properties is its magnetization. Writing this as an expression for the total amount of magnetization and dividing it by the total amount of spin. We get the expression for magnetization in the system on a per-spin basis $m(\mathbf{s})$.

$$m(\mathbf{s}) = \frac{M(\mathbf{s})}{N}. \tag{13}$$

Calculating the expression for the expected absolute value of magnetization per spin $\langle|m|\rangle$ is done by taking the sum of the absolute value of the magnetization over all states multiplied by the probability of each state.

$$\langle|m|\rangle = \frac{1}{N}\sum_{\text{all possible } \mathbf{s}} |M(\mathbf{s})|p(\mathbf{s}) = \frac{1}{NZ}\sum_{\text{all possible } \mathbf{s}} |M(\mathbf{s})|e^{-E(\mathbf{s})\beta}. \tag{14}$$

Computing $\langle m^2\rangle$, we square the values of $|m(\mathbf{s})|$.

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

Finding the analytical expressions for $\langle|m|\rangle$ and $\langle m^2\rangle$ for a $2 \times 2$-system using (7, 14, 15) and the values in table (I) we get.

$$\begin{aligned}\langle|m|\rangle_{2\times 2} &= \frac{1}{4(4\cosh 8\beta J + 12)}(4e^{8\beta J} + 2\cdot 4 + 2\cdot 4 + 4e^{8\beta J}) \\ &= \frac{e^{8\beta J} + 2}{2\cosh(8\beta J) + 6}.\end{aligned} \tag{16}$$

$$\begin{aligned}\langle m^2\rangle_{2\times 2} &= \frac{1}{4^2(4\cosh 8\beta J + 12)}(4^2 e^{8\beta J} + 2^2\cdot 4 + (-2)^2\cdot 4 + (-4)^2 e^{8\beta J}) \\ &= \frac{e^{8\beta J} + 1}{2\cosh(8\beta J) + 6}.\end{aligned} \tag{17}$$

*iv. Specific heat capacity (normalized to the number of spins)*

The definition for the specific heat capacity is the amount of energy needed for the given material to increase its temperature by one degree. It is normalized to the number of spins to make it easier to work with.

$$C_V = \frac{1}{N}\frac{1}{k_B T^2}(\langle E^2\rangle - \langle E\rangle^2). \tag{18}$$

Where $T$ is the temperature of the system and $E$ is the energy in the system. Computing $\langle E\rangle^2$ and $\langle E^2\rangle$, we remind ourselves that $E(\mathbf{s}) = \epsilon(\mathbf{s})N$ and use equations (9, 10).

$$\langle E\rangle = N\langle\epsilon\rangle = \frac{1}{Z}\sum_{\text{all possible } \mathbf{s}} E(\mathbf{s})\, e^{-E(\mathbf{s})\beta}. \tag{19}$$

$$\langle E^2 \rangle = N^2 \langle \epsilon^2 \rangle = \frac{1}{Z} \sum_{\text{all possible } \mathbf{s}} E(\mathbf{s})^2 \, e^{-E(\mathbf{s})\beta}. \tag{20}$$

Calculating the analytical expression for $C_V$ for a $2 \times 2$-system using equations (19, 20) and the results in equations (11, 12) we get.

$$\begin{aligned} C_{V(2\times2)} &= \frac{1}{4k_B T^2} \left( \frac{64 \cosh(8\beta J)}{\cosh(8\beta J) + 3} - \frac{64 \sinh^2(8\beta J)}{(\cosh(8\beta J) + 3)^2} \right) \\ &= \frac{16}{k_B T^2} \frac{\cosh(8\beta J)(\cosh(8\beta J + 3)) - \sinh^2(8\beta J)}{(\cosh(8\beta J) + 3)^2} \\ &= \frac{16}{k_B T^2} \frac{1 + 3\cosh(8\beta J)}{(\cosh(8\beta J) + 3)^2}. \end{aligned} \tag{21}$$

Where we use the trigonometric identity $\cosh^2 x - \sinh^2 x = 1$.

### v. *Susceptibility (normalized to the number of spins)*

The susceptibility of a system normalized to the number of spins is given by $\chi$, where

$$\chi = \frac{1}{N} \frac{1}{k_B T} (\langle M^2 \rangle - \langle |M| \rangle^2). \tag{22}$$

$M$ is defined by equation (6). To compute $\langle M \rangle^2$ and $\langle M^2 \rangle$, we note that $M(\mathbf{s}) = m(\mathbf{s})N$ and use equations (14, 15).

$$\langle |M| \rangle = N \langle |m| \rangle = \frac{1}{Z} \sum_{\text{all possible } \mathbf{s}} |M(\mathbf{s})| e^{-E(\mathbf{s})\beta}. \tag{23}$$

$$\langle M^2 \rangle = N^2 \langle m^2 \rangle = \frac{1}{Z} \sum_{\text{all possible } \mathbf{s}} M(\mathbf{s})^2 e^{-E(\mathbf{s})\beta}. \tag{24}$$

Using equations (23, 24) and the results in equations (16, 17). We calculate $\chi$ for our $2 \times 2$-system.

$$\begin{aligned} \chi_{(2\times2)} &= \frac{1}{4}(16\langle m^2 \rangle_{2\times2} - 16\langle |m| \rangle^2_{2\times2}) \\ &= \frac{4e^{8\beta J} + 4}{2\cosh(8\beta J) + 6} - \frac{4(e^{8\beta J} + 2)^2}{(2\cosh(8\beta J) + 6)^2} \\ &= \frac{(4e^{8\beta J} + 4)(2\cosh(8\beta J) + 6) - 4(e^{8\beta J} + 2)^2}{(2\cosh(8\beta J) + 6)^2}. \end{aligned} \tag{25}$$

### vi. *5 different energy transitions*

In the Metropolis algorithm, which is described in more detail in section (4 ii). A key step is flipping one spin and looking at the ratio between the probabilities of each state. Where the

probability is defined in equation (1).

$$\frac{\text{Probability of current state i}}{\text{Probability of state j, after flip}} = e^{-(E_j - E_i)\beta}. \tag{26}$$

As we are flipping only one spin, we can deduce that it's only the surrounding particle's spin that affects the difference in energy, from the formula for energy given in equation(5). By symmetry of equation (5), it doesn't matter how the spins around are oriented, only the number of $+1$ and $-1$ states matter. Using this we get the different values of $E_j - E_i$ described in table (3 ii vi).

| +1 spins around / Spin state after flipping | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| -1 | 8 | 4 | 0 | -4 | -8 |
| 1 | -8 | -4 | 0 | 4 | 8 |

Table II. Table showing the energy difference between two states after flipping one spin. Where we subtract the energy before from the energy after.

From table (3 ii vi), we see that the energy difference after only flipping one spin can be five different values: $-8, -4, 0, 4, 8$.

## 4. Method

The system representing the Ising model is a $L \times L$-grid with $N = L^2$ number of spins. As mentioned earlier in the paper the total number of configurations for this system is $2^N$, which for big $N$ becomes too computationally expensive to compute analytically. Especially the partition function $Z$ defined in equation (7), which requires evaluating all possible states. To circumvent this we are going to make use of a Markov Chain Monte Carlo approach to simulate the Ising model.

### i. Markov chain and Monte Carlo methods

A Markov chain is a stochastic model, where the next state in a sequence only depends on the current state. This can be thought of as someone taking a random walk between different states of a system, and the probability of the next state they visit only being dependent on where they currently are. A consequence of this is that over time, given that it satisfies some conditions which will be discussed later, the Markov chain will reach an equilibrium distribution. Meaning we get a stable distribution of the different states of the system. Combining this with Monte Carlo methods in what's known as the Metropolis algorithm, we will use this to estimate the different distributions of energy and magnetism for grids of different sizes. Monte Carlo methods are a large class of methods, where we utilize pseudo-random sampling on a computer to compute the desired value. A famous example using this is Monte Carlo integration. Where its possible to estimate an integral of a function, by sampling random values from the function.

## ii. The Metropolis algorithm

In order for a Markov chain to be able to reach equilibrium it needs to satisfy two conditions, ergodicity and detailed balance. For a chain to be ergodic, we need that all microstates can be reached from any other microstate in a finite number of steps. Where a step is going one state forward in the chain. By definition, our system satisfies this, as it's always possible to flip all spins and it's finite. Detailed balance requires that in an equilibrium, two states $i$ and $j$ satisfy equation (27) for all $i$ and $j$.

$$\pi_i p_{ij} = \pi_j p_{ji} \qquad \forall i, j. \tag{27}$$

In equation (27), $\pi_i$ is the probability distribution of state $i$ in equilibrium and $p_{ij}$ the transition probability from state $i$ to $j$. The distribution of states follows a Boltzmann distribution, meaning $\pi_i$ is determined from equation (1). The transition probability $p_{ij}$, is the total probability of going from state $i$ to $j$.

$$p_{ij} = T_{ij} A_{ij}. \tag{28}$$

In equation (28), $T_{ij}$ is the probability of trying the transition between states $i$ and $j$, and $A_{ij}$ is the probability of this transition being accepted. If we insert equation (1) into equation (28), we get.

$$T_{ij} A_{ij} = \frac{1}{Z} e^{-\beta E_i}. \tag{29}$$

As noted earlier, equation (29) is computationally expensive to calculate as $Z$ requires us to evaluate all possible states. To circumvent this, we will look at the ratio between two transition probabilities.

$$\frac{p_{ij}}{p_{ji}} = \frac{T_{ij} A_{ij}}{T_{ji} A ji} = e^{-(E_j - E_i)\beta} = e^{-\Delta E \beta}. \tag{30}$$

Where $\Delta E = E_j - E_i$. We note that in equation (30), $Z$ is canceled out which is very beneficial for the computational expense. We now assume that $T_{ij} = T_{ji}$ meaning that the ratio between acceptance probabilities is given as.

$$\frac{A_{ij}}{A ji} = e^{-\Delta E \beta}. \tag{31}$$

To make our Monte Carlo Chain reach equilibrium fast, we want to accept as many moves as possible. We therefore set $A_{ji} = 1$, meaning that equation (31) simplifies to.

$$A_{ij} = e^{-\Delta E \beta}. \tag{32}$$

Noting that $A_{ij} = \min(e^{-\Delta E \beta}, 1)$, as it is a probability. We are now ready to describe the Metropolis algorithm, which we will implement to simulate our the system through the Ising model. First, we generate a $L \times L$-grid with $N = L^2$ spins being either $+1$ or $-1$ generated from a uniform distribution. Next, we do the following procedure, which is one Monte Carlo cycle, as many times as we deem necessary to ensure that we are close to equilibrium:

1. Repeat the following N times:
   (a) Generate two random integers x and y, both generated from a uniform distribution between 0 and $L-1$.
   (b) Find the difference in energy $e^{-\Delta E\beta}$, if we flip the particle with x-coordinate x and y-coordinate y.
   (c) Set the acceptance probability, $A_{ij} = \min(e^{-\Delta E\beta}, 1)$.
   (d) Generate a random number r, from a uniform distribution between 0 and 1.
   (e) If $r <= A_{ij}$ we flip the particle, if not we do nothing.
2. Save the values from the grid we are interested in.

Using this algorithm it's possible to estimate the expected values of for example energy and magnetization. Which we will discuss in more detail later in section (4 iv). As the Metropolis algorithm satisfies the ergodicity and detailed balance conditions. The Monte Carlo chain describing the system will if we run enough cycles, approach equilibrium, and get estimates close to the analytical solution.

## iii. Random Number Generators

In the Metropolis algorithm, we generate $3N$ random numbers for each cycle and $N$ random numbers when we initialize the grid. It is therefore important that the Random Number Generator (RNG) we use, samples independent numbers. To ensure this, we use the Merseene twister RNG which has a period of $2^{19937}$. This ensures that even though we might generate $10^9$ random numbers, which is around the period of some RNGs. We minimize the number of correlated numbers in the simulation.

## iv. Desired values and estimators

In this paper, the values from the Metropolis algorithm we are interested in are energy and magnetization of the system. More specifically the expected value of the energy per spin, $\langle\epsilon\rangle$, and the expected value of the absolute value of the magnetization per spin $\langle|m|\rangle$. To estimate these quantities, we take an average over all saved values of energy and magnetization we get after each cycle. More precisely we are going to use $\tilde{E}_\epsilon$ to estimate $\langle\epsilon\rangle$ and $\tilde{E}_{|m|}$ to estimate $\langle|m|\rangle$.

$$\tilde{E}_\epsilon = \frac{1}{K}\sum_{i=1}^{K}\epsilon_i. \qquad \tilde{E}_{|m|} = \frac{1}{K}\sum_{i=1}^{K}|m_i|. \tag{33}$$

In equation (33), $K$ is the total number of values saved and $\epsilon_i$ and $m_i$ are the values at time step $i$. We can think of $\epsilon_i$ and $m_i$ as stochastic variables, we observe when we simulate.

$$\langle\tilde{E}_\epsilon\rangle = \langle\frac{1}{K}\sum_{i=1}^{K}\epsilon_i\rangle = \frac{1}{K}\sum_{i=1}^{K}\langle\epsilon_i\rangle = \langle\epsilon\rangle. \tag{34}$$

$$\langle \tilde{E}_{|m|} \rangle = \langle \frac{1}{K} \sum_{i=1}^{K} |m_i| \rangle = \frac{1}{K} \sum_{i=1}^{K} \langle |m_i| \rangle = \langle |m| \rangle. \tag{35}$$

The reason for choosing these estimators, is that given our system is in equilibrium they are unbiased estimators as we can see in equations (34, 35). The reason the system needs to be in equilibrium is that it ensures that $\langle \epsilon_i \rangle = \langle \epsilon \rangle$ and $\langle |m_i| \rangle = \langle |m| \rangle$

## v. Optimizing running time of the Metropolis algorithm

To ensure we get close to equilibrium in the Monte Carlo Chain, we are going to simulate our system for up to $10^7$ cycles. As the size of the grid increases, so does the running time. To ensure feasible running times we have parallelized the code in "main.cpp". To do this we have used the "OpenMP" library and parallelized the simulation. This makes it possible to run multiple Markov Chains at once, meaning that instead of one chain running for $10^7$ cycles, we have 10 chains run for $10^6$ cycles, and combine the results. Greatly decreasing the running time for large grids. To ensure that each Markov Chain is not correlated, we add $10^6$ extra to the base seed for the RNG, for each chain except the first one. We also have to make sure that each chain reaches equilibrium, by choosing the number of cycles to be large enough.

As shown in table (3 ii vi), the energy difference after flipping one spin-state can only take on five different values. Using this, it's possible to precompute the different values we can get when computing $e^{-\Delta E \beta}$. Thus speeding up our algorithm even further.

## vi. Generating the data

We are now ready to use the Metropolis algorithm, to study the Ising model. The values we are interested in saving after each cycle, are the energy and magnetization of the system. To do this we run the following simulations:

- $2 \times 2$- grid, $T = 1.0$, $10^7$ cycles, random initial position and base seed 1000000.
- $20 \times 20$- grid, $T = 1.0$, $10^7$ cycles, random initial position and base seed 1000000.
- $20 \times 20$- grid, $T = 1.0$, $10^7$ cycles, ordered initial position and base seed 1000000.
- $20 \times 20$- grid, $T = 2.4$, $10^7$ cycles, random initial position and base seed 1000000.
- $21 \times 21$- grid, $T = 1.0$, $10^6$ cycles, random initial position and base seed 1000000.
- $41 \times 41$- grid, $T = 1.0$, $10^6$ cycles, random initial position and base seed 1000000.
- $61 \times 61$- grid, $T = 2.4$, $10^6$ cycles, random initial position and base seed 1000000.
- $61 \times 61$- grid, $T = 2.4$, $10^5$ cycles, random initial position and base seed 1000000.
- $121 \times 121$- grid, $T = 2.4$, $10^4$ cycles, random initial position and base seed 1000000.
- $121 \times 121$- grid, $T = 2.4$, $10^5$ cycles, random initial position and base seed 1000000.

- $40 \times 40$- grid, $T$ starting at 2.1 and ending at 2.4 with a increment of 0.006 for each simulation, $2 \cdot 10^6$ cycles, random initial position and base seed 1000000.

- $60 \times 60$- grid, $T$ starting at 2.1 and ending at 2.4 with a increment of 0.006 for each simulation, $2 \cdot 10^6$ cycles, random initial position and base seed 1000000.
- $80 \times 80$- grid, $T$ starting at 2.1 and ending at 2.4 with a increment of 0.006 for each simulation, $2 \cdot 10^6$ cycles, random initial position and base seed 1000000.
- $100 \times 100$- grid, $T$ starting at 2.1 and ending at 2.4 with a increment of 0.006 for each simulation, $2 \cdot 10^6$ cycles, random initial position and base seed 1000000.

### vii. Estimating critical temperature for an infinite grid

The critical temperature for a system is the temperature where the specific heat capacity or susceptibility goes from increasing to decreasing. Meaning that to find the critical temperature for a system, we are going to compute the specific heat capacity for the system. Then set the critical temperature to be the temperature, where the specific heat capacity is largest.

Using the critical temperature for different systems with different $L \times L$-grids. We estimate the critical temperature as a function of the inverse of the side length $L$, using linear regression. By doing this, it is possible to find an approximation of the critical temperature of an infinite grid. As we can look at the value of the regression slope when it crosses the y-axis. This is because we are looking at the inverse of $L$, so when $L$ goes to infinity the inverse goes to zero or more precisely, $\lim_{L\to\infty} \frac{1}{L} = 0$.

## 5. Discussion and Results

### i. Comparing analytical and numerical values for a $2 \times 2$-grid

We start by looking at a $2 \times 2$-grid, and checking that the generated data agrees with the analytical expressions. Which are expected energy per spin, expected absolute value of magnetization per spin, specific heat capacity, and susceptibility.

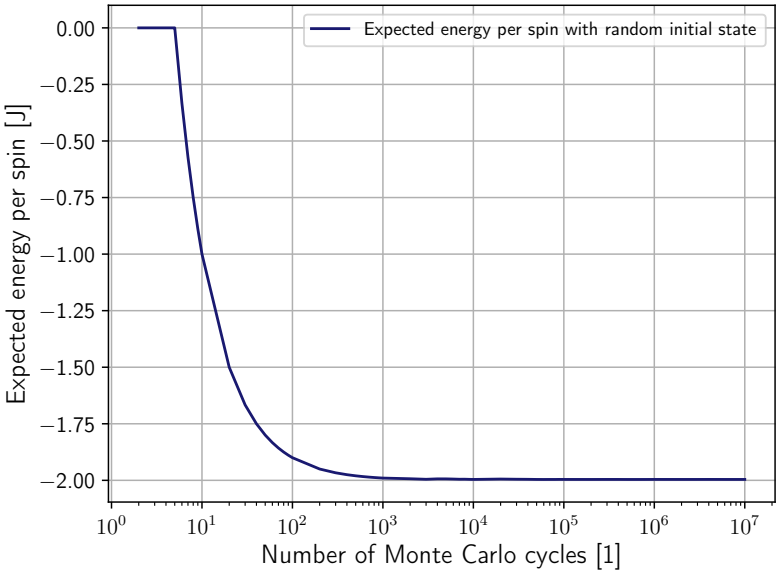


Figure 1. Plot of the approximation for expected energy per spin in a $2 \times 2$-grid with a random initial state, and $T = 1$ as a function of the number of cycles.

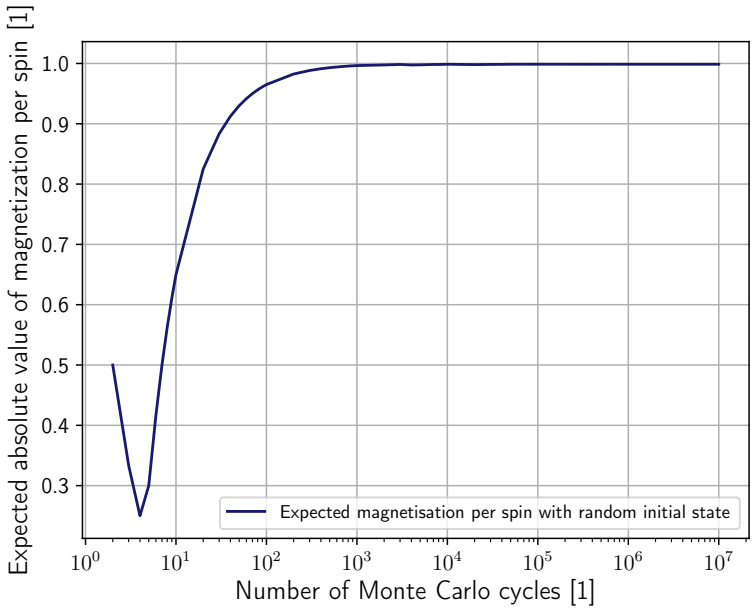


Figure 2. Plot of the approximation for the expected absolute value of magnetization per spin in a $2 \times 2$-grid with a random initial state, and $T = 1$ as a function of the number of cycles.

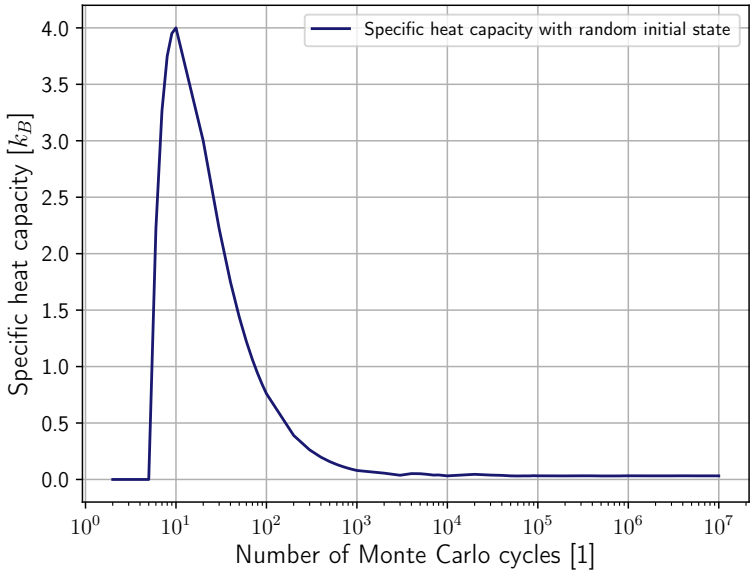


Figure 3. Plot of the approximation for specific heat capacity in a $2 \times 2$-grid with a random initial state, and $T = 1$ as a function of the number of cycles.

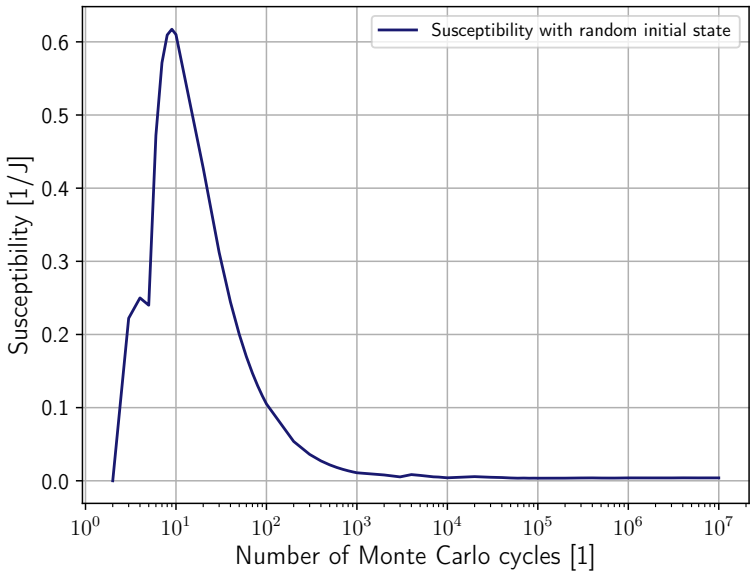


Figure 4. Plot of the approximation for susceptibility in a $2 \times 2$-grid with a random initial state, and $T = 1$ as a function of the number of cycles.

Using equations (11, 16, 21, 25) and reminding ourselves that $\beta = 1/(k_B T)$. We get the analytical values for a $2 \times 2$-grid with $T = 1$.

$$\langle \epsilon \rangle = -1.996[J] \quad \langle |m| \rangle = 0.999[1] \quad C_V = 0.032[k_B] \quad \chi = 0.004[1/J]. \tag{36}$$

Using the estimators in equation (33) and equations (18, 22), to compute the numerical values.

$$\langle \epsilon \rangle_{\text{num}} = -1.996[J] \quad \langle |m| \rangle_{\text{num}} = 0.999[1] \quad C_{V\text{num}} = 0.032[k_B] \quad \chi_{\text{num}} = 0.004[1/J]. \tag{37}$$

Comparing the values of equation (36) and equation (37), by observation it seems our simulations agree with the analytical values. This gives us an indication that our Metropolis algorithm is implemented correctly, and that the values in the rest of the report are correct. By observing figures (1, 2, 3, 4), it seems like we get the correct values after approximately $10^5$ cycles.

## ii. Analyzing the burn-in time for a $20 \times 20$-grid with random and ordered initial state

Analyzing the data from the simulation for a $20 \times 20$-grid. We are interested in studying the difference in burn-in time, meaning the number of cycles the Markov chain needs before it reaches equilibrium. We are especially interested in the difference when the initial state of the particles is random, and when they are all the same.

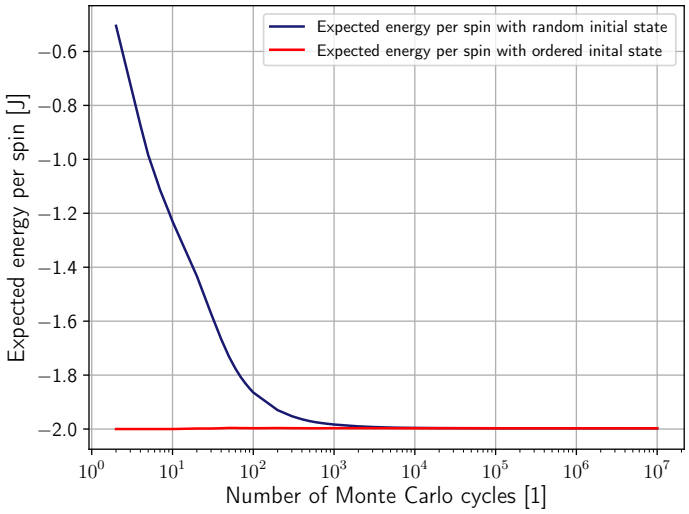


Figure 5. Plot of the approximation for expected energy per spin in a $20 \times 20$-grid, with $T = 1$ as a function of the number of cycles. The blue line represents the system where the initial state is random, and the red line represents the ordered system. Meaning all the spin-states start in the same state, which for this plot is $+1$.

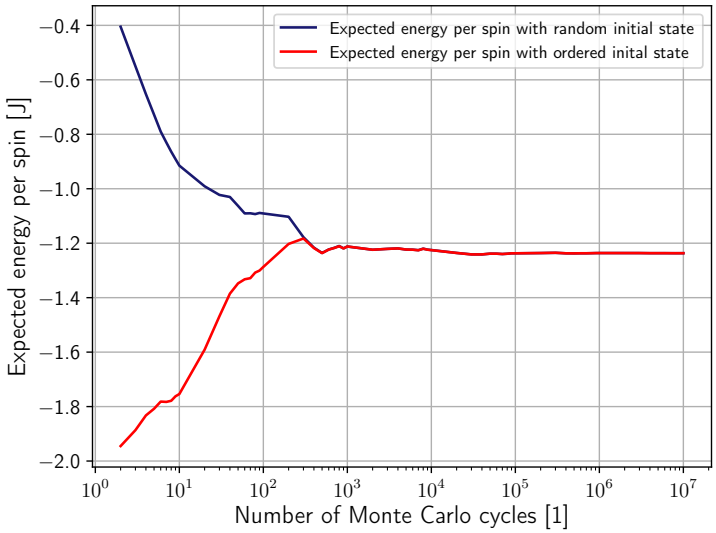


Figure 6. Plot of the approximation for expected energy per spin in a $20 \times 20$-grid, with $T = 2.4$ as a function of the number of cycles. The blue line represents the system where the initial state is random, and the red line represents the ordered system. Meaning all the spin-states start in the same state, which for this plot is $+1$.

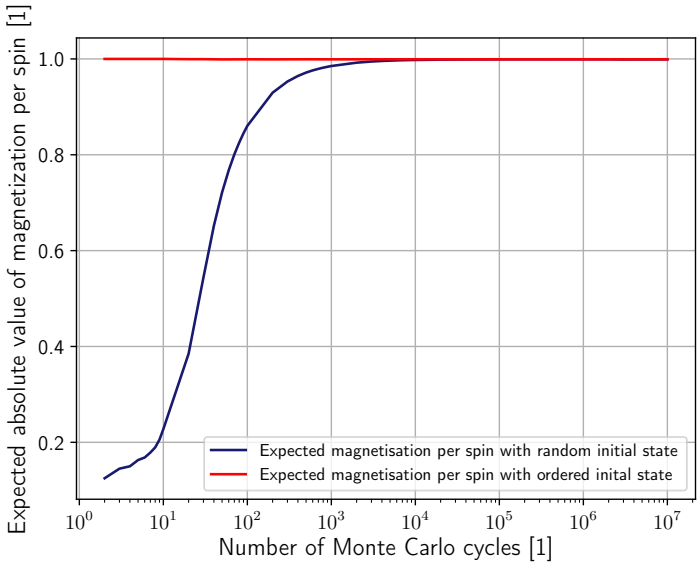


Figure 7. Plot of the approximation for the expected absolute value of magnetization per spin in a $20 \times 20$-grid, with $T = 1$ as a function of the number of cycles. The blue line represents the system where the initial state is random, and the red line represents the ordered system. Meaning all the spin-states start in the same state, which for this plot is $+1$.

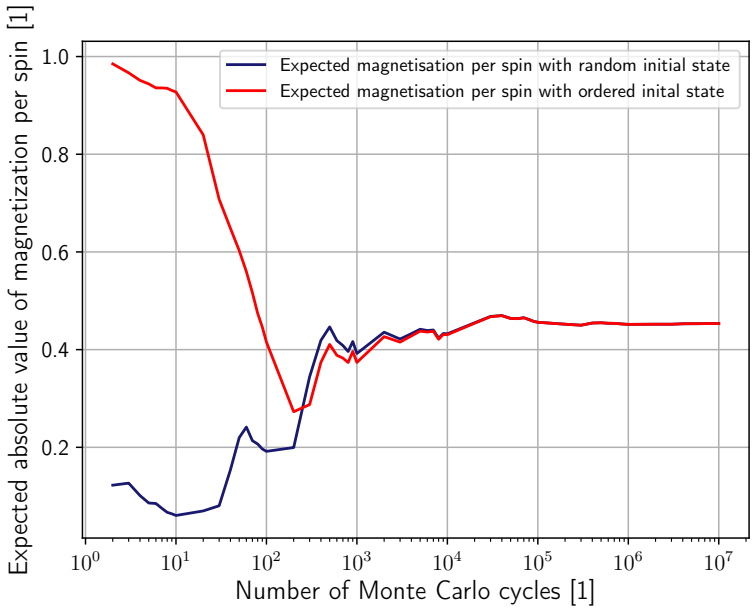


Figure 8. Plot of the approximation for the expected absolute value of magnetization per spin in a $20 \times 20$-grid, with $T = 2.4$ as a function of the number of cycles. The blue line represents the system where the initial state is random, and the red line represents the ordered system. Meaning all the spin-states start in the same state, which for this plot is $+1$.

By observation of figures (5, 7) we conclude that for a system with an ordered initial state. The burn-in-time when $T = 1$, is 0 as the system starts in equilibrium. This is because at low temperatures the magnetization is naturally aligned, meaning when you look at an ordered system it won't change because it is already close to equilibrium. The reason the unordered state converges towards a state with the same absolute expectation value at such a rate, is that a lot more states get accepted by the Metropolis condition. The closer the system is to equilibrium the fewer states meet the conditions.

When $T = 2.4$ by observation of figures (6, 8) we see that the burn-in time for both initial states are equal, meaning both are just as good approximations of the equilibrium. This is because the system needs more cycles to reach equilibrium, and therefore has to evolve before reaching it. It needs more time than when $T = 1$, because the higher energy causes the system to jump more between states. This oscillation continues even after the system reaches equilibrium because it wants to be as close as possible to its true value. We also observe that both initial states converge towards the same expected absolute magnetization per spin and that they both reach equilibrium after around $10^5$ cycles.

This is very much in contrast to when $T = 1$, where the ordered initial state converges immediately and the random initial state converges after $10^4$ cycles. From this, we conclude that the system modelled with ordered initial states gives a better representation of the true values for lower temperatures. This is because when $T = 2.4$ the system has a longer burn-in-time from the higher temperature. Meaning the equilibrium state has a larger variance, causing it to need a higher

number of MC cycles for a good approximation. It is useful to know these burn-in times, as we are only really interested in the values of the system when it is in equilibrium. Meaning that if we skip the values from the burn-in time, it's possible to get better approximations faster at the expense of computing redundant states which gives us no values.

### iii. Estimating the probability distribution for a $20 \times 20$-grid for $T = 1$ and $T = 2.4$

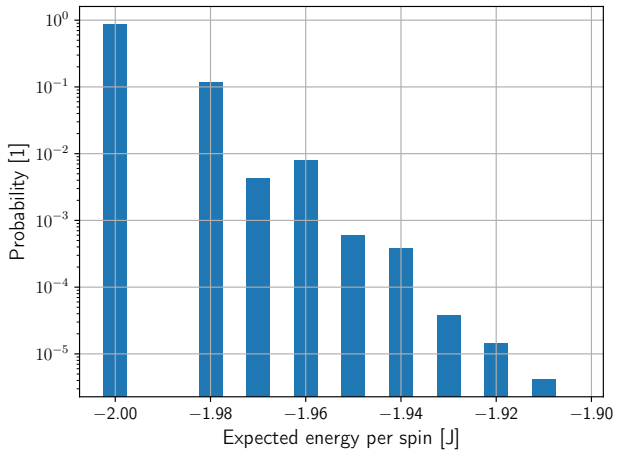


Figure 9. Plot of the discrete probability distribution for the energy per spin, in a $20 \times 20$-grid with a random initial state after $10^7$ cycles and $T = 1$. The variance $\sigma^2$ is, $\sigma^2 = 7.9 \cdot 10^{-5}$.

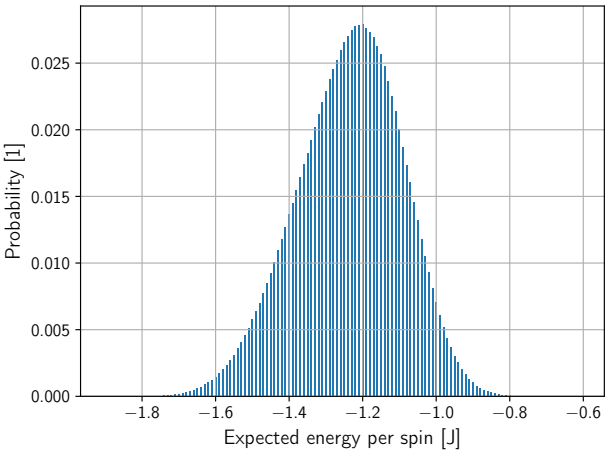


Figure 10. Plot of the discrete probability distribution for the energy per spin, in a $20 \times 20$-grid with an ordered initial state after $10^7$ cycles and $T = 2.4$. The variance $\sigma^2$ is, $\sigma^2 = 0.02$.

Observing figures (9, 10), we see clear differences in the distributions of expected energy per spin when $T = 1$ and $T = 2.4$. When $T = 1$, the probability of getting $\epsilon = -2$ is almost 1 Computing the variance we get that $\sigma^2_{T=1} = 7.9 \cdot 10^{-5}$, with a distribution which looks like an exponential distribution. This is very much in contrast to when $T = 2.4$, where the variance $\sigma^2_{T=2.4} = 2.0 \cdot 10^{-2}$, and we get a distribution that looks like a Gaussian distribution.

The reason for this large difference in the distribution comes from the energy difference between when $T = 1$ and $T = 2.4$. When $T = 2.4$, the energy in the system is much larger compared to $T = 1$, causing it to switch states more often. On the other side when $T = 1$, the system stays much more constant, and when it reaches a state where all spins are the same it stays there. Causing the distribution to be focused around the state where all spins are the same. The difference in $\sigma^2_{T=1}$ and $\sigma^2_{T=2.4}$ supports this, as more oscillations between states mean higher variance as the difference is larger. Meaning that $\sigma^2_{T=2.4}$ should be larger than $\sigma^2_{T=1}$, which it is by a factor of $2.5 \cdot 10^2$.

## iv. Comparing run times between parallelized and non-parallelized simulations

To compare the run time between parallelized and non-parallelized simulations. We will compile the program with parallelization using 10 cores and run some simulations, afterwards we compile without parallelization and run the same tests. The results from this can be found in table (5 iv). All simulations used base seed $10^6$ and a random initial state.

| L | Cycles | T | Run time with parallelization [s] | Run time without parallelization [s] | Speedup |
|---|---|---|---|---|---|
| 21 | $10^6$ | 1 | 4.615 | 28.262 | 6.12 |
| 41 | $10^6$ | 1 | 16.257 | 106.237 | 6.53 |
| 61 | $10^5$ | 2.4 | 3.854 | 25.483 | 6.61 |
| 61 | $10^6$ | 2.4 | 38.217 | 256.849 | 6.72 |
| 121 | $10^4$ | 2.4 | 1.588 | 10.467 | 6.59 |
| 121 | $10^5$ | 2.4 | 15.946 | 108.623 | 6.81 |

Table III. Table showing the different run times of simulations with different grid sizes, temperatures and cycles. The speedup is computed by dividing run time with parallelization by run time without parallelization.

By observing table (5 iv), it's clear to see that the speedup using parallelization is around 6.5. Meaning that it goes 6.5 faster to run the simulation using parallelization. We also see that the larger the simulations are in grid size and cycles, the bigger the speed up becomes.

v. Approximating the critical temperature for an infinite grid using linear regression

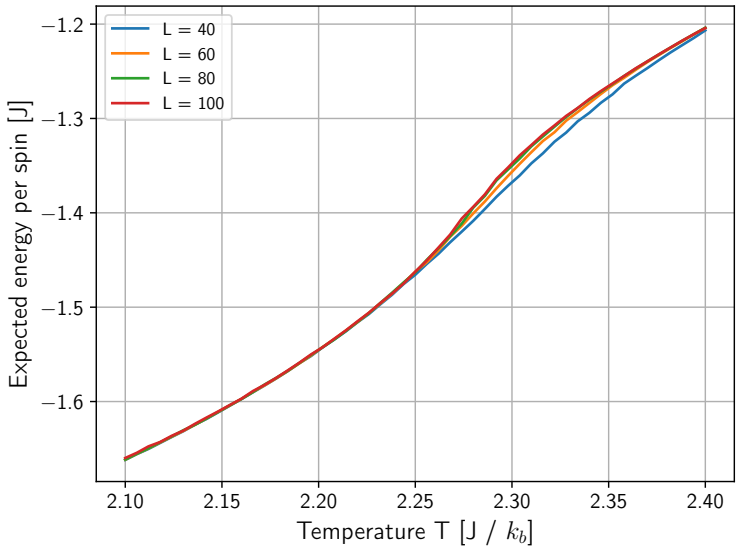


Figure 11. Plot of the expected energy per spin after $2 \cdot 10^6$ cycles as a function of temperature. Run for different side lengths L.

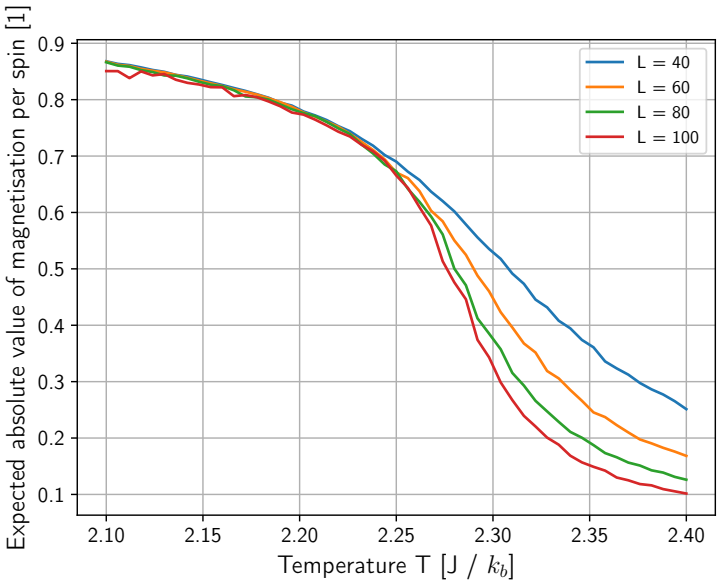


Figure 12. Plot of the expected absolute value of magnetisation per spin after $2 \cdot 10^6$ cycles as a function of temperature. Run for different side lengths L.

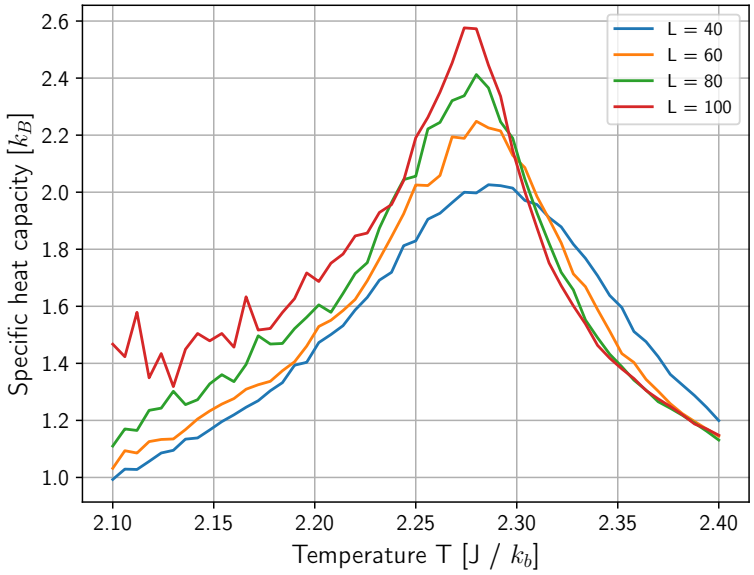


Figure 13. Plot of the specific heat capacity after $2 \cdot 10^6$ cycles as a function of temperature. Run for different side lengths L.

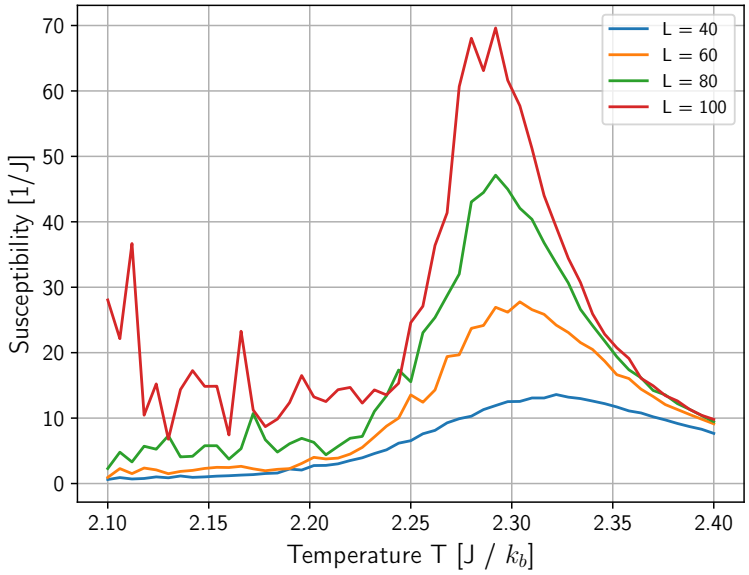


Figure 14. Plot of the susceptibility after $2 \cdot 10^6$ cycles as a function of temperature. Run for different side lengths L.

By observing figures (11, 12, 13, 14), we see clear signs of a phase transition. A phase transition is when a system reaches the critical temperature, and the energy in the system causes the crystal structure to break down. Reaching a phase transition causes a sudden drop in specific heat capacity and susceptibility, and the expected absolute value of magnetization goes toward zero. By observation, we conclude that the figures show this happening around $T = 2.28$ for all the different grid lengths. Using the method described in subsection (4 vii), we get the critical temperatures, $T_c$.

$$T_c(40) = 2.286 \quad T_c(60) = 2.280 \quad T_c(80) = 2.280 \quad T_c(100) = 2.274. \tag{38}$$

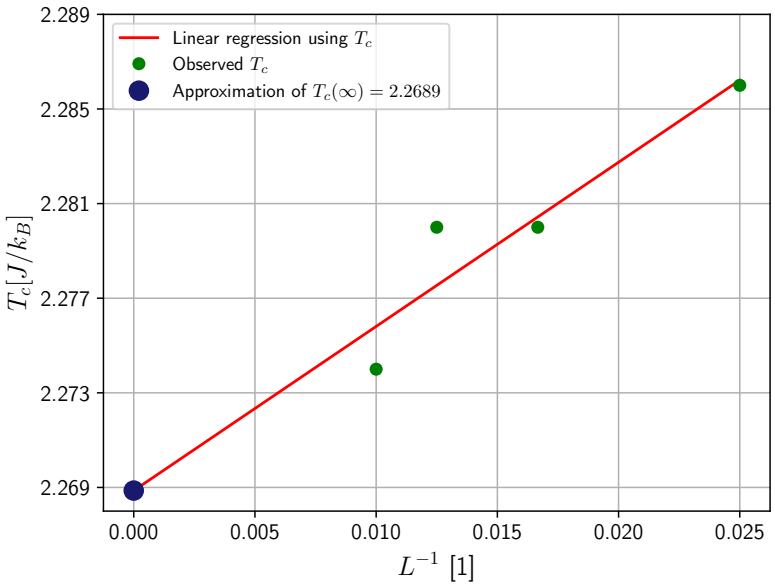


Figure 15. Plot of the estimated critical temperature for an infinite grid. Using linear regression on the critical temperature for different grid sizes L.

Using the data in equation (38), we can do the linear regression in figure (15). From this, we can get an approximation of the critical temperature for an infinite grid, using the method described in section(4 vii). Our approximation is $T_c(\infty) = 2.2689[J/K_B]$, when we compare this to Lars Onsager's [1] analytical results of $2.269[\mathrm{J}/K_B]$ we conclude that our approximation is well within the range of the uncertainty.

## 6. Conclusion

Studying the Metropolis Monte Carlo method in a 2-dimensional lattice with periodic boundary conditions. We start by doing an analytical analysis of a $2 \times 2$-grid giving us values to compare against our analytical results. We compare these analytical values with the numerical result for a $2 \times 2$-grid as a fail-safe for our algorithm. By comparison, we observe that our numerical results

agree with the analytical ones. But needing $10^5$ Monte Carlo cycles for the numerical results to match the analytical ones.

We continue by analyzing a grid with the size $20 \times 20$. For this grid, we analyze how the system converges to an equilibrium state for different temperatures and different initial conditions. Observing that in general, the system needs $10^5$ MC cycles to reach equilibrium. While also noticing that the systems with lower temperatures, reaches the equilibrium state much faster with an ordered initial state than with a random initial state. The ordered initial state starts in equilibrium, and the random initial state reaches equilibrium after $10^4$ cycles.

This was not the case for the system with higher temperatures, where the ordered initial state and the random initial state reached equilibrium at about the same time after $10^5$ cycles. While also observing higher fluctuations around the equilibrium state with higher temperature and smaller fluctuations around the equilibrium state for the system with lower temperature. For future research, it would be interesting to investigate the effects of skipping the values in the burn-in time has on the estimated values.

It becomes too computationally expensive for large systems to calculate the pdf, meaning we have to approximate it numerically instead. Finding the probability distribution function is done by producing a histogram of the saved energy states. Which we do for a $20 \times 20$-grid with temperatures $[1,\ 2.4]J/k_b$. To make running times feasible we parallelize the code across 10 cores making the code approximately 6.5 times faster, noticing that the bigger the simulation, the bigger increase in computational speedup.

Finally, when we study the phase transitions by linear regression, we find the critical temperature to be around $T_c(\infty) = 2.2689[J/k_B]$. Comparing this with analytical results calculated by Lars Onsager [1] at $2.269[\mathrm{J}/k_B]$ we are well within the uncertainty.

Improvements for future research would be to run for bigger grid sizes such as a $500 \times 500$-grid, and do a more fine-grained search of the temperatures to get a better estimate of $T_c(\infty)$. The use of a more efficient method, like the heatbath algorithm, would improve our estimate of the critical temperature. Increasing the number of Monte Carlo cycles per temperature step, and solving the problem of noise coming from the low-temperature system, would also yield a more accurate estimate of the critical temperature.

---

[1] Lars Onsager. Crystal statistics. i. a two-dimensional model with an order-disorder transition. https://journals.aps.org/pr/abstract/10.1103/PhysRev.65.117, 4 October 1943.