The Two-dimensional Ising Model for Ferromagnetic Phase Transitions

James Claxton, Amund Fredriksen

November 26, 2020

Abstract

The ferromagnetic-paramagnetic phase transition in 2 dimensions is modeled and simulated using the Ising model in conjunction with the Metropolis algorithm. The model is compared with the analytic solution in the simple case of a 2×2 spin matrix for different numbers of Monte Carlo cycles used in the Metropolis algorithm. For a 20×20 spin lattice the model shows satisfactory results running the Metropolis algorithm for a total of 100,000 Monte Carlo cycles with the first 20,000 cycles necessary to reach equilibrium.

By computing the mean energy and magnetisation of the system at various temperatures, the critical temperature T_C was found for different sized spin lattices. Using these results we calculated the critical temperature in the thermodynamic limit when the size of the spin lattice goes to infinity. The result from these calculations was $2.273\, J/k_B$, and compared to the exact value this has a relative error of 0.176%.

1 Introduction

The properties of different materials in many ways forms the basis of and shapes the development of technology. When humans discovered that wood could take fire and burn for extended periods of time, a reliable source of heat was made available, making survival easier. When it was discovered that it was possible to use fire to melt metals, a whole new domain of technology was made available. And for today's modern technology, discovering new materials and new properties of known materials continues to be an important contributor to new inventions or improvements of existing technology. For this reason, doing research on the various properties of materials is a very useful endeavour for moving technology forward. Newly found properties can set the stage for a variety of things, such as stronger and lighter vehicles, improved solar cells, improved battery technology, better transistors, and much more.

One property of many materials is that they can be in different phases. As the macroscopic variables of a system are changed, abrupt changes in the system properties occur. This is called a phase transition. The most known phase transitions are the freezing of liquid water to ice water, or the evaporation from liquid water to steam. Which phase the water is in depends upon which has the lowest free energy. The Helmholtz free energy $F = E - T \cdot S$ describes the competition between energy and entropy when temperature increases. At low temperatures energy dominates and so water is in the solid phase. At high temperature, entropy dominates, and so water enter the liquid phase. Specific heat is a quantity defined as the amount energy to raise a unit of material by 1 degree Celcius. What happens in certain phase transitions when temperature increases is that energy goes into changing the phase of the material and so a peak is observed in specific heat. Wherever this peak is visible a phase transition is likely to be found.

Another of the most known phase transitions is the ferromagnetic phase transition. These magnetic phases is what we have simulated and studied. At a specific temperature e.g 1043°C for iron, iron will change from being ferromagnetic to paramagnetic. In this case increasing temperature leads to re-ordering of the magnetic domains, or domain-wall boundary movement. In this report the ferromagnetic phase transition is modeled using the Ising model and simulated using the Metropolis algorithm. The Ising model in three dimensions is computationally intractable and hence no exact simulations have been done. In one dimension the Ising model shows no phase transition at all [1]. On the other hand, in two dimensions the Ising model works very well and exhibits phase transitions and also makes many physical properties computable. For this reason, the Ising model is very popular for modelling ferromagnetic phase transitions. In this report, using the Ising model, the critical temperature in the thermodynamic limit where the size of the spin matrix goes to infinite is estimated. This numerical value is compared to Lars Onsager's exact value using J, the coupling constant, set to 1.0.

2 Theory

The derivations of some of the expressions in this section has been omitted here for brevity. The derivations can be found in the appendix of this document.

2.1 Units

In our programs we have set the Boltzmann constant to $k_B = 1$. Since this differs from the SI units, the results presented from our program's calculations will have non-SI units. Comparing with the SI units of the Boltzmann constant we get

$$\begin{aligned} k_B &= k_{B,\mathrm{SI}} \\ \Leftrightarrow 1 &= 1.38064852 \cdot 10^{-23}\,\mathrm{J/K} \\ \Leftrightarrow 1\,\mathrm{K} &= 1.38064852 \cdot 10^{-23}\,\mathrm{J} \\ \Leftrightarrow 1\,\mathrm{J} &= 7.242973034 \cdot 10^{22}\,\mathrm{K}, \end{aligned}$$

so temperature, usually measured in kelvin (K), will have the unit joules (J). Also, we have put $J = 1$. J is a coupling constant between the nearest neighbour spins, and is dimensionless. Setting $J = 1$ means that we are calculating the energy of the system in units of J.

2.2 Second order phase transitions

The ferromagnetic to paramagnetic phase transition is within the category of second order phase transitions. At the critical temperature between phases, the free energy must be equal. In a first order phase transition the first derivative of the free energy may be discontinuous, appearing as a jump. An example of a first order phase transition is the solid to liquid transition given in the Introduction. In the case of the second order phase transition the first derivative of the free energy is continuous, but the second derivative (C_V) is discontinuous [2].

Landau developed the theory of phase transition with a so called "order parameter". Landau recognised that in any phase transition an order parameter can be defined as zero above the critical temperature and non-zero below the critical temperature. For the ferromagnetic – paramagnetic phase transition this order parameter is the magnetisation. Landau also suggested that expanding the free energy in a Taylor expansion in the order parameter would give understanding of the behaviour of this phase transition. In the region near the critical temperature many physical quantities can be described by a power law behaviour with critical exponents.

The mean magnetisation is given by

$$\mathbb{E}[M(T)] \propto (T - T_C)^{\beta}, \tag{1}$$

where $\beta = 1/8$ is a so-called critical exponent.

Similarily the heat capacity and susceptibility is given by

$$C_V(T) \propto |T - T_C|^{\alpha}, \tag{2}$$

$$\chi(T) \propto |T - T_C|^{\gamma}, \tag{3}$$

where $\alpha = 0$ and $\gamma = 7/4$.

Using these critical exponents its possible to estimate T_C in the thermodynamic limit $L = \infty$ as given by

$$T_C(L) - T_C(L = \infty) = aL^{-1/\nu}, \tag{4}$$

where ν is the critical exponent for the correlation length and has the exact value equal to 1, and a is a constant [3]. The exact result after Lars Onsager is $2/\ln(1 + \sqrt{2}) \approx 2.269$.

2.3 The Ising model

The Ising model is a mathematical model of ferromagnetism [4]. This model starts out with a lattice of atoms, e.g. a two-dimensional lattice. Each atom has an associated magnetic moment vector. In this model this magnetic moment vector is modelled as having only two possible directions, either up ('spin up') or down ('spin down'). Spin up is denoted as $\uparrow$ and spin down is denoted as $\downarrow$. The spins are aligned along some axis, e.g. the z-axis.

Energy: In the Ising model the energy can be expressed as

$$E = -J \sum_{<ij>}^{N} s_i s_j - \mu H \sum_{i=1}^{N} s_i, \tag{5}$$

where N is the total number of spins, $< ij >$ indicates the sum over only the nearest neighbours, J is called the exchange energy which has to do with the interaction between adjacent spins, μ is the magnitude of the magnetic moment for each atom and H is an external magnetic field. In this project we will assume $H = 0$ so the energy is

$$E = -J \sum_{<ij>}^{N} s_i s_j. \tag{6}$$

The number of nearest neighbours of each spin depends on the dimensionality of the lattice. If we are working in a one-dimensional lattice, there will be two nearest neighbours, one on each side. If we are working in a two-dimensional lattice, there will be four nearest neighbours, two along each axis. The nearest neighbour sum is to go through each spin in the lattice, and for each spin go through all of its nearest neighbours, multiply them and add together all of those products.

Periodic boundary condition: In this project we will assume periodic boundary conditions. This means that the spins at any end points defines its nearest neighbour as the spin at the start of that row or column in the lattice.

Degeneracy: The degeneracy of a given energy level is defined as the number of states that have that energy.

Magnetic moment: The magnetic moment M for a given state of spins is defined as the sum of all of the spins:

$$M = \sum_{i=1}^{N} M_i. \tag{7}$$

2.4 The probability density function

The probability density for each of the states or spin configurations s is given by the Boltzmann-distribution, which has the following mathematical form:

$$P_s(\beta) = \frac{e^{-\beta E_s}}{Z}, \tag{8}$$

where $\beta \equiv 1/(k_B T)$ and Z is the partition function for the canonical ensemble defined as

$$Z = \sum_{s}^{N} e^{-\beta E_s}, \tag{9}$$

where N is the total number of microstates. P_s is the probability of finding the system in the state s at a given temperature [3].

2.5 Expected values

For a stochastic variable X with a discrete probability distribution, the expected value of a general function $r(X)$ is given by

$$\mathbb{E}[r(X)] = \sum_{i=1}^{N} P(X = x_i) r(x_i) \tag{10}$$

where $x_1, x_2, \ldots, x_N$ are all the possible values of X and $P(X = x_i)$ are the corresponding probabilities for X to take these values.

The expectation value of the energy is, with a Boltzmann distribution,

$$\mathbb{E}[E] = \sum_{s}^{N} E_s P_s = \frac{1}{Z} \sum_{s}^{N} E_s e^{-\beta E_s}. \tag{11}$$

Analytically, this can be written in a more compact way using the fact that for $y = \exp(-\lambda x)$, $dy/dx = -\lambda y$. This gives

$$\mathbb{E}[E] = -\frac{1}{Z} \frac{\partial Z}{\partial \beta} = -\frac{\partial \ln Z}{\partial \beta}. \tag{12}$$

The expectation value of the magnetisation is

$$\mathbb{E}[M(\beta)] = \frac{1}{Z} \sum_{s}^{N} M_s e^{-\beta E_s}, \tag{13}$$

where M_s is the magnetisation of the state s, not to be confused with saturation magnetisation.

2.6 Sample mean

The sample mean, or average, of some quantity X is defined as

$$\langle X \rangle \equiv \frac{\sum_{i=1}^{N} X_i}{N}. \tag{14}$$

This serves as an approximation for the actual expected value $\mathbb{E}[X]$ for a finite set of measurement data $X_1, X_2, \ldots, X_N$. The larger N is, the better the approximation will be.

2.7 Variance

The variance of a random variable X is defined as

$$\mathrm{Var}(X) \equiv \sigma_X^2 \equiv \mathbb{E}\left[(X - \mathbb{E}[X])^2\right] = \mathbb{E}[X^2] - \mathbb{E}[X]^2. \tag{15}$$

Therefore, an empirical approximation of the variance is

$$\sigma_X^2 \approx \langle X^2 \rangle - \langle X \rangle^2 \tag{16}$$

since $\langle X^2 \rangle \approx \mathbb{E}[X^2]$ and $\langle X \rangle \approx \mathbb{E}[X]$ for large samples.

2.8 Magnetic susceptibility and heat capacity

The magnetic susceptibility according to [5] and [6] is

$$\chi = \beta\sigma_M^2 = \beta\left(\mathbb{E}[M^2] - \mathbb{E}[|M|]^2\right) \approx \beta\left(\langle M^2\rangle - \langle|M|\rangle^2\right), \tag{17}$$

where σ_M^2 is the variance in magnetisation. The specific heat C_V is given by

$$C_V = \left(\frac{\partial\mathbb{E}[E]}{\partial T}\right)_V = \cdots = \frac{1}{k_B T^2}\frac{\partial^2 \ln Z}{\partial\beta^2}. \tag{18}$$

This can also be found using the relation

$$C_V = \frac{1}{k_B T^2}\sigma_E^2 \approx \frac{1}{k_B T^2}\left(\langle E^2\rangle - \langle E\rangle^2\right) \tag{19}$$

where σ_E^2 is the variance in energy.

2.9 Jumps in energy and magnetisation for single spin flips in a 2D lattice

2.9.1 Energy jumps, ΔE

When one spin flips, from up to down or opposite, the energy of the system changes because of the interaction between the nearest neighbours, as seen in the expression for the energy in Eq. (6). As seen in Eq. (13.6) in [3, p. 438], the energy difference from a spin flip is given by

$$\Delta E = 2Js_l^1 \sum_{<k>}^{N} s_k \tag{20}$$

where s_l^1 is the initial value of the spin that is flipped and the sum over k is a sum over its nearest neighbour spins s_k. In a 2D lattice each spin has four nearest neighbours, so $N = 4$. The different possible values of the energy difference is, as seen in eq. (20), mainly determined by the different possible sums of the nearest neighbour spins. There are four spins, and each spin has either value $+1$ or -1. The sum depends only on the number of up spins and down spins, and not on the order. The following possibilities exist:

- 4 up spins and 0 down spins: $\sum_{<k>} s_k = 4 \implies \Delta E = 2J\cdot(\pm1)\cdot 4$
- 3 up spins and 1 down spin: $\sum_{<k>} s_k = 2 \implies \Delta E = 2J\cdot(\pm1)\cdot 2$
- 2 up spins and 2 down spins: $\sum_{<k>} s_k = 0 \implies \Delta E = 0$
- 1 up spin and 3 down spins: $\sum_{<k>} s_k = -2 \implies \Delta E = 2J\cdot(\pm1)\cdot(-2)$
- 0 up spins and 4 down spins: $\sum_{<k>} s_k = -4 \implies \Delta E = 2J\cdot(\pm1)\cdot(-4)$,

where (±1) represent the two possible initial values of the spin that is flipped. It is seen that, when taking all the signs into consideration, there are five, and only five, different possible energy jumps:

$$\Delta E \in \{-8J,\, -4J,\, 0, 4J, 8J\}. \tag{21}$$

This means that the weights, given by $e^{-\beta\Delta E}$, also have only five possible values:

$$e^{-\beta\Delta E} \in \{e^{8J\beta},\, e^{4J\beta},\, 1,\, e^{-4J\beta},\, e^{-8J\beta}\}. \tag{22}$$

This means that the weights can be pre-calculated, which can help to reduce computation time.

2.9.2 Magnetisation changes, ΔM

The magnetisation change for a single spin flip is simple since the magnetisation change simply is the sum of all of the spins. There are only two cases:

- When a spin down (-1) is changed to a spin up $(+1)$, the difference is $\Delta M = (+1) - (-1) = +2$.
- When a spin up is changed to a spin down, the difference is $\Delta M = (-1) - (+1) = -2$.

These two cases for ΔM are retrieved from this single expression:

$$\Delta M = -2 \cdot s_{\text{before}} = 2 \cdot s_{\text{after}}, \tag{23}$$

where s_{before} and s_{after} is the spin before and after the spin flip is performed, respectively (either one can be used).

2.10 The Metropolis algorithm

In order to achieve our goal in this project, which is to study phase transitions of a system of magnetic spins in its equilibrium state, we need to have a method for calculating the mean values of the important quantities. This is because the phase transitions are studied by looking at the mean energy, mean magnetisation, heat capacity and magnetic susceptibility as functions of temperature. To calculate the mean values, we simulate the spin system using the Metropolis algorithm, which will be explained in this sub-section. The heat capacity and magnetic susceptibility are calculated directly from mean values of energy and magnetisation respectively, so all that is needed is the mean values of the energy and the magnetisation.

In theory we could calculate the expected values directly using the Boltzmann probability function in eq. (8), but then we would have to calculate the partition function Z for the spin system. For small spin systems this is feasible, but for larger ones, like a 20×20 system, this is not feasible since the amount of possible states (spin configurations) grows exponentially with the number of spins. For a 20×20 system the number of possible states is $2^{20 \cdot 20} \approx 10^{120}$, so computing the sum for Z is not possible, at least with today's computational power.

In essence, the Metropolis algorithm is a specific *sampling rule* used in a *Markov chain Monte Carlo method*. These terms are explained below.

2.10.1 Markov chain Monte Carlo methods

Markov chain Monte Carlo (MCMC) methods are, as the name might suggest, Monte Carlo methods based on Markov chains. A Markov chain is a sequence of random events where the next outcome is only dependent on the current state of the system, and not on the previous history of the system. In our example, the Ising model, this means that the probabilities for the next spin configuration is only dependent on the current spin configuration. Or more specifically, the *energy* of the current spin configuration, since this is what affects the Boltzmann probability distribution for the different spin states.

Our Ising spin system is a Markov chain since the probabilities for the different possibilities for the next state is dependent on the current state. We are simulating the system, and in the process finding averages of energy and magnetisation, by using a Monte Carlo method.

Monte Carlo methods are based on drawing samples from a probability distribution and thus finding averages. This is what we are doing in this project: we are drawing samples from a uniform probability distribution, which determine whether or not the proposed new state will be accepted or not, and from this evolution of the system we calculate the average values of energy and magnetisation.

2.10.2 The Metropolis algorithm, sampling rule

The Metropolis algorithm is essentially a sampling rule used in a series of Monte Carlo cycles.

The sampling rule is as follows:

- If the proposed transition is towards a state of *higher* probability (according to the Boltzmann distribution): Accept 100% of these transitions.
- If the proposed transition is towards a state of *lower* probability: Generate a random real number r between 0 and 1 from a uniform distribution. If $r < p_{\text{after}}/p_{\text{before}} \equiv e^{-\beta(E_{\text{after}}-E_{\text{before}})} \equiv e^{-\beta\Delta E}$, accept the transition. Otherwise, reject the transition.

As shown by Eq. 8 the probability of the system being in the state s depends on the partition function, given in Eq. 9. This quantity is very difficult to compute since it requires all the states. The huge advantage of the Metropolis algorithm is that it only requires the ratio of probabilities between states, and therefore it is not necessary to compute the partition function.

Pseudocode for the Metropolis algorithm is shown in algorithm 1. Note that zero-indexing has been used in this pseudocode.

Algorithm 1: The Metropolis algorithm: One Monte Carlo cycle

```
N = L*L (number of spins in lattice)
for i = 1,2,..., N do
    /* Get a random spin in the lattice:  */
    iy = random number between 0 and L-1
    ix = random number between 0 and L-1
    calculate deltaE
    weight = exp(-beta*deltaE)
    randNum = random floating-point number between 0 and 1
    if randNum < weight then
        /* Perform the spin flip:  */
        spinMatrix(iy, ix) *= -1
        /* Update the energy and magnetisation:  */
        E += deltaE
        M += 2*spinMatrix(iy, ix)
```

Algorithm 1 shows one Monte Carlo cycle of the Metropolis algorithm. In the complete program many Monte Carlo cycles are performed in order to reach equilibrium. The values of the energy and magnetisation are saved in lists, and the mean values are computed from these lists. It is the sampling rule of the Metropolis algorithm that ensures that this simulation of the spin system will converge towards the physically most likely state, based on the Boltzmann distribution. More details on the program is found in the Methods section.

2.11 A simple 2×2 lattice, analytical expressions

Here we will assume that we have a total of four spins, organized in a 2×2 lattice. Since each of the four spins has two options, either up or down, the total number of states for this system is $2^4 = 16$. We want to find the analytic expressions for the energy, magnetisation, specific heat and magnetic susceptibility. The derivations of these expressions can be found in the appendix.

The system has degeneracies for certain energy levels, and these are shown in Table 1. Using this table, the partition function is calculated to be

$$Z = 4\cosh(8J\beta) + 12. \tag{24}$$

Table 1: Spin configurations in a 2×2 matrix of spins in which there are 4 spins and $2^4 = 16$ different configurations. PBC is an abbreviation for periodic boundary conditions.

Number of up spins	Degeneracy	Energy (PBC)	Magnetisation
4	1	-8J	4
3	4	0	2
2	4	0	0
2 (diagonal)	2	+8J	0
1	4	0	-2
0	1	-8J	-4

Table 2: Analytic expectation value for a 2×2 spin matrix with $J = 1$, $k_B = 1$ and $T = 1\,\mathrm{J}$. All values have been divided by $N = 4$ since there are four spins.

$\mathbb{E}[E]$	$\mathbb{E}[\lvert M\rvert]$	C_V	χ
-1.99598	0.99866	3.2082×10^{-2}	4.0107×10^{-3}

Using eq. (12) and the explicit expression for Z we can calculate the expectation value of the energy for this system:

$$\mathbb{E}[E] = -\frac{1}{Z} \cdot 32J \sinh(8J\beta) \tag{25}$$

Starting by evaluating $\mathbb{E}[E^2]$ for the 2×2 case and then using eq. (19) the analytic expression for the heat capacity at constant volume can be found.

$$\mathbb{E}[E^2] = \frac{1}{Z} \cdot 256J^2 \cosh(8J\beta). \tag{26}$$

This gives the heat capacity for a 2×2 spin matrix,

$$C_V = \frac{256J^2}{k_B T^2 Z} \left[\cosh(8J\beta) - \frac{1}{Z}\sinh^2(8J\beta)\right]. \tag{27}$$

The same process can be done for the magnetic susceptibility. First we find $\mathbb{E}[M^2]$,

$$\mathbb{E}[M^2] = \frac{1}{Z} \cdot 32\left(e^{8J\beta} + 1\right), \tag{28}$$

and then we find $\mathbb{E}[|M|]$,

$$\mathbb{E}[|M|] = \frac{1}{Z} \cdot 8\big(e^{8J\beta} + 2\big). \tag{29}$$

Then, using Eq. 17 we get an expression for the susceptibility:

$$\chi = \frac{32\beta}{Z}\left[1 + e^{8J\beta} - \frac{2}{Z}\big(e^{8J\beta} + 2\big)^2\right]. \tag{30}$$

Using $k_B = 1$ and $T = 1\,\mathrm{J}$ we have $\beta \equiv 1/(k_B T) = 1\,\mathrm{J}^{-1}$. From Eqs. (25, 55, 27, 30) we get the numerical values of the mean value quantities, which are shown in table 2.

3 Method

The spin system is simulated using the Metropolis algorithm. Our programs are written in C++ and Python. C++ is used to perform the heavy-duty calculations of the Metropolis algorithm. The C++ linear algebra library Armadillo is used for the handling of vectors and matrices. The results are saved in `.csv` files, and the plotting of the results is done in Python using the Matplotlib library. The C++ program was parallelised using OpenMP in order to speed up the computation time, allowing for longer simulations of the system.

3.1 Periodic boundary conditions

In the Ising model the spins affect and are affected by their nearest neighbours. At the edges and corners of the lattice, there are either one or two nearest neighbour slots that are empty since it is at the end of the lattice. So a boundary condition is needed for the edges. One common boundary condition is the periodic boundary condition, which means that at the end of a row or column, set the nearest neighbour spin in the empty slot equal to the spin at the beginning of the row or column, respectively.

A simple function can be used to implement the periodic boundary conditions for an $L \times L$ lattice. The function is as follows:

```
function PBC_idx(idx)
    return (L+idx)%L;
end
```

where `idx` is an integer (an index) and `%` is the modulo operator. In this function, the input `idx=-1` returns the index `L-1`, and the input `idx=L` returns the index `0`. Accessing the spins, and also changing them, while upholding the periodic boundary conditions, can then be done in a cost-effective manner. Below is an illustrative example where the four nearest neighbours around a spin are flipped:

```
iy = random integer between 0 and L-1
ix = random integer between 0 and L-1
spinMatrix(iy, PBC_idx(ix-1)) *= -1
spinMatrix(iy, PBC_idx(ix+1)) *= -1
spinMatrix(PBC_idx(iy-1), ix) *= -1
spinMatrix(PBC_idx(iy+1), ix) *= -1
```

This method is what we have used in our code to implement the periodic boundary conditions.

Two alternative methods that can be used to implement the periodic boundary conditions are shown in the appendix.

3.2 Calculating the energy and magnetisation of the spin lattice

Calculating the current magnitude of the net magnetisation, M: The magnetisation is calculated simply as the sum of all of the spins in the lattice. The magnitude is simply the absolute value of the sum.

```
M = 0
for row indices r = 0,1,...,L-1
    for column indices c = 0,1,...,L-1
        M = M + spinMatrix(r,c)
    end for
end for
M_abs = abs(M)
```

Calculating the current energy, E: The energy is calculated using the interaction energy between nearest neighbours as seen in eq. (6).

```
// Get the energy sum of all of the row strings in the lattice:
for row indices r = 0,1,...,L-1 // For each row, get the binding energies between the
// spins in that row.
    for column indices c = 0,1,...,L-1
        s1 = spinMatrix(r, c);
        s0 = spinMatrix(r, PBC_idx(c-1)); // Previous spin in row
        E += s1*s0;
    }
}
// Get the energy sum of all of the column strings in the lattice:
for column indices c = 0,1,...,L-1 // For each column, get the binding energies between the
// spins in that column.
    for row indices r = 0,1,...,L-1
        s1 = spinMatrix(r, c);
        s0 = spinMatrix(PBC_idx(r-1), c); // Previous spin in column
        E += s1*s0;
    }
}
E *= -J; // Multiply with the factor -J to get the correct energy.
```

3.3 Energy jumps and pre-calculation of weights

As seen in Eqs. (21, 22), there are only five possible different energy jumps ΔE and five different values for the weights. In our program we have save some computation time by choosing $J = 1$, since then the values of ΔE are all integers. And these integers can be used directly as indices in an array. By pre-calculating the values of the weights in an array $\mathbf{w} = e^{-8J\beta}, e^{-7J\beta}, \ldots, e^{7J\beta}, e^{8J\beta}$ (17 elements), we can add the number 8 to the value of $\Delta E \in \{-8, -4, 0, 4, 8\}$ in order to directly get the corresponding weight from $\mathbf{w}$. In code this can be written as `weight = w(deltaE+8)`. This is an efficient way of getting the weigh corresponding to an energy jump ΔE without needing to calculate it every time.

3.4 Running the Metropolis algorithm

Initialisation: The spin system was initialised in a random spin state, meaning that each spin was given randomly spin up or spin down with a 50/50 probability. We tested the system first with a 2×2 lattice. Our main simulations were of bigger lattices between 20×20 spins and 100×100 spins.

Performing the Monte Carlo cycles: For each Monte Carlo cycle, $N = L \cdot L$ random positions are generated and a flip is tested for each of those random positions, and some of those flips are performed, determined by the sampling rule of the Metropolis algorithm. The energy and magnetisation are updated after every performed spin flip according to equations (23) and (20). After the Metropolis algorithm is finished, that is, after all the Monte Carlo cycles have been performed, the results are exported in a `.csv` file which can then be read by a Python script for plotting.

3.5 Calculating mean values and variances from the Metropolis results

Our program, when running the Metropolis algorithm, saves the energy and magnetisation, as well as the squares of their values, are saved in lists. One value is saved for each Monte Carlo cycle. The mean values of these lists are what enables us to calculate the specific heat and the magnetic susceptibility of the spin system. The mean values are calculated using the `mean()` function from the Armadillo library; this is the same as the sample mean as defined in Eq. (14).

In our calculations we discard at least the first 10% of the Monte Carlo cycles before starting to store the values for energy and magnetisation for calculating the mean values. This is done in order to reach the equilibrium state first since we are interested in the equilibrium states of a given temperature. Including the initial 10% would most likely give a slight error in the mean values compared to the true expected values at equilibrium.

With $\langle E \rangle$ and $\langle E^2 \rangle$ we can calculate C_V directly using eq. (19). Similarly, with $\langle |M| \rangle$ and $\langle M^2 \rangle$ we can calculate χ using eq. (17). So running the whole Metropolis algorithm gives us one value for C_V and one value for χ, for the specified temperature T.

In order to study the phase transitions of the spin system in temperature T, we simply run the Metropolis algorithm one time for each temperature, in a range of temperatures. This way we are able to see how the averages of the system are affected by the temperature. The chosen temperature range was between $T = 2.0$ and $T = 2.36$, and the chosen temperature step was $dT \approx 0.02$.

4 Results

Unit Testing on the 2×2 system

Unit testing was performed on the 2×2 lattice. We created the system for all 16 possible spin configurations of the four spins and calculated the energy and magnetisation per spin for each of them. For all of the spin configurations the calculations of the energy and magnetisation agreed with the analytical expressions in Table 1.

We also tested the Metropolis algorithm on the spin system to see how the mean values compared to the analytical values. The results are seen in Table 3. For 10^5 Monte Carlo cycles, the mean energy had an error of 2.81×10^{-2}%, the mean magnetisation had an error of 2.00×10^{-2}%, the specific heat had an error of 13.89%, and the magnetic susceptibility had an error of 15.84%.

Evaluating the Number of Monte Carlo Cycles to reach Equilibrium

Figs. 1 and 2 show the mean energy and magnetisation respectively as functions of Monte Carlo cycles, beginning with both an ordered and random configuration. These figures are plotted between 20,000 and 100,000 Monte Carlo cycles. The insets shows all the data from 0 to 100,000 Monte Carlo cycles with the x axis in logarithmic scale. Equivalent plots, for increased temperature $T = 2.4$, are shown in Figs. 3 and 4 for energy and magnetisation respectively.

The number of accepted flips vs. Monte Carlo cycles

Fig. 5 shows the number of accepted configurations divided by the total possible configurations. An accepted configuration is determined by the Metropolis algorithm and results in a spin being flipped or not.

The distribution of the mean value of E

Fig. 6 shows the energy distribution, which approximates its probability distribution, after reaching the equilibrium state. It is plotted with 20 data bins.

Calculating the critical temperature T_C

The expectation values are calculated in the temperature range between $T = 2.0$ and $T = 2.36$ with steps of $dT = 0.01875$, and the results are shown in Fig. 7.

Estimating the critical temperature in the thermodynamic limit

By re-arranging Eq. 4, the critical temperature in the thermodynamic limit can be estimated (the exact value $\nu = 1$ was used). Fig. 8 shows the calculated T_C values plotted against $1/L$. A best fit line was then plotted through these points. The gradient and intercept of this line corresponds to the constant a and $T_C(L = \infty)$, respectively. This gives $T_C(L = \infty) = 2.273$ (to 3 decimal places), which to the exact result has a relative error equal to 1.76×10^{-3}. As a percentage this is an error of 0.176 %.

Table 3: Results for the mean values from running the Metropolis algorithm on the 2×2 spin system for different numbers of Monte Carlo cycles.

Monte Carlo cycles	$\mathbb{E}[E]$	$\mathbb{E}[\lvert M \rvert]$	C_V	χ
Analytic value	-1.99598	0.99866	3.2082×10^{-2}	4.0107×10^{-3}
10^3	-1.99556	0.99889	3.5437×10^{-2}	3.6148×10^{-3}
10^4	-1.99533	0.99861	3.7242×10^{-2}	3.6585×10^{-3}
10^5	-1.99542	0.99846	3.6538×10^{-2}	4.6460×10^{-3}

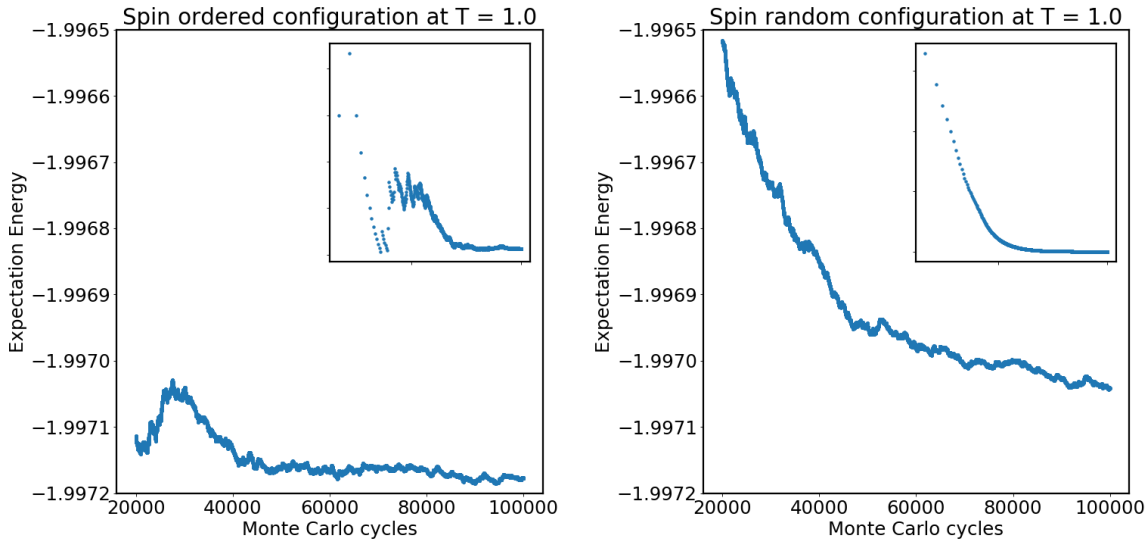


Figure 1: Expectation energy as a function of Monte Carlo cycles. (a) The spin matrix is initialised with all spins equal to $+1$ and in (b) with random spin configuration. The spin matrix has dimensions 20×20 and the system is at temperature $T = 1.0$. Insets show the full range from 0 to 100,000 cycles as a log scale; no tick labels are shown.

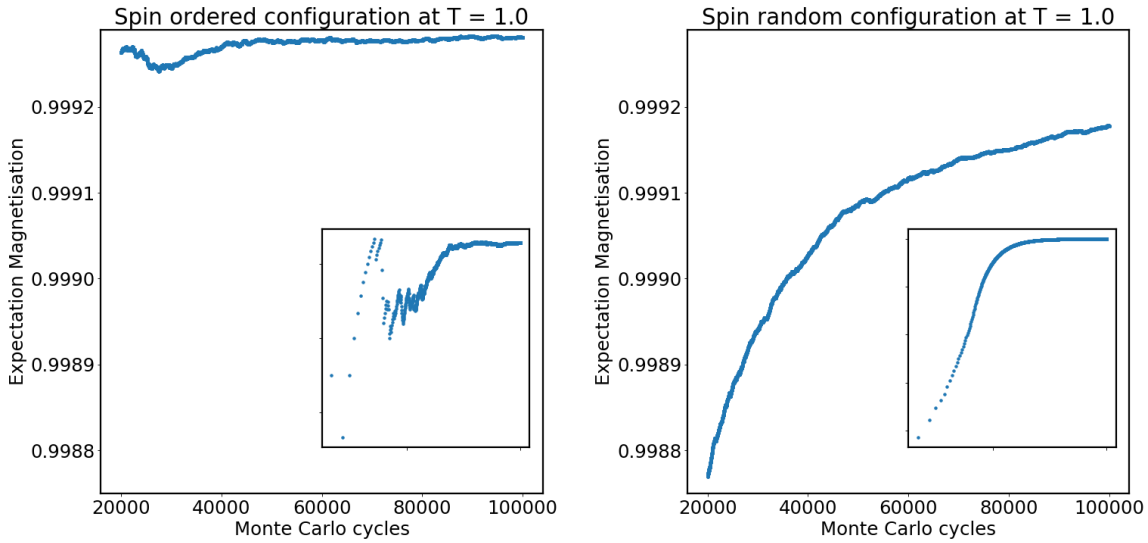


Figure 2: Expectation magnetisation as a function of Monte Carlo cycles. (a) The spin matrix is initialised with all spins equal to +1 and in (b) with random spin configuration. The spin matrix has dimensions 20×20 and the system is at temperature $T = 1.0$. Insets show the full range from 0 to 100,000 cycles as a log scale; no tick labels are shown.

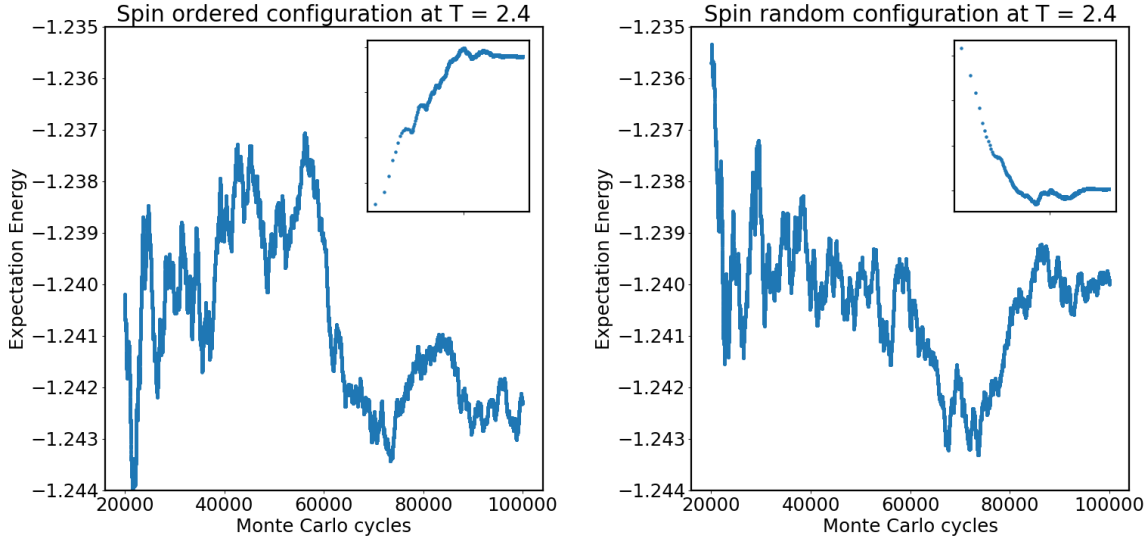


Figure 3: Expectation energy as a function of Monte Carlo cycles. (a) The spin matrix is initialised with all spins equal to +1 and in (b) with random spin configuration. The spin matrix has dimensions 20×20 and the system is at temperature $T = 2.4$. Insets show the full range from 0 to 100,000 cycles as a log scale; no tick labels are shown.

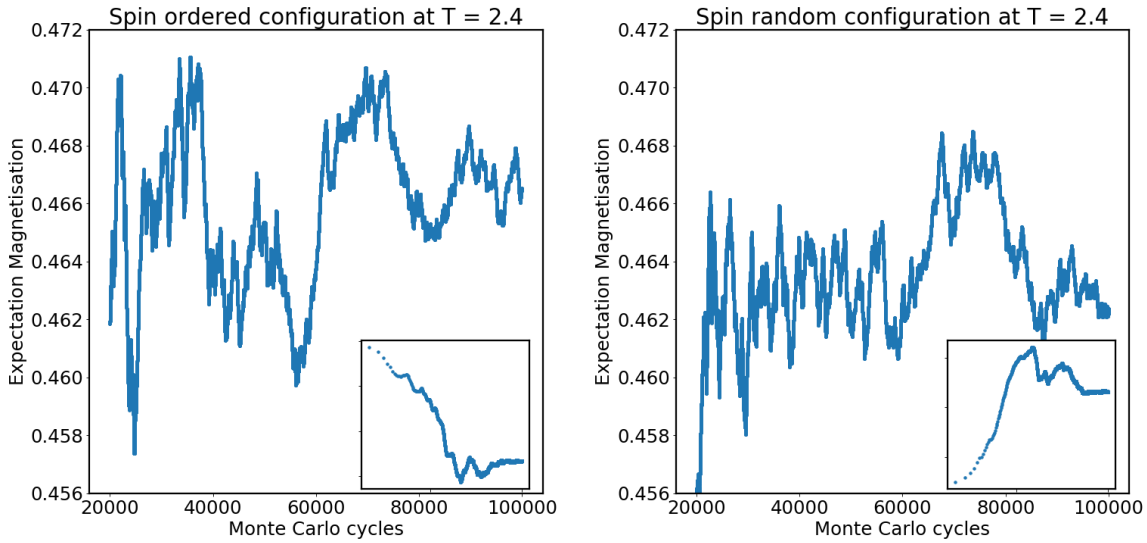
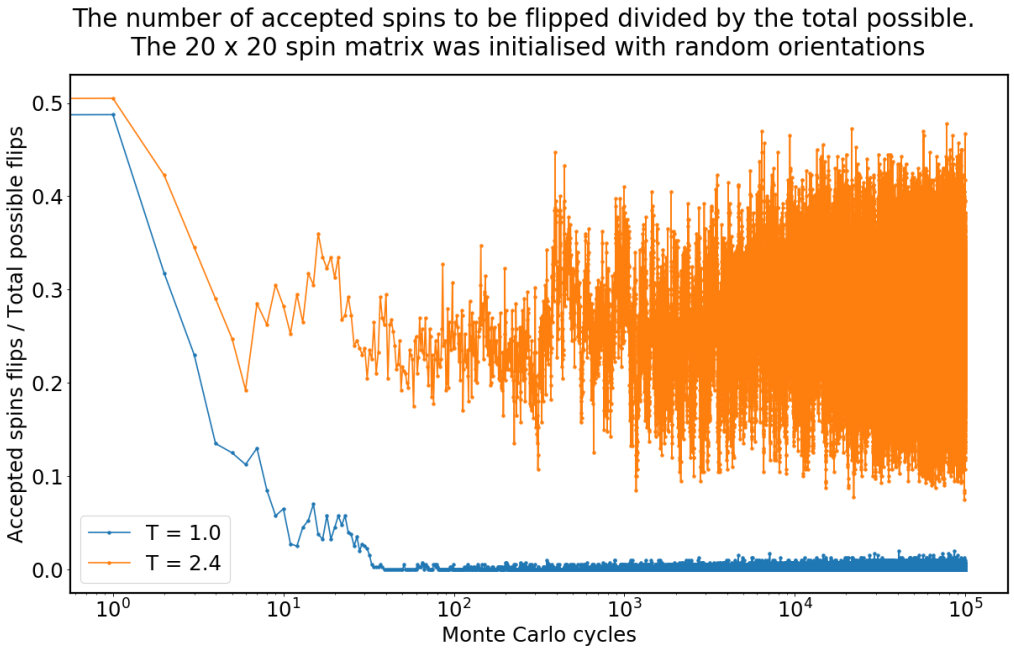
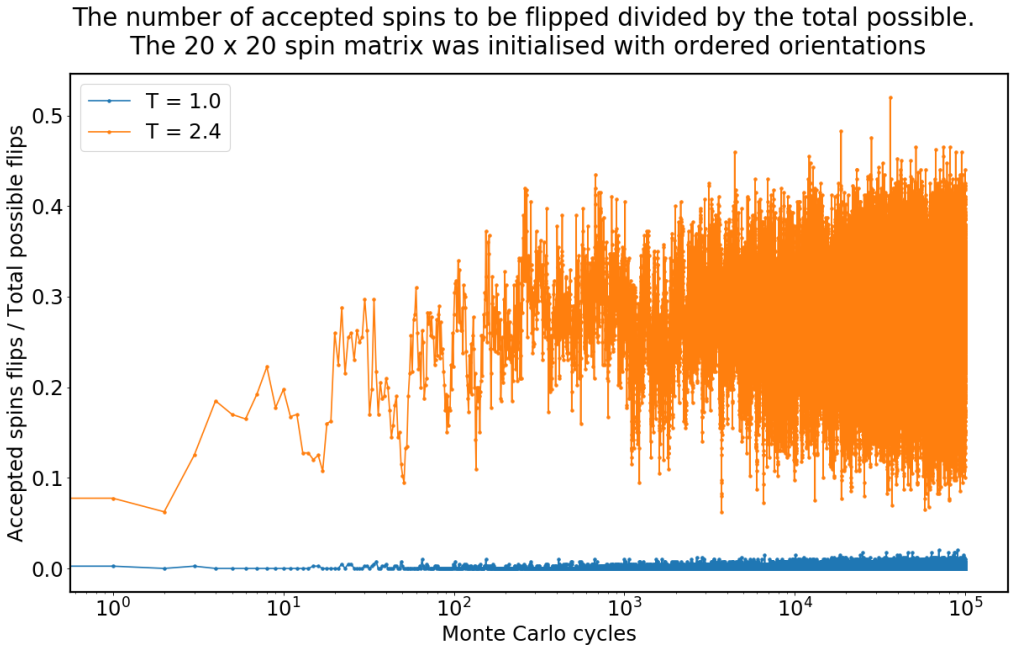


Figure 4: Expectation magnetisation as a function of Monte Carlo cycles. (a) The spin matrix is initialised with all spins equal to +1 and in (b) with random spin configuration. The spin matrix has dimensions 20×20 and the system is at temperature $T = 2.4$. Insets show the full range from 0 to 100,000 cycles as a log scale; no tick labels are shown.

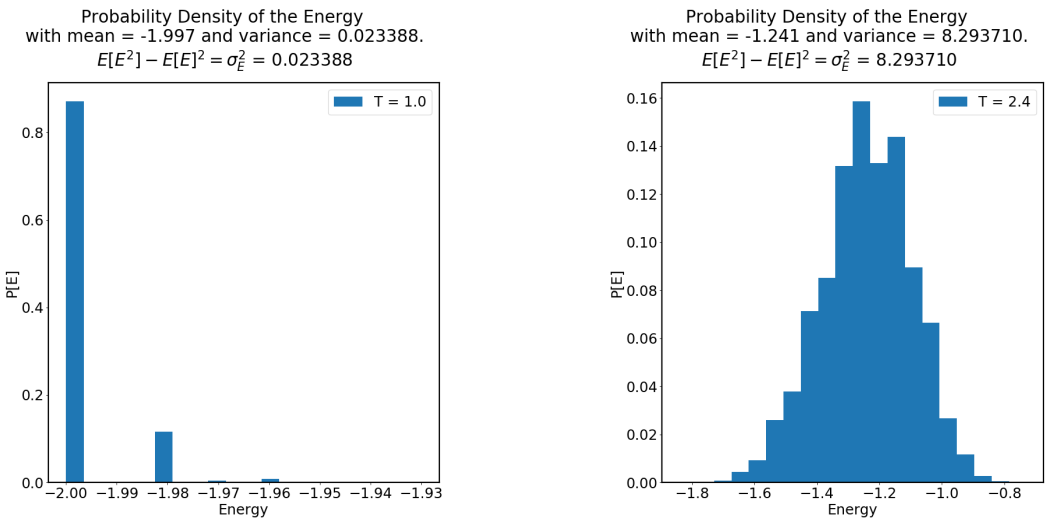


(a)



(b)

Figure 5: The number of accepted spins flips divided by total possible flips as a function of Monte Carlo cycles. This is for a 20×20 spin matrix, initialised with a (a) random and (b) ordered spin configuration.



(a) (b)

Figure 6: The probability density of the energy calculated with 10^5 Monte Carlo cycles with 20% for reaching equilibrium. (a) T = 1.0 (b) T = 2.4. Both sub-figures were run beginning with a randomly ordered spin matrix.

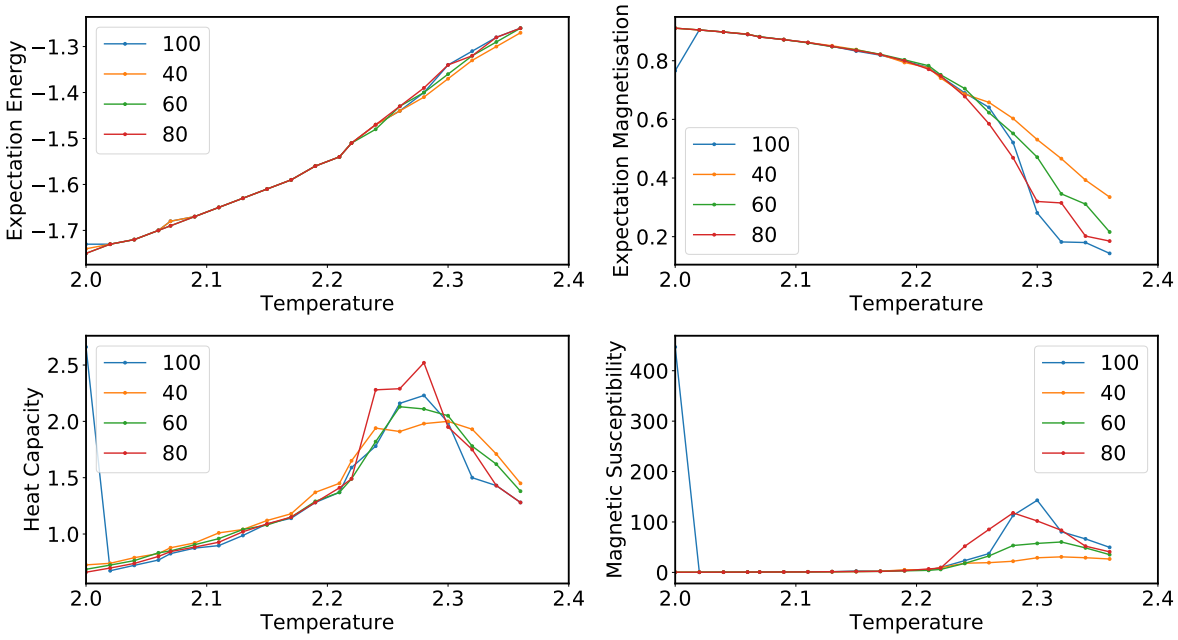


Figure 7: Expectation values as a function of temperature. Different colours indicate different values of L, the lattice size. 10^5 Monte Carlo cycles with 20% for reaching equilibrium were used. The temperature ranges between $T = 2.0$ and $T = 2.36$ with steps of $dT = 0.01875$.

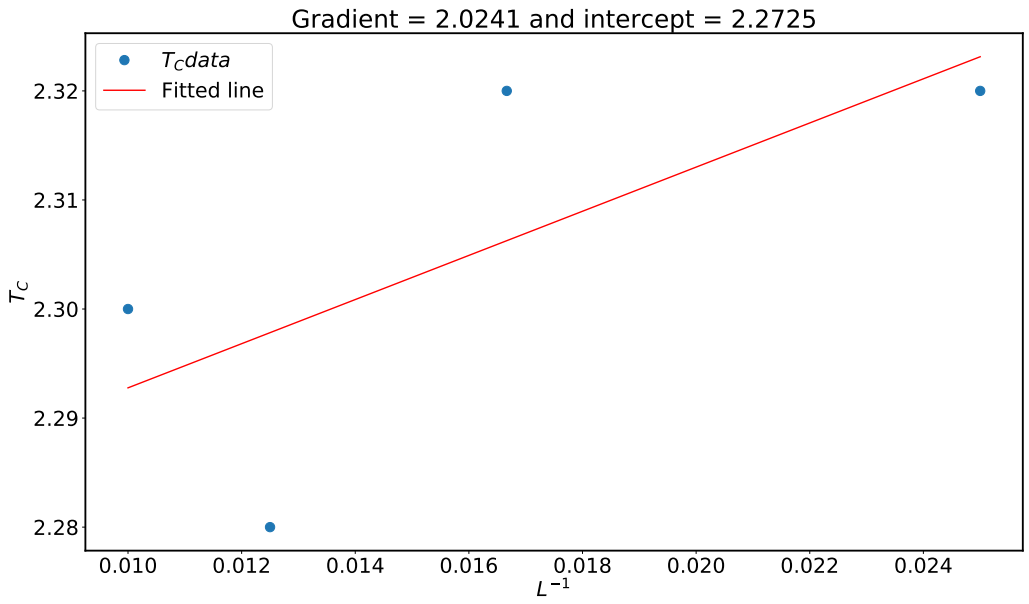


Figure 8: The critical temperatures extracted from Fig. 7 plotted against 1/L, where L is the spin matrix dimensionality. By making a best fit through these points the gradient corresponding to a and the intercept corresponding to $T_C(L = \infty)$ is evaluated using Eq. 4.

5 Discussion

5.1 Unit Testing on the 2×2 system

The results in Table 3 show that the results for the mean values agree with the analytic values of $\mathbb{E}[E]$ and $\mathbb{E}[|M|]$ to a good degree. The agreement for the heat capacity and the susceptibility were not that good though, at $\approx 14\%$ and $\approx 16\%$ respectively. The reason for this could be that the differences $\mathbb{E}[E^2] - \mathbb{E}[E]^2$ and $\mathbb{E}[M^2] - \mathbb{E}[|M|]^2$ used in calculating C_V and χ are small numbers, so a small error in any of these mean values will cause these differences to be relatively quite a bit bigger than their analytical values, giving a larger error. The number of Monte Carlo cycles were large though, up to 100 000, and the 2×2 system converged fast. Also, the first 10% of the Monte Carlo cycles were discarded in order to avoid errors in the mean values. So exactly why the results did not converge closer to the analytical values for the specific heat and the susceptibility is not clear.

5.2 The number of Monte Carlo cycles required to reach equilibrium

As seen in Fig. 1, after 100,000 Monte Carlo cycles the mean energy differs between (a) and (b) by approximately 0.002. We can expect that by performing more Monte Carlo cycles these two curves will converge to the same value. After approximately 20,000 Monte Carlo cycles a satisfactory equilibrium state is reached. Although the curve in (b) is still decreasing, the rate of this decrease is small. Between 20,000 and 100,000 Monte Carlo cycles the mean energy changes by approximately 5×10^{-4}. For this reason it can be said that this has reached the equilibrium state. A similar trend can be seen for the mean magnetisation in Fig. 2.

Going to a higher temperature $T = 2.4$, the change in mean energy between 20,000 and 100,000 Monte Carlo cycles is approximately an order of magnitude greater. In Fig. 3 (a) the range in which the mean varies is 0.007. This suggests that at higher temperatures the stability of the mean energy is lower and therefore more Monte Carlo cycles could be necessary. As was the case in Fig.1 after 100,000 Monte Carlo cycles the mean energy differs between (a) and (b) by approximately 0.002. With more Monte Carlo cycles these two should converge to the same value. Similar trends are seen for the mean magnetisation in Fig. 4.

For the 20×20 spin matrix it was found that by running 100,000 Monte Carlo cycles with 20,000 Monte Carlo cycles for equilibrium was satisfactory, as shown in Figs. 1 - 4. More Monte Carlo cycles would, however, improve the expectation values since there is a discrepancy between the runs with ordered and randomly initialised spin matrices.

5.3 The number of accepted spin flips vs. performed Monte Carlo cycles

Fig. 5 indicates that when the temperature is equal to 2.4 more of the spins are flipped than when $T = 1.0$. It is also seen that there is more variance for $T = 2.4$ between Monte Carlo cycles. This suggests that as temperature increases it is more likely for a spin to be flipped in the spin matrix. This is reasonable when taking into account the weights and the sampling rule in the Metropolis algorithm. The weights have, as mentioned in the theory section, values $e^{-\beta \Delta E}$, and the random number determining whether the spin is accepted or not is between 0 and 1. The temperature affects this because $\beta = 1/(k_B T) = 1/T$ with $k_B = 1$. The maximum/minimum values for the weights are $e^{\pm 8/T}$. For $T = 1.0$ these values are $e^{\pm 8} \in \{3.35 \times 10^{-4}, 2981\}$. For $T = 2.4$ these values are $e^{\pm 8/2.4} \in \{3.57 \times 10^{-2}, 28\}$. So the higher temperature increases the lowest weight and lowers the highest weight. But since the random number is only between 0 and 1 all values above 1, meaning all positive exponents, are automatically accepted, so when the negative exponents are *less negative* (since they are divided by $T = 2.4$), those probabilities for being accepted are increased. So it makes sense that the number of accepted spins should be greater on average for the higher temperature.

5.4 The probability density

As seen in Fig. 6, the variance of the energy distributions agree with the computed variance in energy σ_E^2 as shown in the figure title. The probability density for $T = 1.0$ shows that the energy is approximately equal to -2.00 with a probability of more than 0.8. This is likely why, as shown in Fig. 5 at $T = 1.0$, there are very few spin flips accepted. On the other hand, the probability density for $T = 2.4$ shows a much larger variance and larger mean than $T = 1.0$. This suggests that at each Monte Carlo cycle the energy is oscillating around its mean value, although the mean can be changing as is shown in Fig. 1.

5.5 The critical temperature T_C

Fig. 7 indicates that in the temperature range between 2.0 and 2.36 there is a phase transition. Both mean energy and mean absolute magnetisation are continuous as expected. However, specific heat has a peak for all spin matrix sizes around 2.6 approximately. This discontinuity suggests there is a phase transition and the peak can be used to indicate the critical temperature. The magnetic susceptibility should also be discontinuous, but this is not clear from the figure. As mentioned in the theory, the magnetic susceptibility should be undefined below the critical temperature and shows a sudden change just above T_C. The data for $L = 100$ has large deviations at $T = 2.0$. It could be a numerical error in the program, or perhaps a systematic error, but we are not certain about this.

The critical temperature was evaluated for several spin matrices with dimensions $L = 40, 60, 80$ and 100. This was done by sweeping a temperature range between 2.0 and 2.36. The peak in specific heat indicated the presence of a phase transition and also gave an estimate of T_C.

Using the calculated T_C values for each spin matrix, the critical temperature in the thermodynamic limit was estimated using Eq. 4. This gave the value $T_C = 2.273J/k_B$ and compared to the exact result, $2.269J/k_B$. The relative error between the estimated value and the exact result is 0.176 %. This suggests that the simulations in this report are in good agreement with Lars Onsager's exact result.

6 Conclusion

In conclusion, the ferromagnetic phase transition was successfully modelled using the Ising model in 2 dimensions. The model was tested against the analytic solution for a 2×2 spin matrix and found to be in very good agreement with the mean energy and mean magnetisation, and reasonably good agreement with the specific heat and the magnetic susceptibility.

The model was extended to using a larger 20×20 matrix, and the number of Monte Carlo cycles necessary for the system to reach equilibrium was investigated. It was found that running for a total of 100,000 Monte Carlo cycles with 20,000 for equilibrium was satisfactory. There is room for improvement, however, since there was a discrepancy (0.002) for example in the energy at $T = 1.0$ between the spin matrices initialised with ordered and random spins. More Monte Carlo cycles would likely improve the accuracy of the expectation values.

By parallelising the C++ code it was possible to run over many nodes (processors) and compute the expectation values for several matrices with dimensions $L = 40$, 60, 80 and 100. After confirming that there was indeed a phase transition in the temperature range between 2.0 and 2.36, the critical temperatures were extracted and plotted against $1/L$. These points allowed a fitting to be made and the intercept of this line through the y-axis gave the estimate result for the critical temperature as L goes to infinity. The result was $T_C = 2.273 J/k_B$ with a relative error of 0.176 % compared to Lars Onsager's exact solution.

7 Github address

Github address: `https://github.com/amundwf/comp-phys-project4.git`

References

[1] Barry M McCoy and Tai Tsun Wu. *The two-dimensional Ising model*. Courier Corporation, 2014.

[2] Hagen Kleinert and Verena Schulte-Frohlinde. *Critical Properties of [Greek Letter Phi] 4-theories*. World Scientific, 2001.

[3] Morten Hjorth Jensen. Computational physics, lecture notes 2015, 2015. `https://github.com/CompPhysics/ComputationalPhysics/blob/master/doc/Lectures/lectures2015.pdf`, Retrieved September 9, 2020.

[4] Richard Fitzpatrick. The ising model, 2006. `http://farside.ph.utexas.edu/teaching/329/lectures/node110.html`, Retrieved October 30, 2020.

[5] Franz Schwabl. Statistical mechanics. 2002.

[6] Marcin Wieśniak, Vlatko Vedral, and Časlav Brukner. Magnetic susceptibility as a macroscopic entanglement witness. *New Journal of Physics*, 7(1):258, 2005.

[7] Charles Kittel. *Introduction to solid state physics*. Wiley, 2004.

A Ferromagnetism

Ferromagnetism is the state of a material where there is a spontaneous magnetic moment, i.e it has magnetisation even without any applied field. We can consider a paramagnetic material of some number of atoms, each with a spin, and these spins are not in anyway aligned. Now consider that there is an internal interaction which causes these spins to align with other, this is called ferromagnetism. This internal interaction can be called the exchange field B_E and is assumed to be proportional to the magnetisation M. In the mean field approximation, atoms experience the effect of all the atoms in the material, although it may really be that they only feel the presence of their closest neighbours. The exchange field is then

$$B_E = \lambda M, \tag{31}$$

where λ is some temperature independent constant [7].

If we look at the paramagnetic phase of a material and apply a external field B_A to it, the magnetisation is given by

$$M = \chi(B_A + B_E), \tag{32}$$

where χ is the magnetic susceptibility. Using the Curie law that $\chi = C/T$ where C is the Curie constant, we can now substitute Eq. 31 into Eq. 32 and re-arrange to give

$$\chi = \frac{M}{B_A} = \left(\frac{C}{T - \lambda C}\right). \tag{33}$$

This is known as the Curie-Weiss law, which works well in the region just above the Curie temperature. It gives a singularity when λC is equal to T. This temperature is known at the critical temperature or Curie temperature T_C. Below it, the magnetic susceptibility is undefined and the material is in the ferromagnetic phase. Above this temperature the magnetic susceptibility is defined and the material is in the paramagnetic phase. Eq. 33 means that when the magnetic susceptibility is infinite we can have zero applied field, but still have a finite magnetisation [7].

Using the definition of the Curie temperature T_C it is possible to calculate the mean field constant λ. In iron for example, this gives $\lambda \approx 5000$, and with a magnetisation $M = 1700$, gives $B_E = 10^3 T$. The exchange field gives a value much greater than the real magnetic field of iron. The exchange field is an approximation of the quantum mechanics interaction between atoms with spin [7]. The energy of this interaction between atoms i and j with spin s_i and s_j can be written as

$$E = -2J s_i \cdot s_j, \tag{34}$$

where J is the exchange integral.

B Derivation of analytical expressions

The general expression for the specific heat C_V is

$$\begin{aligned} C_V &= \left(\frac{\partial \mathbb{E}[E]}{\partial T}\right)_V \\ &= -\frac{\partial}{\partial T}\left(\frac{\partial \ln Z}{\partial \beta}\right) \\ &= -\frac{\partial}{\partial \beta}\left(\frac{\partial \ln Z}{\partial \beta}\right)\frac{\partial \beta}{\partial T} \\ &= \frac{1}{k_B T^2}\frac{\partial^2 \ln Z}{\partial \beta^2}. \end{aligned}$$

B.1 Analytical expressions for the 2×2 spin system

The partition function: Using table 1 the partition function can be calculated from the expression in eq. (9):

$$\begin{aligned} Z &= 1 \cdot e^{-\beta \cdot (-8J)} + 4 \cdot e^{-\beta \cdot 0} + 4 \cdot e^{-\beta \cdot 0} + 2 \cdot e^{-\beta \cdot 8J} + 4 \cdot e^{-\beta \cdot 0} + 1 \cdot e^{-\beta \cdot (-8J)} && (35) \\ &= e^{+8J\beta} \cdot (1+1) + e^{-8J\beta} \cdot 2 + (4+4+4) && (36) \\ &= 4\left(\frac{e^{8J\beta} + e^{-8J\beta}}{2}\right) + 12 && (37) \\ &= 4\cosh(8J\beta) + 12 && (38) \end{aligned}$$

where the definition of the cosine hyperbole function, $\cosh x \equiv (e^x + e^{-x})/2$, has been used.

Expectation value of E: Using eq. (12) and the explicit expression for Z we can calculate the expectation value of the energy for this system:

$$\begin{aligned} \mathbb{E}[E] &= -\frac{1}{Z}\frac{\partial Z}{\partial \beta} && (39) \\ &= -\frac{1}{Z} \cdot 4\sinh(8J\beta) \cdot 8J && (40) \\ &= -\frac{1}{Z} \cdot 32J \sinh(8J\beta) && (41) \end{aligned}$$

Expected value of E^2: Starting by evaluating $\mathbb{E}[E^2]$ for the 2×2 case and then using eq. (19) the analytic expression for the heat capacity at constant volume can be found.

$$\mathbb{E}[E^2] = \frac{1}{Z}\sum_{s}^{N} E_s^2 e^{-\beta E_s} \tag{42}$$

$$= \frac{1}{Z}\left(1 \cdot (-8J)^2 e^{+8J\beta} + 2 \cdot (+8J)^2 e^{-8J\beta} + 1 \cdot (-8J)^2 e^{+8J\beta} + 0\right) \tag{43}$$

$$= \frac{1}{Z} 4 \cdot (8J)^2 \left(\frac{e^{8J\beta} + e^{-8J\beta}}{2}\right) \tag{44}$$

$$= \frac{1}{Z} \cdot 256J^2 \cosh(8J\beta) \tag{45}$$

Specific heat: This expressions for $\mathbb{E}[E]$ and $\mathbb{E}[E^2]$ enable us to calculate the expression for the specific heat C_V:

$$C_V = \frac{1}{k_B T^2}\left(\mathbb{E}[E^2] - \mathbb{E}[E]^2\right) \tag{46}$$

$$= \frac{1}{k_B T^2}\left((8J)^2 \frac{\cosh(8J\beta)}{\cosh(8J\beta)+3} - \left(-8J\frac{\sinh(8J\beta)}{\cosh(8J\beta)+3}\right)^2\right) \tag{47}$$

$$= \frac{1}{k_B T^2}\left[\frac{1}{Z} \cdot 256J^2 \cosh(8J\beta) - \left(-\frac{1}{Z} \cdot 32J \sinh(8J\beta)\right)^2\right] \tag{48}$$

$$= \frac{1}{k_B T^2}\left[\frac{1}{Z} \cdot 256J^2 \cosh(8J\beta) - \frac{1}{Z^2} \cdot 1024J^2 \sinh^2(8J\beta)\right] \tag{49}$$

$$= \frac{256J^2}{k_B T^2 Z}\left[\cosh(8J\beta) - \frac{1}{Z}\sinh^2(8J\beta)\right]. \tag{50}$$

Expected value of $|M|$:

$$\mathbb{E}[|M|] = \frac{1}{Z}\sum_{s=1}^{16} |M_s| e^{-\beta E_s} \tag{51}$$

$$= \frac{1}{Z}\Big[1 \cdot (|4| \cdot e^{-\beta\cdot(-8J)}) + 4 \cdot (|2| \cdot e^0) + 4 \cdot |0| + 2 \cdot |0| \tag{52}$$

$$+ 4 \cdot (|-2| \cdot e^0) + 1 \cdot (|-4| \cdot e^{-\beta\cdot(-8J)})\Big] \tag{53}$$

$$= \frac{1}{Z}\Big(4e^{8J\beta} + 8 + 8 + 4e^{8J\beta}\Big) \tag{54}$$

$$= \frac{1}{Z} \cdot 8\big(e^{8J\beta} + 2\big). \tag{55}$$

Expected value of M^2:

$$\mathbb{E}[M^2] = \frac{1}{Z}\sum_{s=1}^{16} M_s^2 e^{-\beta E_s} \tag{56}$$

$$= \frac{1}{Z}\Big[1 \cdot (4^2 \cdot e^{-\beta\cdot(-8J)}) + 4 \cdot (2^2 \cdot e^0) + 4 \cdot 0^2 + 2 \cdot 0^2 \tag{57}$$

$$+ 4 \cdot ((-2)^2 \cdot e^0) + 1 \cdot ((-4)^2 \cdot e^{-\beta\cdot(-8J)})\Big] \tag{58}$$

$$= \frac{1}{Z}\Big(16e^{8J\beta} + 16 + 16 + 16e^{8J\beta}\Big) \tag{59}$$

$$= \frac{1}{Z} \cdot 32\left(e^{8J\beta} + 1\right). \tag{60}$$

Magnetic susceptibility: Using Eq. 17 we get

$$\chi = \beta\left(\mathbb{E}[M^2] - \mathbb{E}[|M|]^2\right) \tag{61}$$

$$= \beta\left[\frac{1}{Z}\cdot 32\left(e^{8J\beta}+1\right) - \left(\frac{1}{Z}\cdot 8\left(e^{8J\beta}+2\right)\right)^2\right] \tag{62}$$

$$= \beta\left[\frac{1}{Z}\cdot 32\left(e^{8J\beta}+1\right) - \frac{1}{Z^2}\cdot 64\left(e^{8J\beta}+2\right)^2\right] \tag{63}$$

$$= \frac{32\beta}{Z}\left[1 + e^{8J\beta} - \frac{2}{Z}\left(e^{8J\beta}+2\right)^2\right]. \tag{64}$$

C Periodic boundary conditions, alternative methods

In this section, some alternative methods for implementing the periodic boundary conditions of the spin system is presented. These methods are less efficient than the method we ended up using in our program to our knowledge.

If-statements

```
iy = random integer between 0 and L-1
ix = random integer between 0 and L-1
if iy==0
    if ix==0
        left = spinMatrix(0, L-1)
        right = spinMatrix(0, ix+1)
        above = spinMatrix(L-1, ix)
        below = spinMatrix(1, ix)
    else if ix==L-1
        left = spinMatrix(0, ix-1)
        right = spinMatrix(0, 0)
        above = spinMatrix(L-1, ix)
        below = spinMatrix(1, ix)
    else
        left = spinMatrix(0, ix-1)
        right = spinMatrix(0, ix+1)
        above = spinMatrix(L-1, ix)
        below = spinMatrix(1, ix)
else if iy==L-1
    if ix==0
        left = spinMatrix(iy, L-1)
        right = spinMatrix(iy, ix+1)
        above = spinMatrix(iy-1, ix)
        below = spinMatrix(0, ix)
    else if ix==L-1
        left = spinMatrix(iy, ix-1)
        right = spinMatrix(iy, 0)
        above = spinMatrix(iy-1, ix)
        below = spinMatrix(0, ix)
    else
        left = spinMatrix(iy, ix-1)
        right = spinMatrix(iy, ix+1)
        above = spinMatrix(iy-1, ix)
```

```
        below = spinMatrix(0, ix)
else
    if ix==0
        left = spinMatrix(iy, L-1)
        right = spinMatrix(iy, ix+1)
        above = spinMatrix(iy-1, ix)
        below = spinMatrix(iy+1, ix)
    else if ix==L-1
        left = spinMatrix(iy, ix-1)
        right = spinMatrix(iy, 0)
        above = spinMatrix(iy-1, ix)
        below = spinMatrix(iy+1, ix)
    else
        left = spinMatrix(iy, ix-1)
        right = spinMatrix(iy, ix+1)
        above = spinMatrix(iy-1, ix)
        below = spinMatrix(iy+1, ix)
```

This method has the downside that it for every spin it evaluates several if-statements to check for boundaries. This demands a lot of extra runtime and is more wasteful the larger the spin system is since the boundaries will be a relatively small part of the system. Below is another method, using an extended spin matrix.

Extended spin matrix The extended spin matrix, or periodic boundary condition (PBC) matrix, is the usual spin matrix containing all of the spins in the system, but it has 'one extra layer' in the sense that it has one extra row/column at each of the four sides of the matrix. The corners of the PBC matrix are not important since they will not be accessed, so they can have any values. The first column/row of the PBC matrix is equal to the last column/row of the normal spin matrix and the last column/row of the PBC matrix is equal to the first column/row of the normal spin matrix.

```
iy = random integer between 0 and L-1
ix = random integer between 0 and L-1
left = PBC_spinMatrix(iy, ix-1)
right = PBC_spinMatrix(iy, ix+1)
above = PBC_spinMatrix(iy-1, ix)
below = PBC_spinMatrix(iy+1, ix)
```

This makes it possible to access the spin matrix elements in an efficient way, without using if-statements. However, if a spin is to be flipped, the periodic boundary conditions must be upheld when flipping spins at any of the edges of the spin matrix.