-Project 4 - FYS3150/FYS4150-

By Simon Schrader (4150), Adrian Kleven (3150) - autumn 2019

Contents

List of Figures **2**

List of Tables **2**

1 Abstract **3**

2 Introduction **3**

3 Methods **3**
3.1 The Ising model in two dimensions 3
3.2 Statistical physics 5
3.2.1 2×2 Lattice 6
3.2.2 Phase transition 7
3.2.3 Extracting the critical temperature 8
3.3 The Metropolis algorithm 8
3.4 Periodic boundary conditions 10
3.5 Computational implementation 10
3.5.1 Units 10
3.5.2 Analytical solution 11
3.5.3 Efficient formulas 11
3.5.4 Simulating the system 12
3.5.5 Parallelization 12
3.5.6 Unit testing 13
3.5.7 Calculating the critical temperature 13

4 Results **13**
4.1 Reaching the equilibrium state 13
4.2 Time benefit from Parallelization and compile flags 15
4.3 Results achieved using 2×2 lattice 15
4.4 Energy distribution in a 20×20 lattice 16
4.5 Behaviour of the system close to the critical temperature 17
4.6 Critical temperature 19

5 Conclusion **20**

6 Critique **20**

7 Appendix **21**
7.1 Figures 21
7.2 Tables 22
7.3 List of programs 22

8 References **23**

List of Figures

1 Average energy, average absolute magnetisation & accepted spins at T=1.0 14
2 Average energy, average absolute magnetisation & accepted spins at T=2.4 14
3 Energy probability distributions in a 20×20 lattice at T=1.0 & T=2.4 16
4 $C_V, \chi, E, |M|$ for T between 2.0 and 2.35 17
5 $C_V, \chi, E, |M|$ for T between 2.2615 and 2.315 19
6 $C_V, \chi, E, |M|$ for T between 2.20 and 2.35 21

List of Tables

1 energies and magnetizations for a 2×2 lattice 4
2 Time usage parallel/non-parallel, compilation flags 15
3 2×2 lattice rel. errors with 10^7 simulation steps 22
4 2×2 lattice rel. errors with 10^6 simulation steps 22
5 Critical Temperatures 22

1 Abstract

Studying phase transitions as a function of temperature is an important aspect of the physical and chemical sciences. Following Boltzmann-statistics, analytical studies become infeasible as the size of the system increases. We implemented the two-dimensional Ising model for different lattice sizes and sampled using the Metropolis algorithm at varying temperatures. We found that it takes roughly 20,000 Monte Carlo cycles times the amounts of spins simulated for the system to equilibrate. Using the heat capacity, the critical temperature for an infinitely large system was found to be $T_C(L_{\rightarrow\infty}) \approx 2.269 \pm 0.002$, which is in accordance with the article by Lars Onsager [5]. We also found that the energy distribution clearly resembles the expected Boltzmann distribution.

2 Introduction

The purpose of this article is to apply the widely popular Ising model in two dimensions to simulate phase transitions in a binary spin, two- dimensional lattice using periodic boundary conditions.
As the system's state depends on the partition function Z, a function of all microstates, analytical descriptions of the system become infeasible, as the amount of microstates is given as $\#microstates = 2^N$ where N is the amount of particles simulated. Thus, even with the help of a powerful computer, solving the analytical function even for small N, such as N=400, would exceed the universe's lifetime.
In order to get an approximation of the system's distribution function, we will apply a Markov chain Monte Carlo method, specifically the Metropolis algorithm. By simulating such a lattice for varying lattice sizes, the aim is to extrapolate the critical temperature of a macroscopic solid using comparatively microscopic lattice sizes using approximate values for the energy and the magnetisation and their respective variance. By comparing values calculated from this model with the exact result obtained by Lars Onsager [5], we can evaluate the accuracy of the model. In addition, we study the energy distributions as functions of energy and compare them to the familiar Boltzmann distribution. Finally, the distribution of magnetization as a function of temperature, and the phase transition from spontaneous magnetization to zero magnetization is observed.

The model is implemented using C++ and parallelized using MPI. The effectiveness of parallelization and compiler flags is tested.

3 Methods

3.1 The Ising model in two dimensions

The Ising model is a way of modelling macroscopic systems by simulating interactions between nearest neighbouring microscopic entities.
The intent is to examine phase transitions in materials at finite temperatures by modelling the 2- dimensional atomic lattice as neighbouring units switching between binary spin states. In the absence of any external magnetic field their

interactions are determined by the equation

$$E = -J \sum_{<kl>}^{N} s_k s_l \tag{1}$$

$s_k = \pm 1$. Where E is the total energy of the system and the quantity N representing the total number of spins and J being the coupling constant expressing the strength of the interaction between neighboring spins [2]. The magnetization is simply a sum over all spin states

$$\mathcal{M}_{total} = \sum_i s_i \tag{2}$$

s_i being either 1 or -1 depending on the spin configuration. For a 2×2 lattice with binary microstates, there are 16 possible spin configurations:

$$\begin{array}{cccccccc} \uparrow\uparrow & \uparrow\uparrow & \uparrow\uparrow & \uparrow\uparrow & \uparrow\downarrow & \uparrow\downarrow & \uparrow\downarrow & \uparrow\downarrow \\ \uparrow\uparrow & \uparrow\downarrow & \downarrow\uparrow & \downarrow\downarrow & \uparrow\uparrow & \uparrow\downarrow & \downarrow\uparrow & \downarrow\downarrow \end{array}$$

$$\begin{array}{cccccccc} \downarrow\uparrow & \downarrow\uparrow & \downarrow\uparrow & \downarrow\uparrow & \downarrow\downarrow & \downarrow\downarrow & \downarrow\downarrow & \downarrow\downarrow \\ \uparrow\uparrow & \uparrow\downarrow & \downarrow\uparrow & \downarrow\downarrow & \uparrow\uparrow & \uparrow\downarrow & \downarrow\uparrow & \downarrow\downarrow. \end{array}$$

Using periodic boundary conditions, according to equation (1) these give instances of energy and magnetization as follows:

Table 1: List of energies and magnetizations for a 2×2 lattice with a binary spin configuration.

#Spin Up	Degeneracy	Energy	Magnetization
4	1	-8J	4
3	4	0	2
2	4	0	0
2	2	8J	0
1	4	0	-2
0	1	-8J	-4

The energy difference from a single spin transition is based on the assumption that every spin state is surrounded by four other spin states. This uses the properties of periodic boundary conditions, meaning the system has no effective surface, reminiscent of the bulk properties of materials. The change in energy from a spin flip is dependent on the Hamiltonian (1) and the state of the four surrounding spins.

$$\begin{array}{ccc} & \uparrow & \\ \uparrow & \uparrow & \uparrow \\ & \uparrow & \end{array} \Rightarrow \begin{array}{ccc} & \uparrow & \\ \uparrow & \downarrow & \uparrow \\ & \uparrow & \end{array} \qquad \Delta E = 8J$$

$$\begin{array}{ccc} & \downarrow & \\ \uparrow & \uparrow & \uparrow \\ & \uparrow & \end{array} \Rightarrow \begin{array}{ccc} & \downarrow & \\ \uparrow & \downarrow & \uparrow \\ & \uparrow & \end{array} \qquad \Delta E = 4J$$

$$\begin{array}{ccccccc} & \downarrow & & & & \downarrow & \\ \uparrow & \uparrow & \uparrow & \Rightarrow & \uparrow & \downarrow & \uparrow \\ & \downarrow & & & & \downarrow & \end{array} \qquad \Delta E = 0$$

The inverse operations naturally giving the negative of their corresponding ΔE, and each state being energetically invariant to all spins being flipped to their opposite.

3.2 Statistical physics

The Boltzmann distribution gives the probability of a system being in a certain configuration, and is expressed like this

$$P_i(\beta) = \frac{e^{-\beta E_i}}{Z}. \tag{3}$$

With $\beta = \frac{1}{kT}$ being the inverse of the product of the temperature and k, k being the Boltzmann constant. E_i is the energy of the system in the configuration i and Z is the partition function for the canonical ensemble

$$Z = \sum_{i=1}^{M} e^{-\beta E_i} \tag{4}$$

summing over all microstates i. [2]
As an example, the partition function for the Ising model on a 2×2 lattice using periodic boundary conditions is

$$Z = \sum_{i=1}^{16} e^{-\beta E_i} = 2\left(e^{-\beta(8J)}\right) + 2\left(e^{\beta(8J)}\right) + 12\left(e^{0}\right) \tag{5}$$

$$Z = 2\left(e^{-\beta(8J)} + e^{\beta(8J)}\right) + 12 \tag{6}$$

By virtue of the partition function Z, the sum of the probabilities $P_i(\beta)$ for all states $i \in [1, 2, \cdots, 16]$ is normalized to 1.

The mean energy of a system at a given temperature is

$$\langle E \rangle = \sum_{i=1}^{M} E_i P_i(\beta) = \frac{1}{\sum_{i=1}^{M} e^{-\beta E_i}} \sum_{i=1}^{M} E_i e^{-\beta E_i} \tag{7}$$

summing over the products of the energy of a configuration with the probability of the system being in that configuration, then dividing by the normalization factor Z.
The energy variance is calculated as

$$\sigma_E^2 = \langle E^2 \rangle - \langle E \rangle^2 = \frac{1}{\sum_{i=1}^{M} e^{-\beta E_i}} \sum_{i=1}^{M} E_i^2 e^{-\beta E_i} - \left(\frac{\sum_{i=1}^{M} E_i e^{-\beta E_i}}{\sum_{i=1}^{M} e^{-\beta E_i}} \right)^2 \tag{8}$$

The heat capacity of the system is calculated as

$$C_V = \frac{1}{kT^2}\left(\langle E^2 \rangle - \langle E \rangle^2\right) \tag{9}$$

with k again, being the Boltzmann constant and T being the temperature.

The mean magnetization is similarly calculated:

$$\langle \mathcal{M} \rangle = \sum_{i=1}^{M} \mathcal{M}_i P_i(\beta) = \frac{1}{\sum_{i=1}^{M} e^{-\beta E_i}} \sum_{i=1}^{M} \mathcal{M}_i e^{-\beta E_i} \tag{10}$$

with the variance

$$\sigma_{\mathcal{M}}^2 = \langle \mathcal{M}^2 \rangle - \langle \mathcal{M} \rangle^2 = \frac{1}{\sum_{i=1}^{M} e^{-\beta E_i}} \sum_{i=1}^{M} \mathcal{M}_i^2 e^{-\beta E_i} - \left(\frac{\sum_{i=1}^{M} \mathcal{M}_i e^{-\beta E_i}}{\sum_{i=1}^{M} e^{-\beta E_i}} \right)^2 . \tag{11}$$

The magnetic susceptibility then is defined as

$$\chi = \frac{1}{kT} \left(\langle \mathcal{M}^2 \rangle - \langle \mathcal{M} \rangle^2 \right) . \tag{12}$$

Because the mean magnetisation will always be zero in the Ising model, as the state with all spins inverted has the same energy and thus the same probability, we look at the absolute magnetisation and the absolute susceptibility, defined as

$$\langle |M| \rangle = \sum_{i=1}^{M} |\mathcal{M}_i| P_i(\beta) = \frac{1}{\sum_{i=1}^{M} e^{-\beta E_i}} \sum_{i=1}^{M} |\mathcal{M}_i| e^{-\beta E_i} \tag{13}$$

$$\chi_{abs} = \frac{1}{kT} \left(\langle \mathcal{M}^2 \rangle - \langle |\mathcal{M}| \rangle^2 \right) . \tag{14}$$

Each time we refer to the magnetic susceptibility, we in fact refer to χ_{abs}.

3.2.1 2×2 Lattice

From the analytical expressions, we can calculate the expectation value for the energy, the mean absolute value of the magnetic moment, the specific heat capacity and the magnetic susceptibility of the 2×2 lattice.
Starting with the energy terms, $\langle E \rangle$ and $\langle E^2 \rangle$ are calculated first using equation (7).

$$\langle E \rangle (\beta) = \frac{1}{\sum_{i=1}^{16} e^{-\beta E_i}} \sum_{i=1}^{16} E_i e^{-\beta E_i}$$

$$\langle E \rangle (\beta) = \frac{8J \left(e^{-\beta(8J)} - e^{\beta(8J)} \right)}{e^{-\beta(8J)} + e^{\beta(8J)} + 6} . \tag{15}$$

Inserting E_i^2 for E_i into equation (7) gives

$$\langle E^2 \rangle (\beta) = \frac{\sum_{i=1}^{16} E_i^2 e^{-\beta E_i}}{2 \left(e^{-\beta(8J)} + e^{\beta(8J)} \right) + 12}$$

$$\langle E^2 \rangle (\beta) = \frac{32J^2 \left(e^{-\beta(8J)} + e^{\beta(8J)} \right)}{\left(e^{-\beta(8J)} + e^{\beta(8J)} \right) + 6} . \tag{16}$$

From equation (8) we then get

$$C_V = \frac{1}{kT^2} \left(\frac{32J^2 \left(e^{-\beta(8J)} + e^{\beta(8J)} \right)}{\left(e^{-\beta(8J)} + e^{\beta(8J)} \right) + 6} - \left(\frac{8J \left(e^{-\beta(8J)} - e^{\beta(8J)} \right)}{e^{-\beta(8J)} + e^{\beta(8J)} + 6} \right)^2 \right) . \tag{17}$$

Next is calculating the mean absolute value of the magnetic moment. In doing this, the magnetization is treated the same if it is of the same magnitude but in the opposite direction. Magnetization from this point forward will mean exactly this. That means the instances of magnetization are as in table 1, but with negative instances of magnetization counting towards their respective absolute value. Using equation (13),

$$\langle \mathcal{M} \rangle = \frac{\left(2\left(4e^{\beta(8J)}\right) + 8\left(2 \cdot 1\right)\right)}{2\left(e^{-\beta(8J)} + e^{\beta(8J)}\right) + 12} = \frac{4\left(e^{\beta(8J)} + 2\right)}{e^{-\beta(8J)} + e^{\beta(8J)} + 6}.$$

For calculating $\langle \mathcal{M}^2 \rangle$, $\mathcal{M}^2$ is inserted for $\mathcal{M}$ in equation (13).

$$\langle \mathcal{M}^2 \rangle (\beta) = \frac{\sum_{i=1}^{16} \mathcal{M}_i^2 e^{-\beta E_i}}{2\left(e^{-\beta(8J)} + e^{\beta(8J)}\right) + 12} = \frac{2\left(16e^{\beta(8J)}\right) + 8\left(4 \cdot 1\right)}{2\left(e^{-\beta(8J)} + e^{\beta(8J)}\right) + 12}$$

$$\langle \mathcal{M}^2 \rangle (\beta) = \frac{16\left(e^{\beta(8J)} + 1\right)}{\left(e^{-\beta(8J)} + e^{\beta(8J)}\right) + 6}. \tag{18}$$

The magnetic susceptibility can then be calculated using equation (14).

$$\chi = \frac{1}{kT}\left(\frac{16\left(e^{\beta(8J)} + 1\right)}{\left(e^{-\beta(8J)} + e^{\beta(8J)}\right) + 6} - \left(\frac{4\left(e^{\beta(8J)} + 2\right)}{e^{-\beta(8J)} + e^{\beta(8J)} + 6}\right)^2\right) \tag{19}$$

These equations will later be useful for benchmarking our program by comparing computed values to the analytical expressions.

3.2.2 Phase transition

Phase transitions are sudden changes in properties relating to a change in external parameters such as temperature. The temperature at which such a transition takes place is called the critical temperature. The Ising model in two dimensions with no external magnetic field undergoes a second order phase transition [2]. This phase transition is characterized by spontaneous magnetization occurring below the critical temperature T_C and an average magnetization of zero above T_C [2].

The correlation length ξ is a function of temperature and it describes the degree with which two spins separated by some distance are correlated. When temperatures are much greater than the critical temperature, ξ is expected to be of the order of the lattice spacing. However, at temperatures close to the critical temperature, the correlation length for a second order phase transition will come to span the length of the entire system [3], meaning every spin is correlated with every other spin. The behaviour of ξ for T close to T_C is characterized by

$$\xi(T) \sim |T_C - T|^{-\nu}. \tag{20}$$

ξ, then for finite latices will be be proportional to the lattice size.

3.2.3 Extracting the critical temperature

Through finite size scaling relations [3], behaviours of finitely sized lattices can be extrapolated for infinite lattices. The critical temperature of a system scales as

$$T_C(L) - T_C(L_{\to\infty}) \propto aL^{-\frac{1}{\nu}} \tag{21}$$

[2], with a being a constant and ν being defined in equation (20). Alternatively the proportionality constant can be absorbed into a, giving the equation

$$T_C(L) - T_C(L_{\to\infty}) = aL^{-\frac{1}{\nu}}.$$

Mean magnetization, heat capacity and magnetic susceptibility are related to the critical temperature by the relations

$$\langle \mathcal{M}(T) \rangle \sim (T - T_C)^{\beta} \tag{22}$$

$$C_V(T) \sim |T_C - T|^{-\alpha} \tag{23}$$

$$\xi(T) \sim |T_C - T|^{-\gamma} \tag{24}$$

with $\beta = \frac{1}{8}$, $\alpha = 0$ and $\gamma = -\frac{7}{4}$ [2]

3.3 The Metropolis algorithm

We seek to model a system moving from an initial state to an equilibrium state. A Markov process used in this application needs to fulfil the conditions of ergodicity and detailed balance. A popular algorithm for applying Markov chains with these conditions is the Metropolis algorithm [2]. Firstly, assigning a random walk with a normalized transition probability, the stochastic matrix $\hat{W}$ with matrix elements ij giving the probability of moving from state i to state j. This will act on the normalized state $\hat{\omega}(t)$, preserving normalization, giving the next state $\hat{\omega}(t + \epsilon)$.

The Markov chain employs the repeated application of a stationary, stochastic transition matrix $\hat{W}$ to a starting state $\hat{\omega}(t = 0)$ with the sum of every entry being normalized to 1 and every entry $\omega_i \in \mathcal{R}|\{0 \leq \omega_i \leq 1\}$ such that

$$\hat{\omega}(t = n\Delta t) = \hat{W}^n \hat{\omega}(t = 0), \quad \Delta t \textit{ is the time step.} \tag{25}$$

The largest eigenvalue of such a matrix $\hat{W}$ being 1 [2] for an eigenstate

$$\hat{\omega}(t) = \hat{W}\hat{\omega}(t) \tag{26}$$

and the lowest possible eigenvalue being greater than or equal to 0, meaning the repeated application of the matrix $\hat{W}$ on the state $\hat{\omega}(t = 0)$ will always converge to a steady state.

The transition probability is not known for the system we seek to model and so it is modelled as the product of the transition probability $T \in [0, 1]$ and the probability of the transition being accepted $A \in [0, 1]$:

$$W(j \to i) = \underbrace{T(j \to i)}_{P(transition)} \cdot \underbrace{A(j \to i)}_{P(acceptance)} \tag{27}$$

A move from state j to state i with transition probability $T(j \to i)$ is suggested, the move is then accepted or rejected according to the acceptance probability $A(j \to i)$. The correct choice of A and T is that which after infinite such iteration, starting from any distribution gives the Boltzmann distribution.
The probability distribution then can be expressed as

$$\omega_i(t+1) = \sum_j \left(\omega_j(t)T(j \to i)A(j \to i) + \omega_i(t)T(i \to j)A(i \to j)\right). \tag{28}$$

We now need to impose the condition of detailed balance into our model, this condition takes the form of an equation:

$$W(j \to i)\omega_j = W(i \to j)\omega_i. \tag{29}$$

courtesy of Lars Onsager [4]. Rearranging this equation gives

$$\frac{W(j \to i)}{W(i \to j)} = \frac{\omega_i}{\omega_j}$$

then substituting for the transition probabilities,

$$\frac{T_{j\to i}A_{j\to i}}{T_{i\to j}A_{i\to j}} = \frac{\omega_i}{\omega_j}$$

then introducing the Boltzmann distribution

$$\omega_i = \frac{e^{-\beta(E_i)}}{Z}. \tag{30}$$

For large lattices, calculating the partition function Z becomes infeasible, but calculating based on the ratio of the two PDFs eliminates the partition function:

$$\frac{\omega_i}{\omega_j} = e^{-\beta(E_i - E_j)}.$$

Rearranging our equation from before

$$\frac{A_{j\to i}}{A_{i\to j}} = \frac{\omega_i T_{i\to j}}{\omega_j T_{j\to i}}$$

then assuming that the transition probability is symmetric, and therefore that $T_{i\to j} = T_{j\to i}$ gives at last, the equation

$$\frac{A_{j\to i}}{A_{i\to j}} = \frac{\omega_i}{\omega_j} = e^{-\beta(E_i - E_j)}. \tag{31}$$

This relation only tells us the ratio of the acceptance probabilities. To find the acceptance probability of a transition, we have to consider energy.
If the change in energy from state j to state i, $E_i - E_j = \Delta E < 0$ then the largest acceptance value is $A(j \to i)$. Then there is only to choose $A(j \to i)$ according to the equation

$$A(j \to i) = \min\left(1, \frac{w_i}{w_j}\right) \tag{32}$$

which fixes the value of $A(i \to j)$ by the relation in equation (31). This means that any transition resulting in the lowering of the energy is accepted as a move. If however, the energy increases under the suggested move, a random number $R \in [0, 1]$ with a uniform PDF is chosen and the suggested move $j \to i$ accepted if

$$R \leq \frac{w_i}{w_j} = e^{-\beta \Delta E}. \tag{33}$$

This suggestion method does not violate ergodicity as every end state could be reached by the Markov process given any beginning state given an infinite runtime. That is to say, there are no fixed points, only stable states.
After sufficiently many iterations, or strictly, the limit $t \to \infty$ the algorithm is required to produce a steady state in which

$$\frac{d\omega_i(t)}{dt} = \sum_j [W(j \to i)\omega_j - W(i \to j)\omega_i] = 0 \tag{34}$$

the former part of that equation being the so-called Master equation derived from the so-called Chapman-Einstein-Enskog-Kolmogorov equation [2]. Requiring this equality in the limit $t \to \infty$ gives the equation

$$\sum_j \omega_j T_{j \to i} A_{j \to i} = \sum_j \omega_i T_{i \to j} A_{i \to j} \tag{35}$$

indicating the steady state.

3.4 Periodic boundary conditions

Because it is not computationally possible to simulate a system in the magnitude of real matter (which is close to Avogadro's number), a simulated system has a much higher surface-per-volume ratio [1] than a real system. This needs to be taken into consideration, because otherwise, simulation results would ignore a large part of interactions existing in larger systems. One way to tackle this issue, is to employ periodic boundary conditions. The idea is that the particles on the system's borders interact with each other. That is, in a lattice of size $n \times n$, the particles in the nth row or column interact with the particles in the first row or column, as if they were in the $(n+1)^{\text{th}}$ row or column[2]. The advantage of this is that there are no surface particles, thus simulating an infinitely large system, resembling reality closer than a very small system. The disadvantage is that particles are correlated in an unnatural way. Especially in smaller systems, this will lead to particles interacting with their own "mirror image", where a particle's spin conformation might affect the surrounding in a way that the particle itself will be affected by its own spin. All simulations were done using periodic boundary conditions.

3.5 Computational implementation

3.5.1 Units

In this simulation, we simply assume the individual spins to be unitless. Thus, the magnetisation is unitless as well. The energy is scaled as the unitless $\tilde{E} =$

[1] In the case of a two-dimensional system, it is technically a border-per-surface ratio.

[2] Technically, by correlating spins in this way, we are simulating spins interacting on the surface of a torus

E/J, the Temperature is scaled as $\tilde{T} = \frac{k}{J}T$ and also unitless. This ensures that the exponent is unitless. Then, the Heat Capacity C_V, defined as $\frac{\sigma_E^2}{kT^2}$, is given as $\frac{J^2\sigma_{\tilde{E}}^2}{k\frac{J^2}{K^2}\tilde{T}^2} = \frac{\sigma_{\tilde{E}}^2}{\tilde{T}^2}k$, that is, it is reported as a multiple of k. Similarly, The susceptibility χ, defined as $\frac{\sigma_{|M|}^2}{kT}$, has units $\frac{1}{J}$. For simplicity, tilde-symbols are dropped in later discussions, and when stating the temperature in the text, its scaling factor is omitted.

3.5.2 Analytical solution

The implementation of the analytical solution is rather straightforward. For each spin, a for-loop is implemented (within a previous for-loop), that loops over each permitted state (1 and -1). That system's magnetization and energy are then calculated using the formulas stated above and stored in an array. From the energies, the probabilities and the partition function can be calculated for a given temperature. One can than calculate $E[E], E[E^2], E[M]$, and $E[M^2]$ and thus find the system's susceptibility and heat capacity. However, this scales as $\mathcal{O}(2^N)$ because there are 2^N microstates where N is the amount of spins, this becomes a very expensive calculation even for latices where e.g. the lattice size L=5 (N=25), and is clearly not doable for large lattices. Here, only the case L=2 was implemented.

3.5.3 Efficient formulas

The system's total energy upon switching one spin state is changed by any of $\Delta E \in \{-8J, -4J, 0, 4J, 8J\}$ because

$$\Delta E = E_2 - E_1 = J \sum_{<kl>}^{N} s_k^1 s_l^1 - J \sum_{<kl>}^{N} s_k^2 s_l^2$$

Because only one individual spin s_l is changed, that can be rewritten as

$$\Delta E = -J \sum_{<kl>}^{N} s_k (s_l^2 - s_l^1)$$

Then, because $s_l^1 - s_l^2$ is either 2 or -2, depending on the value of s_l^1, the final change in energy can be written as

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

As the change in magnetisation only depends on the particle s_l, it is either -2 (1 to -1) or 2 (-1 to 1), so the change in the magnetisation is given as

$$\Delta M = -2s_l^1 = 2s_l^2$$

The relative difference in probability between two spin states, which is the actual selection rule for the Metropolis algorithm, is then given by

$$\frac{exp(-E_2/T)}{exp(-E_1/T)} = exp(-\Delta E/T)$$

This can be precomputed for each individual ΔE for a given temperature and stored in an array to be reused, which saves computational time.

3.5.4 Simulating the system

The lattice was implemented as a two-dimensional integer matrix with values -1 or 1, representing the spin states. The initial matrix is either randomized or has all spins pointing in the same direction (see Results-part). Then, the initial energy and magnetisation can be calculated using the formulas above. The Metropolis algorithm was then implemented the following way after the system has been set up:
N times, where $N = L^2$ and L is the lattice size, a suggested spin to flip is suggested. Then the change in energy and $exp(-\Delta E/T)$ are calculated.If $exp(-\Delta E/T)$ is larger than a randomly distributed number R [3], the move is accepted, otherwise the loop over N is continued. If the move is accepted, the system's magnetisation and the energy are updated. After the loop is finished, the expectation values for E, E^2, M, M^2 are updated. The loop is then repeated for the desired amount of simulation steps. Finally, the final expectation values for E, E^2, M, M^2, C_V and χ are calculated. To reiterate, for each Monte Carlo cycle, the energy- and magnetization- difference is added to the total energy and magnetization:

$$E \mathrel{+}= \Delta E, \qquad M \mathrel{+}= \Delta M.$$

If the move was not expected, the current values were added to the total. Then E^2 and M^2 are calculated

$$E^2 \mathrel{+}= E \cdot E, \qquad M^2 \mathrel{+}= M \cdot M.$$

Then, after the final Monte Carlo cycle has been completed, the final tally is divided by the number of Monte Carlo cycles

$$\langle E \rangle = E/\#MC, \quad \langle E^2 \rangle = E^2/\#MC, \quad \langle M \rangle = M/\#MC, \quad \langle M^2 \rangle = M^2/\#MC.$$

In order to do this for different lattice sizes L and different temperatures, this whole procedure was implemented in two more loops, one looping over the temperatures and one looping over the lattice sizes.

3.5.5 Parallelization

In order to increase computational performance, the time-consuming calculations were parallelized using openMPI. The computational implementation has 3 nested loops: One loop iterating over each temperature, one loop iterating over the different lattice sizes, and one loop implementing the actual Metropolis algorithm[4]. Because the Metropolis algorithm explicitly depends on the earlier state, and because simulating smaller lattice sizes requires less time, the temperature loop was parallelized. This was done by giving each thread a different start temperature (with a spacing of the given temperature step between each thread), and giving each thread a "new" temperature step, which is the original temperature step times the amount of threads. The advantage of this implementation is that it is fairly simple to implement, and from the results, it is easy to inspect that it was implemented correctly. The disadvantage is that it

[3] The Mersenne-Twister RnG is used to create this number

[4] Technically, two more nested loops were implemented within the Metropolis algorithm, but they could equally well be implemented as a part Metropolis loop itself.

does not give ideal run time if the total amount of time steps is not divisible by the amount of threads, leading to one or several threads finishing earlier than the rest.
In order to improve the run time even more, we also tested several compiler flags, both for the parallelized and the non-parallelized version, in order to see how to achieve the best possible time.

3.5.6 Unit testing

We tested that the system was sat up correctly and that the resulting numbers and calculations are correct by comparing the simulation for a 2×2-lattice with the analytical solution at different temperatures. The requirement is that, after 10^6 simulations, the standard error compared to the analytical solution is required to be less than than 1%.

3.5.7 Calculating the critical temperature

The critical temperature can, as described earlier, be extracted using the formula

$$T_C(L) - T_C(L_{\to\infty}) = a \cdot L^{-1/\nu}$$

where L is the lattice size and $T_C(L)$ is the critical temperature in the non-infinite system. Here, $\nu = 1$. This can be reformulated as

$$T_C(L) = a \cdot L^{-1} + T_C(L_{\to\infty})$$

As $T_C(L)$ will be known from the simulations, and L is a preset factor, there are only two unknown constants: a and $T_C(L_{\to\infty})$. Those can then be found using linear regression, and the insecurity in the critical temperature as $L_{\to\infty}$ can be estimated to be standard deviation, which for linear regression lines is defined as

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{N-2}} \tag{36}$$

as is shown in [1], where $\hat{y}_i$ is the regressed value and y_i the measured value, and N is the size of the sample set. This will only serve as a rough estimate of both the critical temperature and the error, as y_i itself is not the true value.

4 Results

4.1 Reaching the equilibrium state

When setting up the initial lattice, we tested whether it is better to start with a random distribution of spins, or start with all spins pointing in the same direction. A simple way to do that is to visualize the system's average energy, average magnetisation and percentage of accepted configurations for a given amount of steps, as a function of the total amount of steps. We did that for two temperatures, T=1.0 and T=2.4, for lattices sized 20×20 and 100×100, and plotted the results. This can be seen in figure 1 and figure 2.

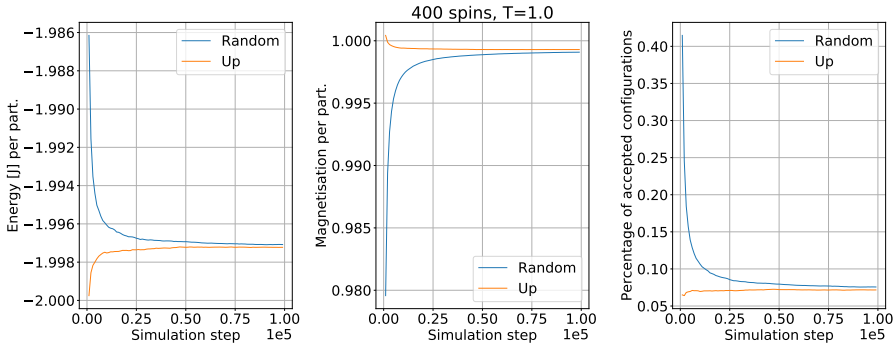

Figure 1: The system's average energy, its absolute magnetisation and the percentage of accepted spins as a function of the simulation step for a 20×20 lattice at temperature T=1.0. Up is all spins pointing up initially, random is randomized spin distribution.

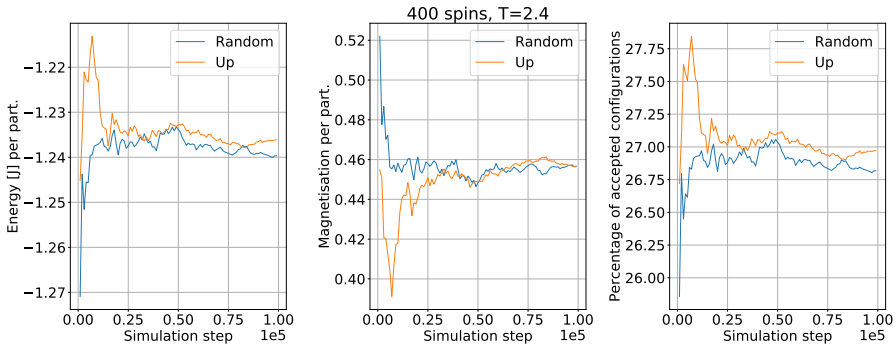

Figure 2: The system's average energy, its absolute magnetisation and the percentage of accepted spins as a function of the simulation step for a 20×20 lattice at temperature T=2.4.Up is all spins pointing up initially, random is randomized spin distribution.

One can clearly see that the magnetisation, the energy and the amount of accepted configurations behave similarly in the sense that they all tend to equilibrate after approximately the same amount of steps. The results produced using a 100×100 lattice gave similar results, indicating that these results can be used for any lattice size between 20 and 100. One can see clearly that the state with all spins pointing up is equilibrated after around 20,000 steps for both temperatures. At T=2.4, both the randomized system and the system with all spins pointing up seem to take the same amount of steps to equilibrate. However, for T=1.0, the randomized spin distribution seems to require more steps.[5]

For these reasons, the larger simulations were sat up using an initial systems

[5]This might however be an illusion, as the plots visualize the *total* averages. Very large discrepancies in the first steps might alter the averages long-term, even though the system already is in equilibrium.

where all spins point up, and the first 20,000 steps were discarded. After the equilibrium state has been reached, one can start to sample in order to calculate several physical units of the system. Another interesting observation is that many more states are accepted at the higher temperature - while only slightly more than 0.05% states are accepted at T=1.0, over 25% are accepted at T=2.4. This is expected - The exponent $\frac{-\Delta E}{T}$, that is, the sampling rule, for positive values of ΔE, is much larger for larger values of T, thus much more moves are accepted. This fits well with the discussion of the energy distribution described below - as moves are accepted, there is much more fluctuation among the different states, which leads to a broader distribution of energies.

4.2 Time benefit from Parallelization and compile flags

Parallelization gave a substantial time benefit, however, different compilation flags barely changed the program's run time. The program was run on a computer with a CPU with 4 cores, and 8GB RAM. Table 2 shows the time usage (single runs) achieved with different compilation flags, with- and without parallelization. A rather short timing experiment was conducted, and it is run over 4 temperatures in order to achieve the ideal speedup from parallelization.

Table 2: Time consumption of a simulation using $4 \cdot 10^4$ simulation steps for 4 different temperatures (T=2.0,2.1,2.2,2.3) and 4 matrix sizes (L=40,60,80,100) with different compiler flags. All results were averaged over 3 runs.

Compiler flag	Time (parallel) [s]	Time (non-parallel) [s]
None	153	507
-O2	149	507
-O3	150	496
-Ofast	137	503

The big improvement in timing clearly comes from parallelization: The time usage is more than 3 times higher for the non-parallelized code. However, it is not 4 times higher for any of the chosen compiler flags - this might be due to imperfect implementation of parallelized code, among them the fact that the program, in the end, only runs with one active thread that is not finished yet. The compiler flags have a very minute effect on the run time of the non-parallelized code, speeding it up by less than 3 percent, while only the -Ofast-flag leads to appreciable improvement of ca. 10 percent in the run time of the parallelized code. However, these numbers fluctuated wildly, as can be seen in the file "time.csv" on GitHub. Thus, it is quite possible that there really is no speed-up on the long run. The larger simulations were run using the -Ofast-flag nevertheless.

4.3 Results achieved using 2×2 lattice

Using a 2×2 lattice, the same results were achieved as for the analytical solution, which indicates that the algorithm was implemented correctly and gives reasonable results. The actual results are not interesting and not presented here. Because these simulations are comparatively fast to run, it is feasible to compute the relative error found in these calculations and use them to get a

very general idea on error estimation for the calculations where no analytical solution can be computed. Using 10^6 simulation steps, the relative error could be reduced to less than one percent for all physical quantities for temperatures between 2.0 and 3.0. However, there were huge fluctuations in the maximum relative error, and the maximum relative error was always found in the heat capacity. Example data can be found in table 3 and 4 in the appendix.

4.4 Energy distribution in a 20×20 lattice

As the temperature is increased, the system's average energy increases as well. The energies' probability distribution is changed as well. These probability distributions for a T=1.0 and T=2.4 in a 20×20 lattice are shown in figure 3.

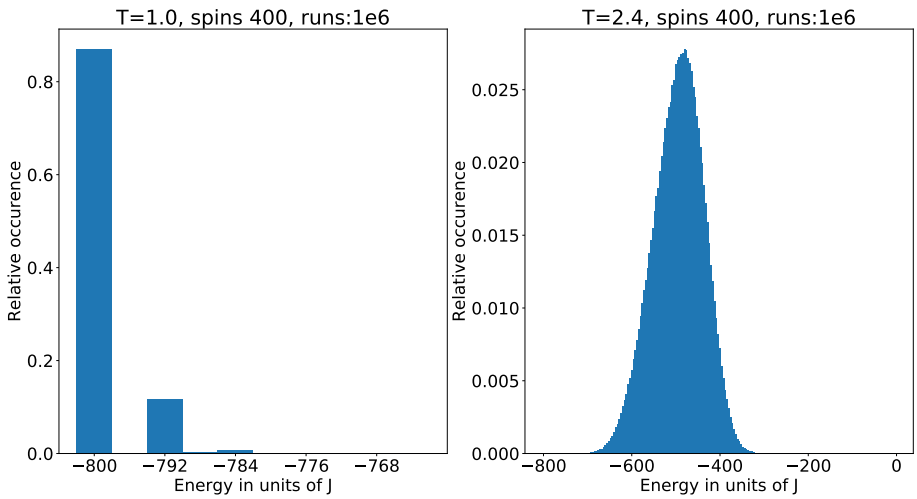

Figure 3: Distribution of energies in a 20×20 lattice at T=1.0 and T=2.4 with 10^6 simulation steps.

This data fits well with the calculated standard deviations. At T=1.0, the standard deviation in energy is $\sigma_E \approx 3.05J$ around the average value $E = -798.8J$, at T=2.4, the standard deviation in energy is $\sigma_E \approx 57.1J$ around the average value $E = -495.0J$. At the higher temperature, the energies are distributed over a much broader range, and far more individual energies have a nonzero distribution. This data fits also very well with Figure 1 and Figure 2 - far more suggested configurations are accepted at T=2.4, which clearly increases the fluctuation in energy and thus the standard deviation, compared to the low-temperature state. From this, we can also see that the system at T=2.4 has a much higher heat capacity, which is closely related to the standard deviation. This fits with the calculated data, where the individual particle's heat capacity is given as C_V=1.4k for T=2.4 and C_V=0.023k for T=1.0.

4.5 Behaviour of the system close to the critical temperature

Thanks to calculations done by Lars Onsager [5], we know that the Ising model has an analytical critical temperature of $T_C \approx 2.269$, where a phase transition should occur. This behaviour is also expected to show up in a simulation, and a large enough lattice with enough simulation steps should give a result close to the analytical solution.
We ran a simulation of lattices of sizes L $\forall L \in \{40, 60, 80, 100\}$ for temperatures ranging from 2.0 to 2.35 with a step length of $\Delta T = 0.05$ in order to get an overview over the systems' behaviour in the vicinity of the critical temperature. These simulations were run with $10^6 - 20,000$ simulation steps. The results are shown in figure 4. Note that the susceptibility χ is calculated from the absolute magnetization.

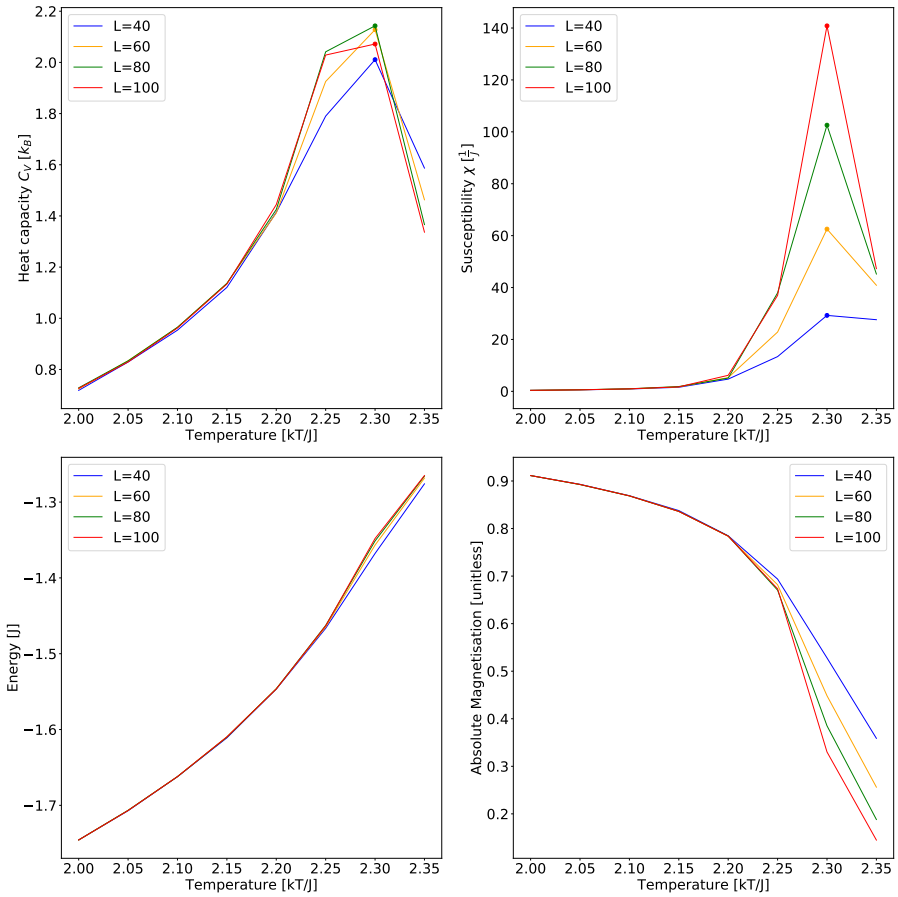

Figure 4: C_V, χ, E and $|M|$ as functions of T for L $\in \{40, 60, 80, 100\}$ for T between 2.0 and 2.35 with a step length of $\Delta T = 0.05$. For each temperature, $10^6 - 20,000$ simulation steps were carried out. Maximum value is marked for χ, C_V.

From these graphs, a whole lot of information can be extracted. All four systems behave almost identically for temperatures below 2.2 (the curves overlap), however, at temperatures close to the analytical critical temperature, the curves separate. This is in accordance with formula (21), which states that the divergence from the real critical temperature is a function of lattice size. This is especially interesting to observe for the absolute magnetisation, where the value at T=2.35 is getting smaller and smaller as the lattice size increases.
Another observation is that the simulated critical temperature gets closer and closer to the real critical temperature as the lattice size increases, which also is in accordance with formula (21). This can be observed by the "steepness" in the curves of the heat capacity: For L=40, it is much higher at T=2.30 than at T=2.25, indicating that it is much closer to 2.30, while at L=100, the values for T=2.30 and T=2.25 are almost identical. The steep drop from T=2.30 to T=2.35 indicates that it is closer to T=2.25 than to T=2.35 for all four systems. Finally, one can observe how the maximum values for χ and C_V increase with the systems size, indicating that they reach infinity as L approaches infinity.

In order to get a better estimate on the simulated systems' critical temperature, another simulation was carried out, with a more narrow range for the temperatures, and more simulation steps. A simulation with T between 2.20 and 2.35 and $3.48 \cdot 10^6$ simulations for 8 temperatures with equal spacing is shown in figure 6 in the appendix. From there, one can see that the critical temperature for both the susceptibility and the heat capacity lies between T=2.265 and T=2.315 (but for χ when L=40). We ran a last simulation consisting of only this temperature interval, with $5.98 \cdot 10^6$ simulation steps. The results can be found in figure 5.

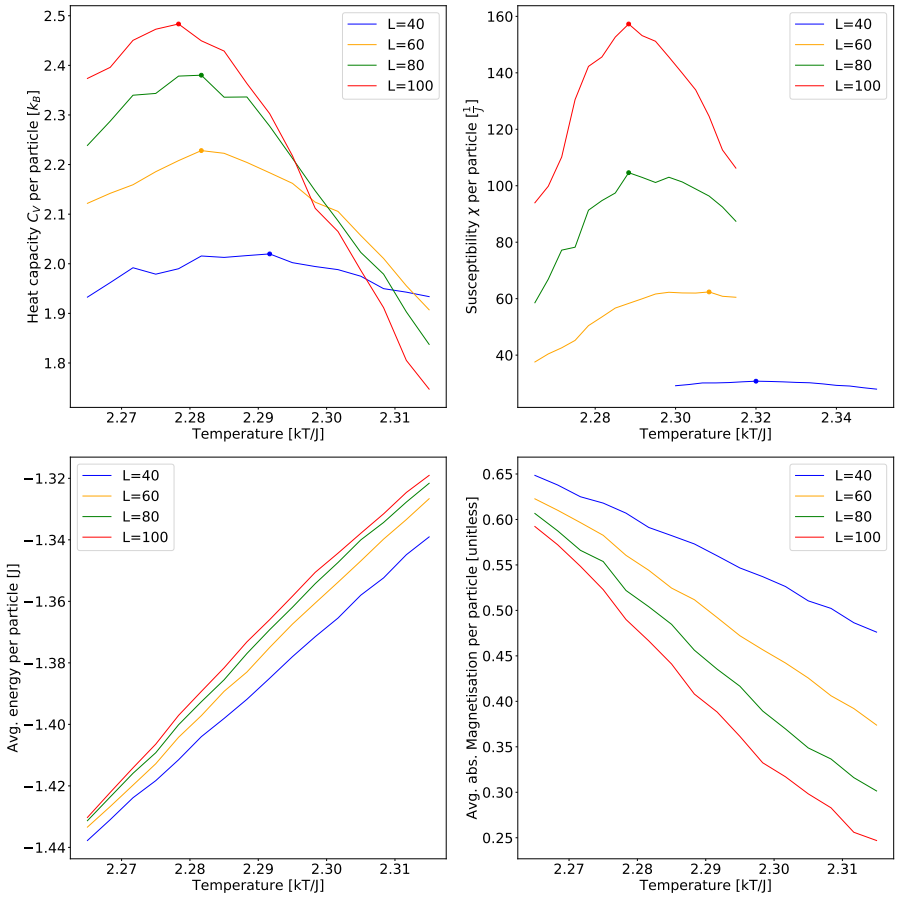

Figure 5: C_V, χ, E and $|M|$ as functions of T for L $\in \{40, 60, 80, 100\}$ for T between 2.2615 and 2.315 (2.30 and 2.35 for the susceptibility at L=40) with a step length of $\Delta T = 1/300$. For each temperature, $5.98 \cdot 10^6$ simulation steps were carried out. Maximum value is marked for χ, C_V.

This data fits very well with our expectations: The critical temperatures approach the analytical value for larger values of L. The data is not perfect, as the maximum heat capacity is identical for L=60 and L=80, and the susceptibility is identical for L=80 and L=100, something that might not have happened when running yet more simulation steps with even smaller step lengths.

4.6 Critical temperature

The critical temperature can be calculated using the procedure described in section 3.2.3. Doing this, and using the maximum values from figure 5, which can be found in table 5 in the appendix, we find that $T_C(L_{\to\infty}) \approx 2.269 \pm 0.002$ using data from the heat capacity and $T_C(L_{\to\infty}) \approx 2.265 \pm 0.005$ using data from the susceptibility. Both results are very close to the analytical result, and the analytical result is within one standard deviation both times. Clearly, the obtained values for $T_C(L)$ are only approximations, as it may well lie at

temperatures that were not simulated, and only testing all temperatures with infinite Monte Carlo cycles would yield the exact results for these systems. That is why the results are not perfectly identical to the analytical result.

5 Conclusion

We found that using simulations, one can learn a lot about physical systems. The Metropolis algorithm has shown to be very reliable, and the simulated result for both the critical temperature and the energy distribution match with the expectations from the analytical expressions. When simulating several lattice sizes, the larger the lattice is, the more it resembles bulk matter, and it is possible to extrapolate to the behaviour at infinite size. This extrapolation is clearly necessary, as these simulations are very time-intensive, and simulating for example L=1000 with the same amount of Monte Carlo cycles would have taken weeks. Our result matched very well with the analytical result for the critical temperature. The Ising model is quite primitive in the sense that several quantities showing up in real systems are ignored, such as length or charge, or that particles only interact with their nearest neighbours, or that matter in fact is three-dimensional. But it still has many important physical attributes that one can calculate, out of which the fact that it employs phase transitions is most important. The loss of magnetisation in ferromagnetic matter above a critical temperature has been observed. In more sensitive simulations, the lack of the model's complexity is something to be taken into consideration, but it exceeds the scope of this article. Using gradient-based techniques, it might be possible to approximate a simulated system's critical temperature without having to do simulations for each temperature where one might suspect the critical temperature to be. We did not employ methods such as reshuffling (Bootstrap, for example) to get better guesses on the critical temperature, and we did not include aspects such as covariance that might be caused by, for example, the random number generator. This could be done in future articles.

6 Critique

Not being physics students, applying the principles of statistical physics was quite demanding and at times frustrating since the concepts were so foreign. An alternative approach to the application of the Metropolis algorithm or a more thorough description behind the principles that were taken for granted would for us be a more helpful approach.

7 Appendix

7.1 Figures

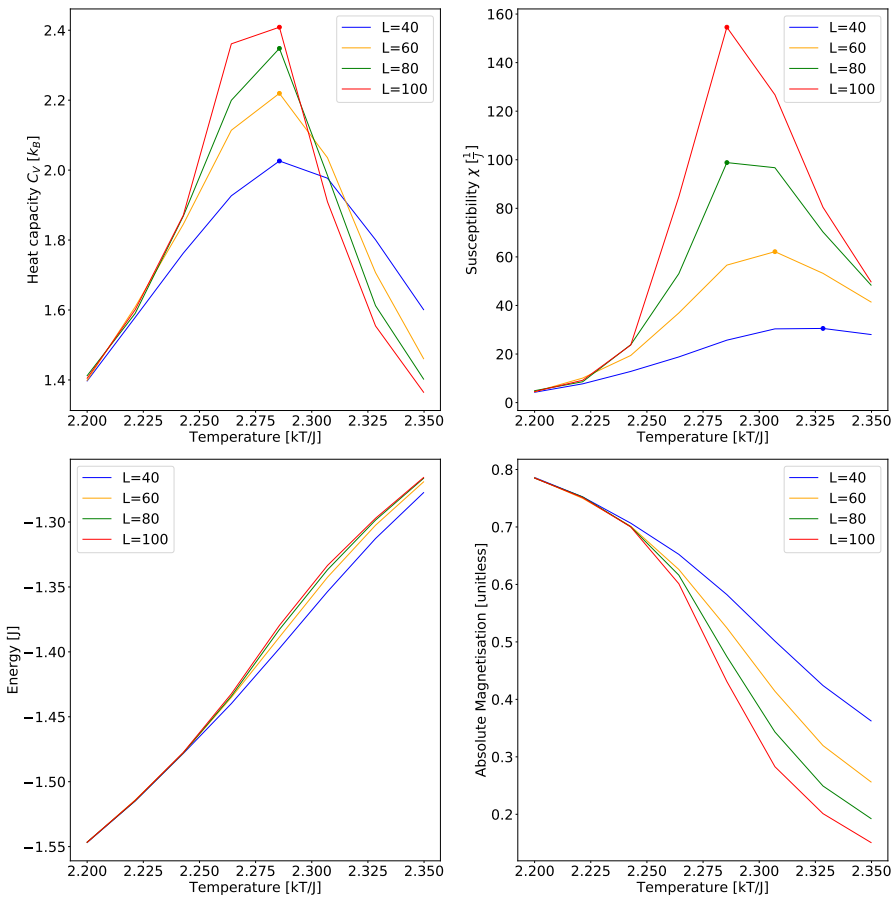

Figure 6: C_V, χ, E and $|M|$ as functions of T for L $\in \{40, 60, 80, 100\}$ for T between 2.20 and 2.35 with a step length of $\Delta T = 0.0214$. For each temperature, $3.48 \cdot 10^6$ simulation steps were carried out. Maximum value is marked for χ, C_V.

7.2 Tables

Table 3: The largest relative error, at which temperature showed up, and where it showed up for 10^7 simulation steps in a 2×2 lattice. Temperatures were run in $[2.0, 3.0)$ with a uniform spacing of 0.1.

Relative error	Temperature	Physical quantity
0.00169	2.0	Heat Capacity
0.00286	2.1	Heat Capacity
0.00180	2.1	Heat Capacity
0.00156	2.0	Heat Capacity
0.000819	2.6	Heat Capacity

Table 4: The largest relative error, at which temperature showed up, and where it showed up for 10^6 simulation steps in a 2×2 lattice. Temperatures were run in $[2.0, 3.0)$ with a uniform spacing of 0.1.

Relative error	Temperature	Physical quantity
0.00598	2.1	Heat Capacity
0.00356	2.0	Heat Capacity
0.00882	2.0	Heat Capacity
0.00823	2.1	Heat Capacity
0.00282	2.9	Heat Capacity

Table 5: Critical temperatures at different lattice sizes L for both the susceptibility χ and the heat capacity C_V. The critical temperature is the temperature where C_V/χ reached their maximum values.

Lattice size L	Critical Temperature (from C_V)	Critical Temperature (from χ)
40	2.2917	2.3200
60	2.2817	2.3083
80	2.2817	2.2883
100	2.2783	2.2883

7.3 List of programs

All programs can be found on `https://github.com/adrian2208/FYS3150_collab` in the folder "Project4".

1. b.cpp - Program with several options of writing to file, different matrices etc., for a given temperature and lattice size
2. e_not_parallel.cpp - Like b, but can run over several lattices and temperatures, more efficient
3. e.cpp - Same as above, but parallelized with openMPI
4. vecop.hpp - Several functions

5. vecop.cpp - Functions, out of which some are used.
6. tests_main.cpp - Unit testing.
7. a.py - Analytical solution. Not properly up to date!
8. plot_cv.py - Creates 2×2 plot for the larger simulations, also calculates $T_C(L_{\to\infty})$
9. plot_distribution.py Plots the distribution histograms at different temperatures.
10. plot_energy_simulation_step.py Plots the avg. energy, the avg. magnetisation and the amount of accepted steps as a function of simulation step.
11. functions.hpp & functions.cpp - Contains functions for parallelization, and all the integrands, and some more.

8 References

[1] Jay L Devore and Kenneth N Berk. *Modern mathematical statistics with applications*. Springer, 2012.

[2] Morten Hjort-Jensen. Computational physics: Lecture notes fall 2015.

[3] Morten Hjort-Jensen. Problem set 4 fall 2019.

[4] Lars Onsager. Reciprocal relations in irreversible processes. i. *Physical review*, 37(4):405, 1931.

[5] Lars Onsager. A two-dimensional model with an order–disorder transition (crystal statistics i). *Physical review*, 65:117–49, 1944.