

Computational analysis of the Ising Model

Blind Grading Number: 6967S

Abstract

A computational Monte Carlo study of the Ising model of a ferromagnetic 2D square lattice is conducted, by generating an ensemble of equilibrium configurations through a vectorised version of the Metropolis algorithm.

First, an analysis of the relaxation and decorrelation times was performed, especially in proximity of the critical temperature. Then, the temperature dependence of the main lattice properties were tested: magnetisation, energy, specific heat and magnetic susceptibility and their critical exponents reproduce the theoretical expectations. In particular, a critical temperature $T_{crit} = 2.269 \pm 0.004$ was estimated, within one sigma of the true value.

Finally, the phenomenon of hysteresis was measured for $T < T_{crit}$.

The analysis was later repeated for higher dimensions ($d = 3, 4, 5$), and our critical exponents were found to be in agreement with mean field theory for $d = 4, 5$.

1 Introduction

The Ising model was first invented to study ferromagnetic metals, represented as lattices of spins that can take values $+1$ and -1 and evolve under nearest-neighbour interactions.

It is one of the simplest examples of systems exhibiting a phase transition: indeed, as the temperature crosses a critical value, the lattice configuration transforms from an ordered state with all spins aligned, to a disordered phase with spins randomly oriented.

Also due to its simplicity, the Ising model finds wide applications to the study of collective phenomena and phase transitions for systems other than ferromagnetic crystals: notable examples include surface wetting[1] and opinion diffusion models in the social sciences[2].

The partition function for its 1D version was calculated in 1924 by Ernst Ising and the 2D problem for a square lattice was solved by Lars Onsager in 1944 [3].

A general ferromagnetic lattice, characterised by an energy coupling $J > 0$ and magnetic moment per spin μ , under the effect of an external magnetic field H , is described by the following Hamiltonian:

$$\mathcal{H} = -J \sum_{\langle ij \rangle} s_i s_j - H \mu \sum_i s_i \quad (1)$$

where $s_i \in \{-1, 1\}$ denotes the spin occupying the i -th site and $\langle ij \rangle$ denotes a sum over nearest neighbours. The Hamiltonian above does not provide a dynamical description of the system, but gives rise to an equilibrium statistic: a lattice in contact with a heat bath at temperature T will be observed in a microstate Γ of energy E_Γ with probability

$$p_\Gamma = \frac{\exp(-\frac{E_\Gamma}{k_B T})}{Z} \quad (2)$$

where Z is the partition function. The unfeasibility of direct calculations of observable expectations in big lattices is evident, since it becomes necessary to consider a number of spin configurations growing as $2^{\mathcal{N}}$ with the number of sites \mathcal{N} . Simulations can avoid this issue by effectively exploring a sequence of microstates where each configuration Γ appears with the probability p_Γ .

In this report the Ising model of a 2D square lattice is simulated with a Monte Carlo technique implementing the Metropolis algorithm. The next section provides the necessary theoretical and computational background for the problem, followed by section 3, where an outline of the code implementation is given. Section 4 presents the results obtained, and section 5 draws the conclusions of the experiment.

2 Background and analysis

In this report the following units convention is adopted:

- $J, k_B = 1$
- $H\mu$ is measured in units of J (and from now simply denoted by H)
- The temperature T is measured in units of J/k_B

2.1 Performing measurements

Our lattice consists of a set of \mathcal{N} spins $\{s_i\}_{i=0}^{\mathcal{N}}$. The magnetisation, used as an order parameter, and total energy per unit spin are defined as:

$$M := \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} s_i \quad (3)$$

$$E := -\frac{1}{\mathcal{N}} \sum_{\langle ij \rangle} s_i s_j - \frac{1}{\mathcal{N}} H \sum_i s_i \quad (4)$$

The specific heat, defined as $C := \frac{dE}{dT}$, can be obtained from the variance of the energy when the system has reached equilibrium [6]:

$$C = \mathcal{N} \frac{Var(E)}{T^2} \quad (5)$$

A similar relation holds between the magnetic susceptibility $\chi := \frac{dM}{dH}$ and the variance of the magnetisation:

$$\chi = \mathcal{N} \frac{Var(M)}{T} \quad (6)$$

2.2 Onsager's results

Here we state some exact results for an infinite square lattice from Onsager's 1944 paper [3].

When $H = 0$, the Hamiltonian is symmetric under spin reversal. At low temperatures, the system breaks this symmetry by relaxing to a minimum-free energy state with finite magnetisation, with magnitude:

$$M = \begin{cases} (1 - \sinh^{-4}(\frac{2}{T}))^{\frac{1}{8}} & \text{if } T \leq T_{crit} \\ 0 & \text{if } T > T_{crit} \end{cases} \quad (7)$$

Hence, the lattice undergoes a second order phase transition at a critical temperature:

$$T_{crit} = \frac{2}{\ln(1 + \sqrt{2})} \quad (8)$$

Expressions for the mean energy and the specific heat are significantly more complicated and involve calculations of multiple elliptic integrals. Nonetheless, an asymptotic expression for the specific heat when $T \rightarrow T_{crit}$ highlights its logarithmic singularity [4]:

$$C = -\frac{8}{\pi} \left(\frac{1}{T_{crit}} \right)^2 \ln \left| 1 - \frac{T}{T_{crit}} \right| + \text{const} \quad (9)$$

2.3 Critical exponents

When $T \rightarrow T_{crit}$, several quantities of interest display a power law singularity characterised by so-called *critical exponents*. Below the specific naming convention is presented for the exponents of heat capacity, magnetisation and magnetic susceptibility:

$$C, M, \chi \propto (T - T_{crit})^{-\alpha, \beta, -\gamma} \quad (10)$$

In this report we will investigate an additional critical exponent, relating the magnetisation to the external field on the *critical isotherm*, i.e. when $T = T_{crit}$:

$$M \propto H^{1/\delta} \quad (11)$$

Theoretical predictions give for the 2D Model: $\alpha = 0$, $\beta = 1/8$, $\gamma = 7/4$ [3], $\delta = 15$ [5]. While exact calculations for higher dimensions d are not generally possible, mean field theory provides an exceptional approximation when $d \geq 4$, yielding: $\alpha = 0$, $\beta = 1/2$, $\gamma = 1$, $\delta = 3$ [6].

Finite lattices do not display true phase transition since all sensible observables are analytic functions of the temperature [3]. Therefore, critical behaviour very close to T_{crit} is smoothed out and eventually deviates from the power law singularities [10]. When fitting the critical exponents, the range of temperatures considered should be far enough from T_{crit} to reflect this.

2.4 Computing averages

Computing averages is an essential task to perform measurements and estimate their errors, as in the case of the specific heat and susceptibility. Given an observable \mathcal{O} , to compute its average value

$$\langle \mathcal{O} \rangle = \sum_{i=1}^{N_{samples}} \mathcal{O}_i$$

we often need to satisfy two conditions:

- The system needs to be at equilibrium.
- The samples used to compute the average must be independent.

2.4.1 Equilibrium

The 2nd law of thermodynamics guarantees that a sufficiently big system in a thermal bath will eventually relax to its equilibrium state, where its macroscopic properties become constant to a good approximation. In our case, the stabilisation over time of the order parameter M to a constant value, is the best predictor for equilibrium[7]. In the 2D Ising model, when $H = 0$, it is useful to distinguish two cases:

- $T < T_{crit}$, two equilibrium states with finite and opposite magnetisation are available, due to the symmetry $M \rightarrow -M$ in the Hamiltonian.
- $T > T_{crit}$, allows a unique equilibrium corresponding to $M = 0$.

2.4.2 Decorrelation time

As the lattice configuration evolves freely under random thermal spin flips, the magnetisation oscillates over time around its equilibrium value with amplitude $M'(t) = M(t) - \langle M \rangle$. The degree of independence between time-separated configurations can be quantified by defining the *autocovariance* of the order parameter M :

$$A(\tau) = \langle M'(t + \tau)M'(t) \rangle \quad (12)$$

and the corresponding *autocorrelation* $a(\tau) = \frac{A(\tau)}{A(0)}$. Clearly, $a(0) = 1$ and we expect $\lim_{\tau \rightarrow \infty} a(\tau) = 0$, since $A(\tau)$ is computed from the sum of equally probable positive and negative deviations. As a matter of fact, the autocorrelation displays an asymptotic exponential decay[8]:

$$a(t) = \exp(-t/\tau_e) \quad (13)$$

where τ_e is called decorrelation time.

2.4.3 Error correction

Choosing samples separated by a time $\Delta t \gg \tau_e$ in the same simulation provides a robust routine to extract uncorrelated lattice configurations and calculating averages. However, it is possible to use shorter time intervals Δt , by accounting for an increase in the error estimate [9]:

$$\sigma' = \sigma \sqrt{1 + \frac{2\tau_e}{\Delta t}} \quad (14)$$

where σ denotes the standard deviation of a set of measurements for a quantity \mathcal{O} .

It should be noted that for high values of Δt we recover the expected error $\sigma' \approx \sigma$.

2.4.4 Note on magnetisation averaging

One should be careful when taking averages of the magnetisation: in finite lattices, since the Hamiltonian is symmetric under the flip of all spins, the average magnetisation $\langle M \rangle$ is always 0 over sufficiently long times, even when $T < T_{crit}$, since thermal fluctuations can occasionally be strong enough to induce a magnetisation inversion. This is particularly evident when T approaches T_{crit} , since spin fluctuations develop long-range correlations [6] and cause M to flip regularly. This is never observed when $N = \infty$, because any fluctuation in M is suppressed by the infinite size of the lattice.

To mitigate this effect when we try to infer the behaviour of infinite Ising model from our study of finite lattices, we will therefore consider the absolute magnetisation $\langle |M| \rangle$ instead of the mean magnetisation $|\langle M \rangle|$, as it yields stable estimates even over long times.

2.5 Hysteresis

An important class of out-of-equilibrium phenomena explored here is *hysteresis*. Hysteresis occurs when the external field H is cycled in the range from $-H_{max}$ to H_{max} several times, when $T < T_{crit}$. If the field is strong enough, when H is brought to H_{max} , the lattice reaches *saturation*, i.e. $M = 1$ and as H returns back to 0, the magnetisation remains in a stable state with residual magnetisation $M_{res} > 0$.

When H decreases to a *critical field* value H_{crit} , the lattice suddenly transitions from its now metastable configuration to one with a reversed magnetisation.

No such phenomenon occurs at high temperatures, since thermal excitations are strong enough to relax the lattice to equilibrium. In particular, $M = 0$ when $H = 0$

3 Implementation

In this section we present the approaches taken for the main computational tasks. Detailed code listings can be found in the appendix [A](#).

3.1 System evolution

The Ising Hamiltonian [1](#) only describes the lattice at equilibrium, and provides a statistical rather than a dynamical description for a ferromagnet. Therefore, for our purposes, it is sufficient to simulate a system reproducing the correct state occupancies at equilibrium, where the probability of a microstate with energy to be occupied is proportional to its Boltzmann factor $\exp(-\frac{E}{k_B T})$.

3.1.1 Metropolis algorithm

The Metropolis algorithm is a simple strategy that achieves our goal by creating a Markov Chain, i.e. a memoryless process that transforms one state to the next, that stabilises ergodically on the equilibrium distribution of our Hamiltonian[\[7\]](#). The algorithm takes the following steps:

1. An initial configuration $\{s_i\}_{i=0}^N$ is chosen.
2. A random index j is chosen and the corresponding spin flip $s_j \rightarrow -s_j$ is considered.
3. The difference in energies ΔE between the final and initial state is computed using [3](#).
4. The ratio of Boltzmann factors of the final and initial state is then: $p = \exp(-\frac{\Delta E}{k_B T})$.
5. A random number z is sampled from the interval $[0, 1]$ and the spin s_j is flipped if $p > z$.
6. Steps 2 - 5 are repeated until needed.

3.1.2 The Checkerboard decomposition

The speed of the Metropolis algorithm can be increased by parallelising the update of multiple spins at a time. For a square lattice of size $N \times N$, the most widely used approach is the Checkerboard algorithm [\[7\]](#). This is implemented by saving the state of the lattice in a *Numpy* array, which in our task ensures the internal vectorisation of array operations. The main steps are the following:

1. The lattice is decomposed in two sublattices B and W , corresponding to the black and white of a checkerboard. By denoting a generic site of the lattice (x, y) , we define:

$$B = \{(x, y) \mid x + y \text{ even}\}, \quad W = \{(x, y) \mid x + y \text{ odd}\}$$

Since spins in the same sublattice do not interact with each other, we can simultaneously update all the spins in B and then W .

2. Two $N \times N$ matrices are generated:

- P containing the Boltzmann ratios for spins in sublattice B , as in step 4 of the Metropolis algorithm, and 0 in correspondence of W 's sites.
- Z containing random generated numbers from the interval $[0, 1]$

3. The element-wise operation $P > Z$ is performed, and spins corresponding to the True entries of the resulting matrix are flipped.
4. Steps 2 - 3 are repeated exchanging $B \leftrightarrow W$.
5. Steps 1 - 4 define a *lattice sweep* and are repeated for the duration of the simulation.

The checkerboard decomposition requires N to be even.

3.1.3 Ensemble simulation

In section 2.4 we highlighted the importance of computing observables by averaging an independent set of measurements. The efficiency of this process was increased by simulating in parallel an ensemble of identical lattices. This was achieved by expanding the structure of the initial *Numpy* array to hold an additional dimension, indexing the corresponding lattice in the ensemble.

3.2 Boundary conditions

Since simulations are performed on finite systems, it becomes necessary to deal with the 'edges' of the lattice. While boundary effects become unimportant as $N \rightarrow \infty$, they can drastically change the behaviour of the lattice for small N , e.g. by increasing the convergence time or creating surface energy terms.

In section 4.1 we compare the performance of two types of boundary conditions, with *periodic* (pbc) and *free* (fbc) boundary.

The first effectively eliminates the boundary by wrapping the lattice edges to form a torus, hence ensuring the system looks homogeneous, which is an essential feature of infinite lattices. It is achieved on a square lattice by making any site (x, y) interact with $(x, y \pm 1 \text{ (mod } N))$ and $(x \pm 1 \text{ (mod } N), y)$. The latter condition limits the spins on the edge to only interact with the inner neighbours, which resembles a small crystal with unconnected edges. A free lattice is obtained by padding a $N + 1$ -th column and row of the array with zeros and applying the same evolution algorithm as in the pbc case.

3.3 State storage

In this report it was often necessary to explore the evolution of the system in a narrow range of temperatures and external fields, respectively near the critical temperature and near $H = 0$, to fit the critical exponents. Therefore, the final spin configuration in a simulation was saved and used as the initial state for the following, significantly reducing the time of relaxation.

This is essential when measuring hysteresis, as this effect is only present when the same system is cycled through a range of external fields.

The adopted solution was to use a Python class to store the state and main methods of the lattice (see appendix A).

3.4 Extension to other dimensions

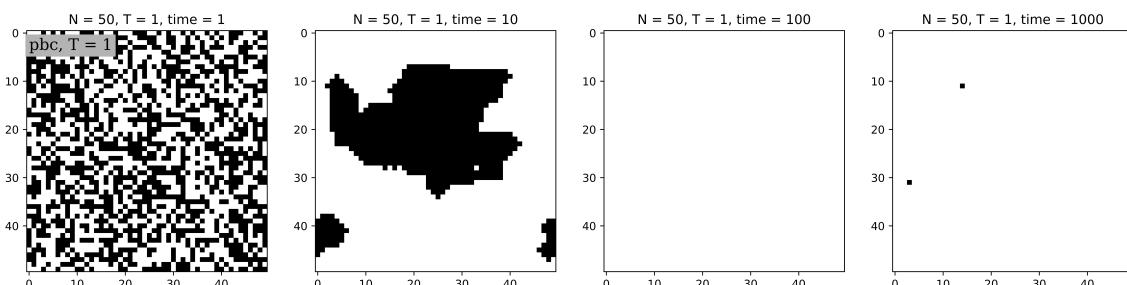
The checkerboard algorithm can be easily generalised to higher dimensional lattices: for example, a d -dimensional hypercube can be represented by a d -dimensional array and the algorithm outlined in section 3.1.2 can be applied by redefining sets B and W to contain points whose coordinates sum to even and odd integers respectively.

4 Results

4.1 Equilibrium and relaxation time

The system's energy and magnetisation behave as expected from theory, reproducing the features of equilibrium states in the three regimes of interest.

- $T \ll T_{crit}$: we measure $\langle E \rangle = -2J$ and $\langle |M| \rangle = 1$, corresponding to a minimum-energy state with all spins aligned.
- $T > T_{crit}$: $\langle E \rangle = 0$ and $\langle |M| \rangle = 0$, as the equilibrium state has all spins randomly aligned.
- $T \approx T_{crit}$: the magnetisation oscillates strongly around its equilibrium value, even flipping sign.



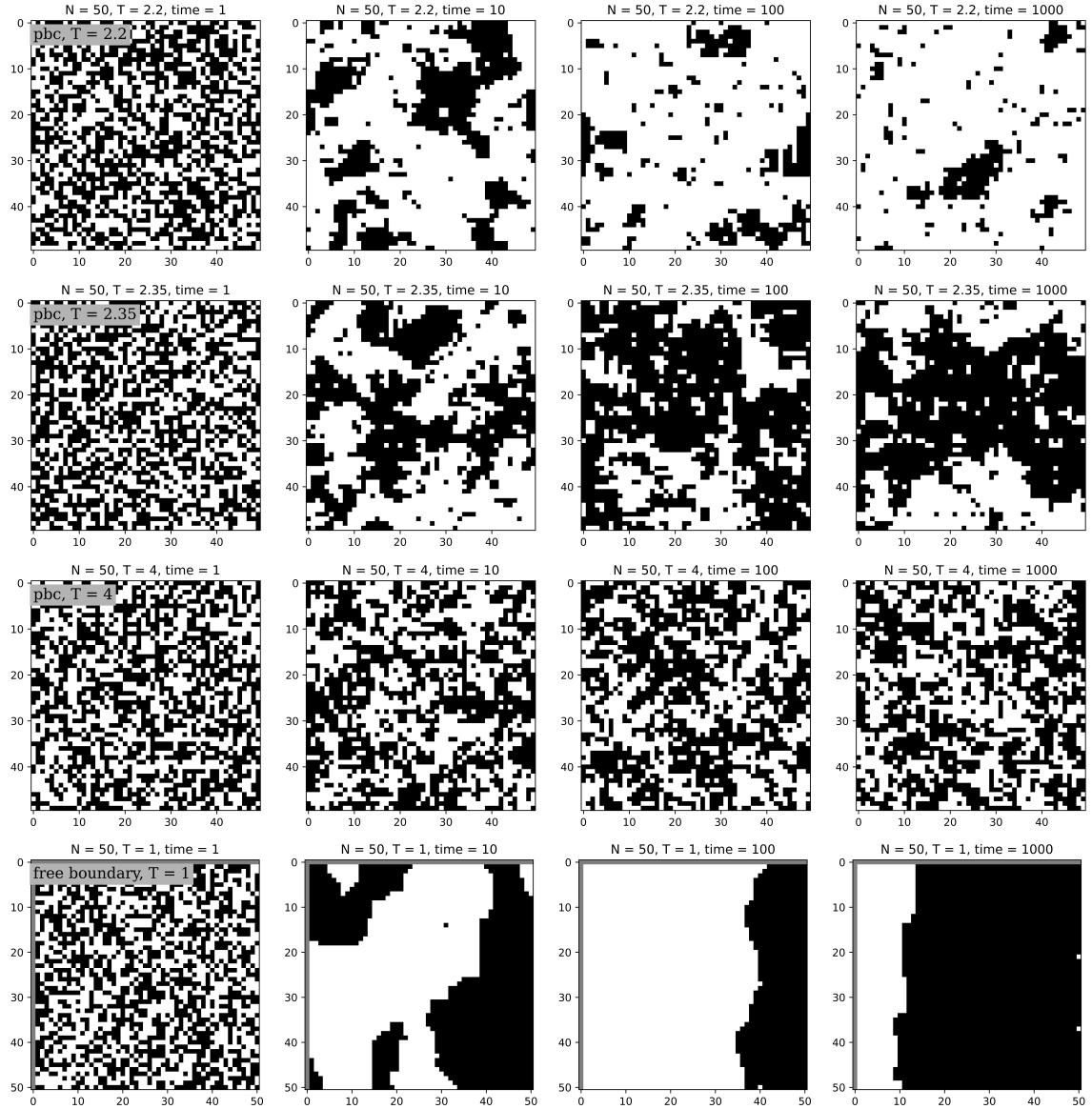


Figure 1: Each row shows the relaxation to equilibrium of lattices at different temperatures. The last row shows how lattices with a free boundary tend to maintain different grains even after long times.

The relaxation time is typically less than 10 lattice sweeps for high temperatures, but reaches peaks of hundreds of iterations for $T \approx T_{crit}$ and $T < 1$. Importantly, pbc guarantee a significantly slower relaxation time than fbc, especially for low temperatures (see fig. 2), hence pbc will be employed throughout the rest of the report.

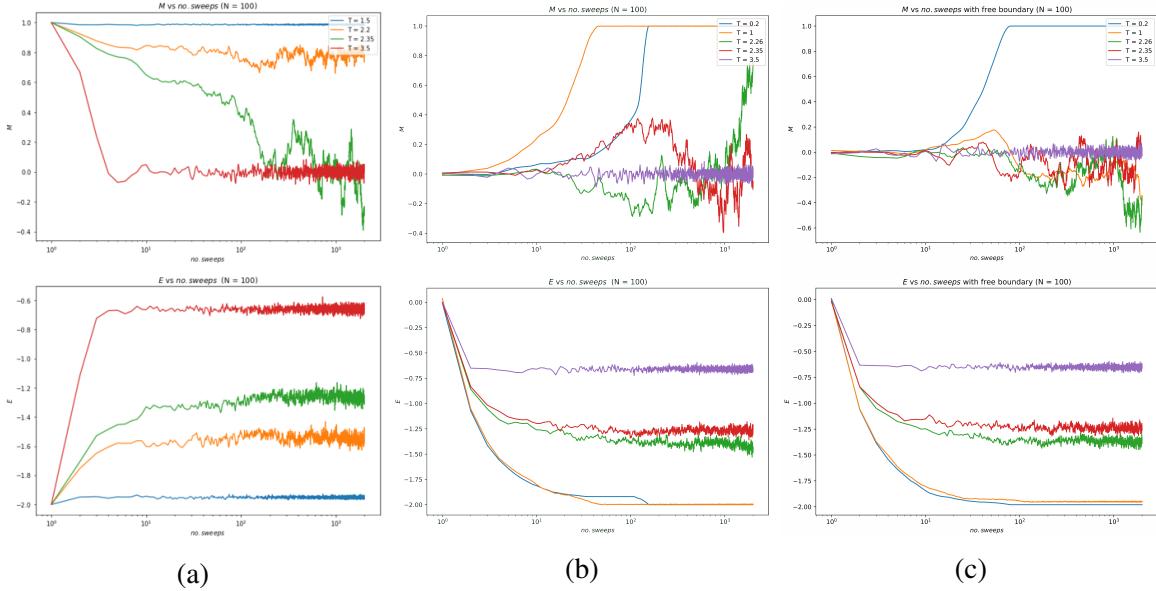


Figure 2: Magnetisation and energy of lattices are plotted over a logarithmic timescale for different temperatures. Fig. a) and b) show the evolution of a periodic lattice starting respectively from a configuration with all spins aligned and in random orientations. Fig c) shows a lattice with free boundary starting from a random microstate. It should be noted how the evolution corresponding to $T = 1$ in fig. c) does not reach equilibrium in 2000 iterations.

4.2 Decorrelation time

The autocorrelation of the order parameter is shown to decay as in equation 13. For low ($T < 1.5$) and high temperatures ($T > 3.5$) a single lattice sweep is sufficient to turn the initial spin configuration into an uncorrelated one. However, near T_{crit} , the decorrelation time rises up to 200 iterations. As a consequence, the error in time averages of observables measured when the lattice evolves around the critical temperature increases by a factor of $\sqrt{200} \approx 14$

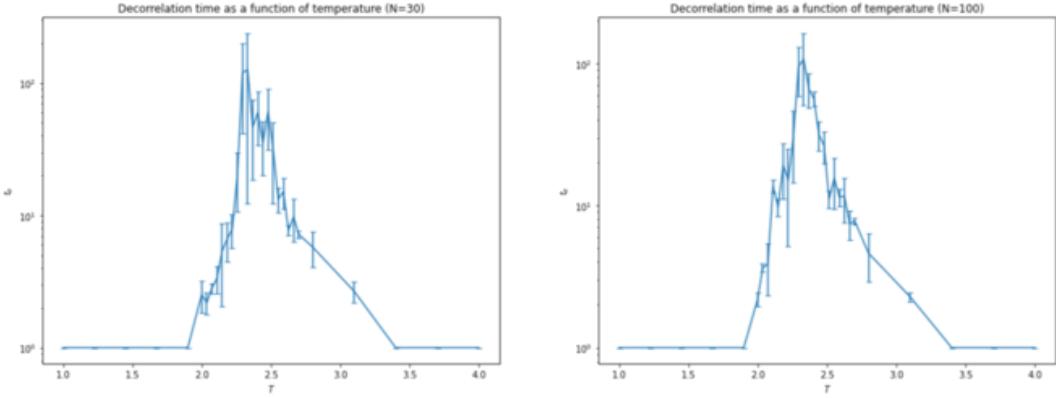


Figure 3: The decorrelation time of the Checkerboard algorithm is plotted near the critical range for lattices with $N = 30$ and $N = 100$.

4.3 Comparison to model

For the reasons explained in section 2.4.4, $\langle |M| \rangle$ will be used in the rest of the report as the approximation for the magnetisation in the infinite Ising model. The magnetisation and specific heat behave as expected, closely matching Onsager's curves as the lattice size grows.

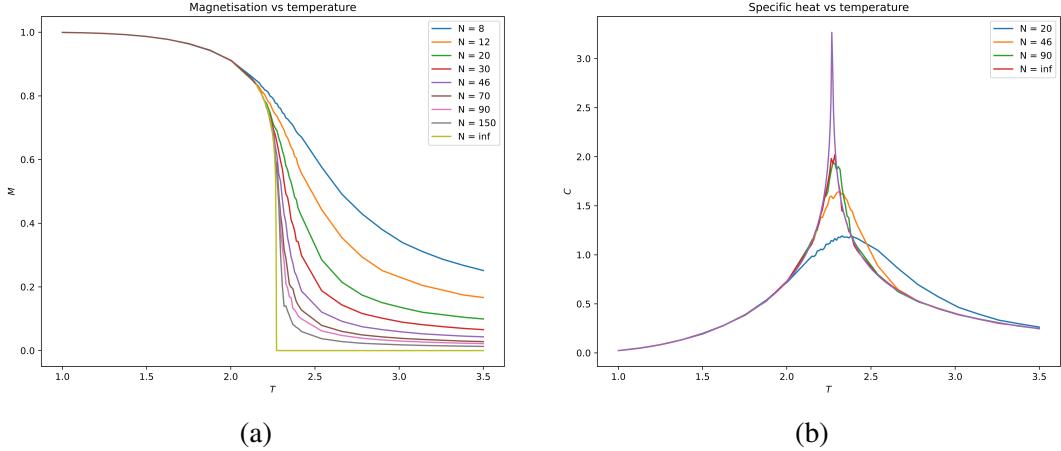


Figure 4: Fig a) shows that for big values of N , the magnetisation drops sharply in correspondence of the critical temperature, indicating an effective phase transition. In fig. b), the specific heat displays an increasingly singular behaviour near the critical temperature as N grows. The curves labelled by $N = \infty$ correspond to Onsager's predictions [3].

4.4 Magnetisation

As $T \rightarrow T_{crit}$, the phase transition produces a steep drop in the magnetisation. Using the function `curve_fit` from the Scipy library and fitting a power law with variable T_{crit} we obtain a critical exponent $\beta = 0.126 \pm 0.005$, in agreement with the theoretical value $\beta = 1/8$.

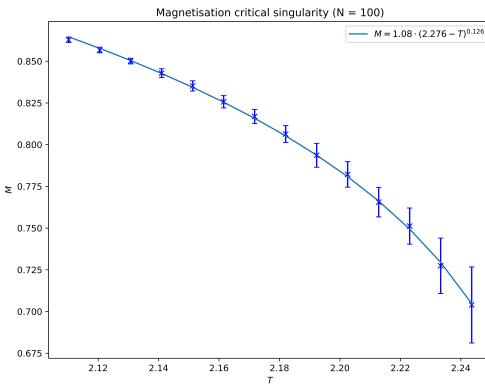


Figure 5: The temperature dependence of M is plotted near T_{crit} .

4.5 Specific heat

The specific heat behaves as expected from theory. Equation 9 predicts the specific heat to have a logarithmic singularity at $T = T_{crit}$, which does not fit the power law characterisation given in section 2.3. However, since a logarithm exhibits a slower growth than any power, fitting its critical exponent in the range where the specific heat is maximum predicts a value $\alpha = 0$, within two sigma of our measurement $\alpha = 0.2 \pm 0.1$. The error is obtained by comparing fitted values of α by removing or adding one data point to the temperature range considered. T_{crit} is chosen to be the temperature in correspondence of the maximum value of C .

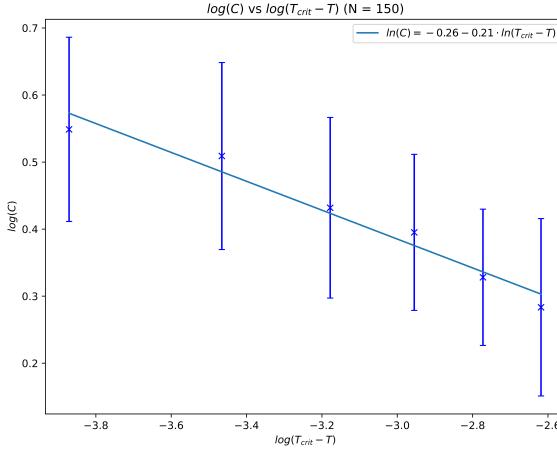


Figure 6: The singular behaviour of C is plotted near T_{crit} .

4.6 Magnetic susceptibility

Fitting a power law dependence for the susceptibility yields a critical exponent $\gamma = 1.80 \pm 0.13$, in agreement with the theoretical value $\gamma = 7/4$.

T_{crit} is chosen to be the temperature in correspondence of the maximum value of χ . The error for γ is bigger than for the previous exponents because of the intrinsically high variance of χ , caused in turn by the oscillating behaviour of $\langle |M| \rangle$.

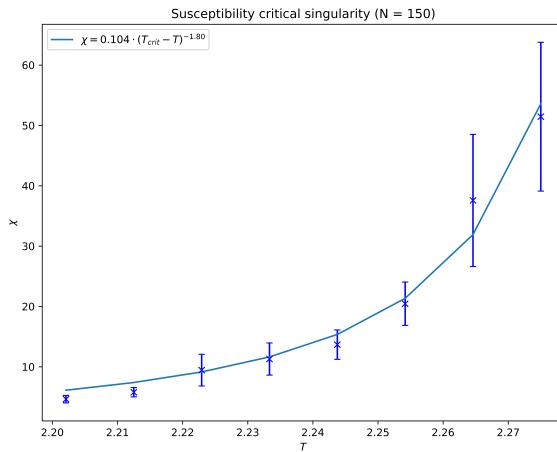


Figure 7: The singular behaviour of χ is plotted near T_{crit} .

4.7 Finite size scaling

Proper phase transitions do not occur in finite lattices, as illustrated in 2.3. Nonetheless, effective phase transitions are measurable even when N is finite in presence of sharp changes of the order parameter near the critical temperature. These are described by a finite scaling law [7]:

$$T_{crit}(N) = T_{crit}(\infty) + aN^{-1/\nu} \quad (15)$$

A dataset of critical temperatures for different lattice sizes is obtained by fitting the magnetisation with a power law singularity:

$$M(T) = b(T_{crit} - T)^\beta \quad (16)$$

where $\beta = 1/8$, as obtained in 4.4.

Fitting equation 15 through this dataset yields $T_{crit}(\infty) = 2.269 \pm 0.004$ and $\nu = 1.10 \pm 0.10$, in accordance with theoretical values $T_{crit}(\infty) = 2.269$ and $\nu = 1$ [10].

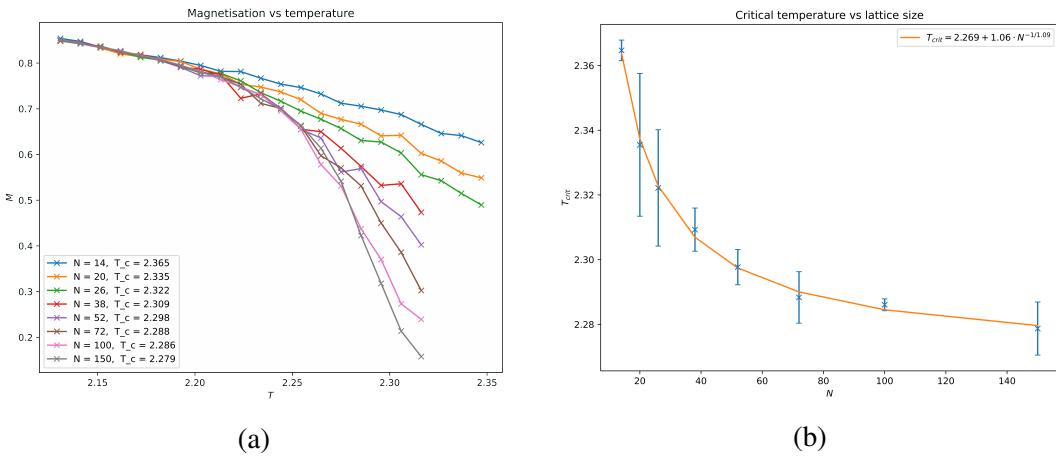


Figure 8: Fig a) shows the magnetisation dependence on temperature for a range of lattice sizes. In fig b) the finite size scaling of T_{crit} vs N is plotted

4.8 Critical isotherm

A power law fit $M = c|H|^{1/\delta}$ on the magnetisation data when $T = T_{crit}$ provides an estimate of $\delta = 15.05 \pm 0.19$, within one sigma of the expected $\delta = 15$.

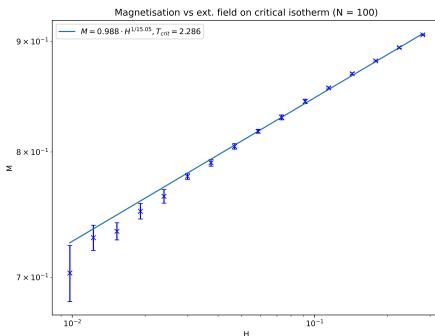


Figure 9: The magnetisation is plotted for small values of the external field with a log-log scale.

4.9 Hysteresis

The magnetic field is cycled 50 times from -1 to 1 in 30 equidistant steps. As expected from theory, for low temperatures we observe a loop in the $H - M$ plane, whose area shrinks progressively as temperature increases and becomes 0 when $T > T_{crit}$. Since hysteresis is an out-of-equilibrium phenomenon, the algorithmic time τ (measured in no. of lattice sweeps) that is given to the system to evolve for every value of the field H in the cycle has an impact on the system behaviour: bigger values of τ favour lattice relaxation, and thus lower magnitudes of H are necessary to drive magnetisation reversal, leading to a decrease in the area of the hysteresis loop.

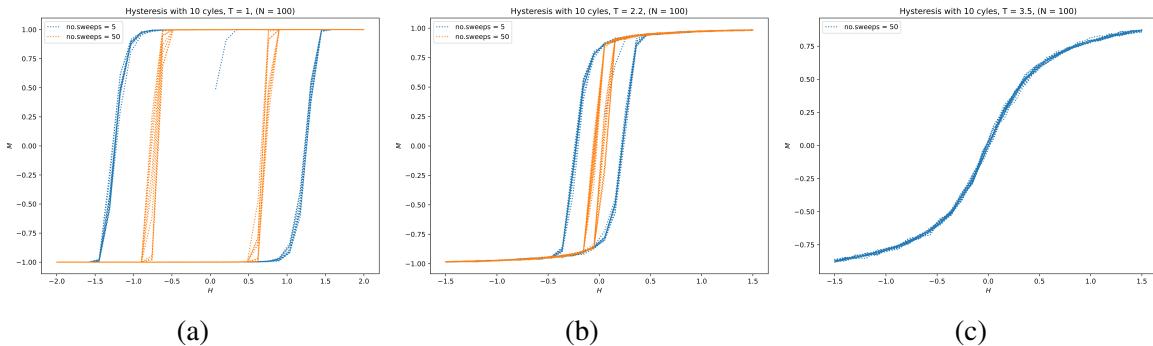
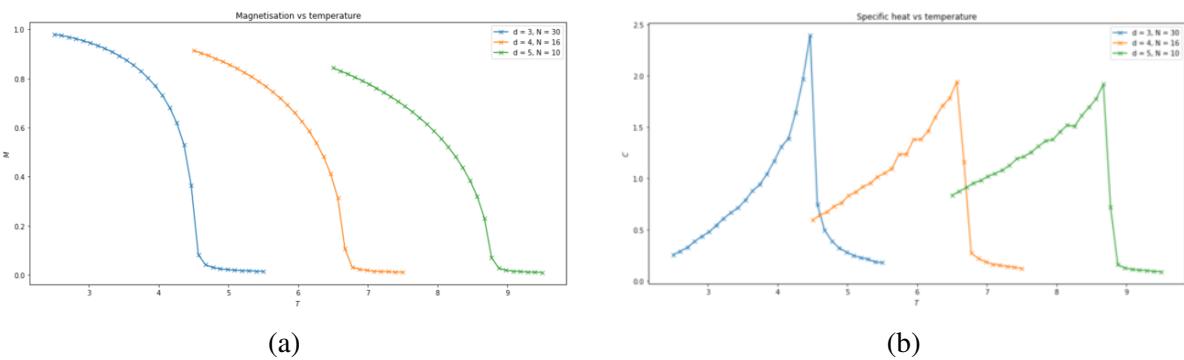
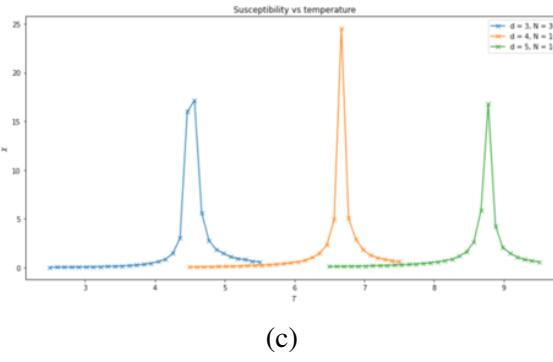


Figure 10: Fig a) and b) exhibit hysteresis, as they correspond to a system evolution at sub-critical temperatures. It is shown how an increased number of lattice sweeps per step decreases the total area of the loops. Fig c) displays no hysteresis loop due to the high temperature.

4.10 Higher dimensions

We present here the behaviour of hypercubic lattices in high dimension ($d = 3, 4, 5$).





(c)

Figure 11: Magnetisation, specific heat and susceptibility are plotted vs temperature in figs. a), b) and c) respectively. The region where their variations are the sharpest indicates the occurrence of a phase transition from an ordered to a disordered state.

4.10.1 3D

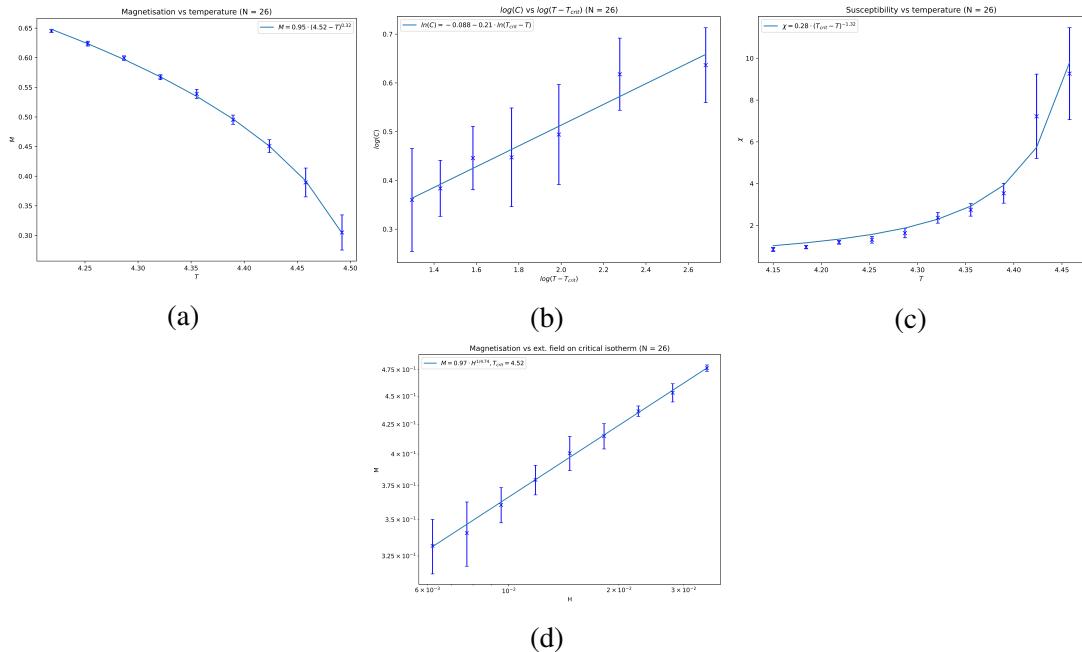


Figure 12: The critical behaviour of M , C and χ is plotted and the corresponding critical exponents are fitted, along with T_{crit} .

4.10.2 4D

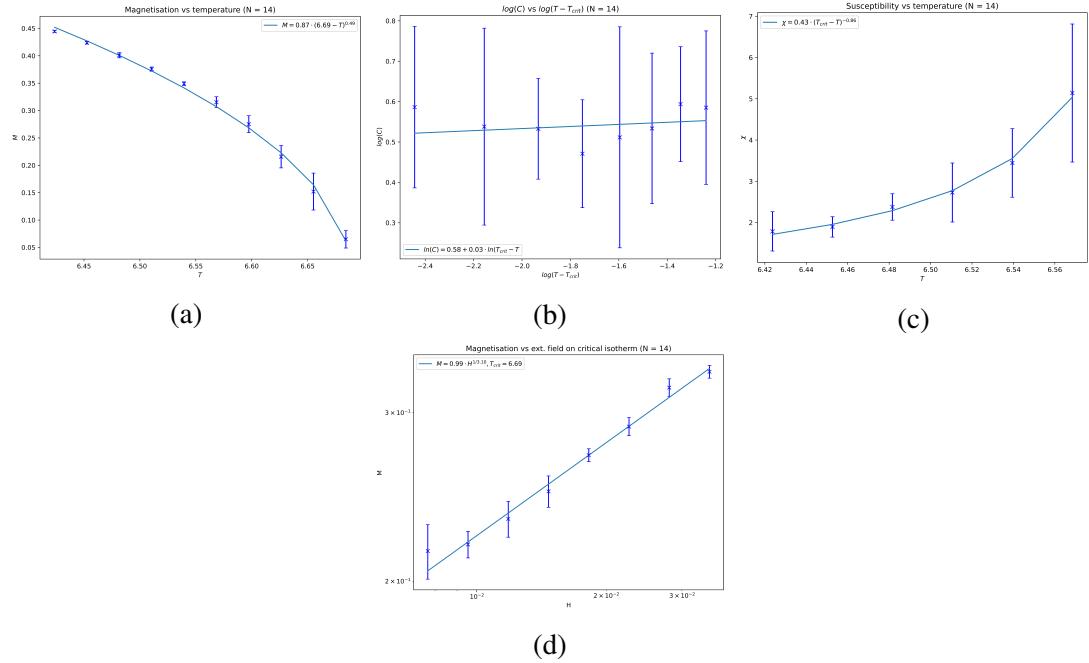


Figure 13: The critical behaviour of M , C and χ is plotted and the corresponding critical exponents are fitted, along with T_{crit} .

4.10.3 5D

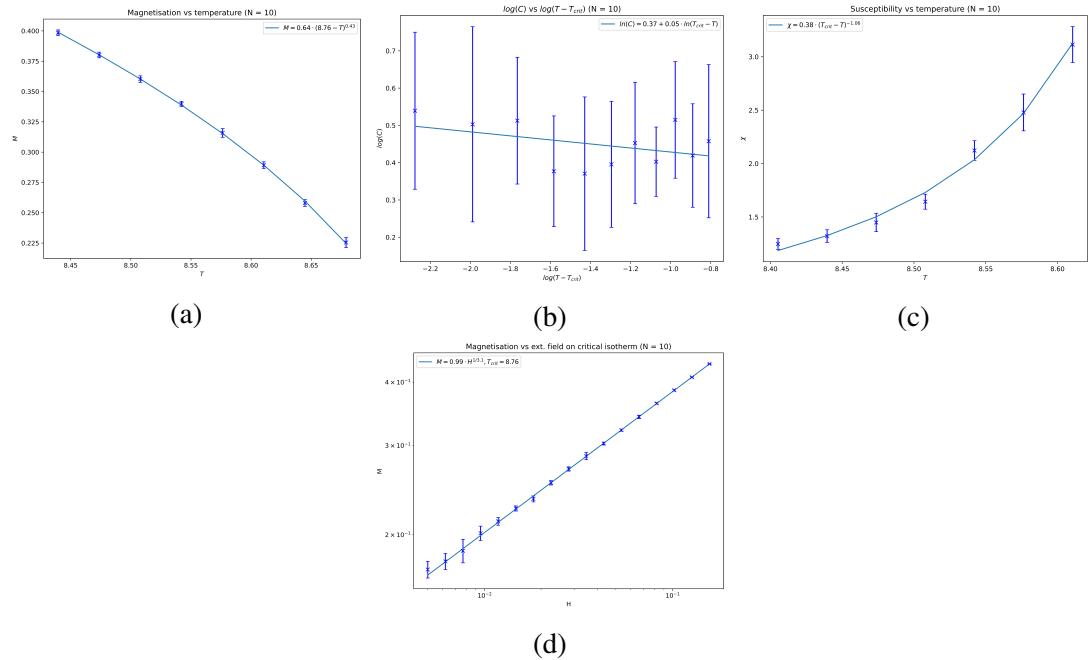


Figure 14: The critical behaviour of M , C and χ is plotted and the corresponding critical exponents are fitted, along with T_{crit} .

4.11 Summary of results

The magnetisation, specific heat and magnetic susceptibility of the Ising model in dimensions $d = 2, 3, 4, 5$ reproduce the behaviour predicted from the theory, where available, and more generally the scientific literature [3][5][11]. In the following table we report the main quantitative results.

	2D		3D		4D		5D	
	Fit.	Lit.	Fit.	Lit.	Fit.	Lit.	Fit.	Lit.
T_{crit}	2.269 ± 0.004	2.269	4.52 ± 0.02	4.511	6.69 ± 0.01	6.680	8.76 ± 0.01	8.778
α	0.2 ± 0.1	0	0.2 ± 0.1	0.110	0.03 ± 0.05	0	0.05 ± 0.03	0
β	0.126 ± 0.005	1/8	0.32 ± 0.2	0.326	0.49 ± 0.07	1/2	0.43 ± 0.04	1/2
γ	1.80 ± 0.13	7/4	1.32 ± 0.11	1.237	0.86 ± 0.1	1/2	1.05 ± 0.14	1
δ	15.05 ± 0.19	15	4.74 ± 0.13	4.79	3.10 ± 0.12	3	3.6 ± 0.1	3

Table 1: Critical temperatures and critical exponents obtained in this report are compared to the results from the literature. For dimensions $d = 4, 5$, the critical exponents match the predictions of mean field theory.

All values in 1 agree with the literature, with the exception of δ in a 5D lattice, which differs from the mean field theoretical value by six standard deviations, probably in part due to finite size effects. For the 2D case we also measured the finite size scaling exponent, $\nu = 1.10 \pm 0.10$, within one sigma of the ground truth $\nu = 1$.

5 Conclusions

In this report, we exploited a Monte Carlo technique to investigate the properties of the Ising model in the ferromagnetic case, for hypercubic lattices of dimensions $d = 2, 3, 4, 5$. The implementation of this method is based on the the Checkerboard algorithm (a vectorised version the Metropolis algorithm) optimised for ensemble simulation, as outlined in section 3.1.2. A preliminary analysis (4.1) of the 2D lattice was conducted, where we determined the convergence and decorrelation times of the lattice, and we verified that lattices under periodic boundary conditions relax faster than ones with a free boundary. This was followed by the study of the magnetisation, specific heat and magnetic susceptibility, whose dependence on temperature strongly agrees with the theory 4.3. In particular, these quantities indicate the presence of an effective phase transition, which is characterised by a critical temperatures and critical exponents reported in table 1. The results obtained find confirmation in the literature and they match mean field theory for $d \geq 4$.

Extensions to this investigation could consider 1D lattices: it would be interesting to measure how the finite size of the system influences the appearance of an ordered phase at low temperatures [12]. Besides, the universality of the critical exponents might be tested with different lattice shapes, such as with in triangular lattice. Lastly, our algorithm could be generalised to include a longer range of interactions between non-neighbouring spins, with differing coupling constants. Similar additions would allow us to model the magnetic properties of simple crystals, such as CuZn [13].

References

- [1] X-T Wu, DB Abraham, and Joseph O Indekeu. Apparent first-order wetting and anomalous scaling in the two-dimensional ising model. *Physical Review Letters*, 116(4):046101, 2016.
- [2] Dietrich Stauffer. Social applications of two-dimensional ising models. *American Journal of Physics*, 76(4):470–473, 2008.
- [3] Lars Onsager. Crystal statistics. i. a two-dimensional model with an order-disorder transition. *Physical Review*, 65(3-4):117, 1944.
- [4] Xin-Zeng Wu, Di Li, and Zheng Dai. Math 505 project: Ising model – phase transition. <https://dornsife.usc.edu/assets/sites/406/docs/505b/Ising.phase.transition.pdf>.
- [5] Jorge Garcia and Julio A Gonzalo. Accurate monte carlo critical exponents for ising lattices. *Physica A: Statistical Mechanics and its Applications*, 326(3-4):464–472, 2003.
- [6] David Tong. Lectures on statistical physics. *Notes online from the author's DAMTP teaching page*, 2012.
- [7] David P. Landau and Kurt Binder. *Importance sampling Monte Carlo methods*, page 68–137. Cambridge University Press, 3 edition, 2009.
- [8] Mark EJ Newman and Gerard T Barkema. *Monte Carlo methods in statistical physics*. Clarendon Press, 1999.
- [9] H Müller-Krumbhaar and K Binder. Dynamic properties of the monte carlo method in statistical mechanics. *Journal of Statistical Physics*, 8(1):1–24, 1973.
- [10] Eduardo Ibarra-García-Padilla, Carlos Gerardo Malanche-Flores, and Freddy Jackson Poveda-Cuevas. The hobbyhorse of magnetic systems: the ising model. *European Journal of Physics*, 37(6):065103, 2016.
- [11] Boris Kryzhanovsky, Leonid Litinskii, and Vladislav Egorov. Analytical expressions for ising models on high dimensional lattices. *Entropy*, 23(12):1665, 2021.
- [12] H Gould and J Tobochnik. Statistical and thermal physics: With computer applications, problem 5.10, 2007.
- [13] Shuaihang Pan, Gongcheng Yao, Maximilian Sokoluk, Zeyi Guan, and Xiaochun Li. Enhanced thermal stability in cu-40 wt% zn/wc nanocomposite. *Materials & Design*, 180:107964, 2019.

Total Words: 2948

A Code: the Lattice class

The lattice class stores the state of the lattice and contains the main methods used to produce useful data.

Ensembles of K d -dimensional lattices of side N are saved as numpy arrays with shape (K, N, \dots, N) where the N was repeated d times.

The most important methods are:

- energy: This produces an array with the same shape of the lattice, containing at each site the sum of the hamiltonian terms involving that site.
- update: corresponds to a lattice sweep of the Checkerboard algorithm.
- crit_data: calculates time and ensamble averages of M , χ and C and their errors for a given range of temperatures.

The class implementation is listed below:

```
#dependencies:
import numpy as np

# The lattice side cannot be odd, so we create an exception to avoid it
class OddSideException(Exception):
    def __init__(self):
        print("The side length (N) must be even")

class Lattice:

    # N: side
    # d: dimensions
    # K: number of lattices in the ensemble
    # free_bound: determines if boundary is free or periodic
    def __init__(self, N, d=2, K=1, free_bound=False):
        if N % 2 == 1:
            raise OddSideException
        self.N = N
        self.d = d
        self.free_bound = free_bound
        self.tot_sites = N ** d
        self.shape = [K] + [N for _ in range(d)]
        self.lat = np.ones(self.shape)
        if free_bound:
            pad_width = [[0, 0]] + [[1, 0] for _ in range(d)]
            self.lat = np.pad(self.lat, pad_width)

    # randomises spin configuration
    def random(self):
        self.lat = np.random.binomial(1, 0.5, self.shape) * 2 - 1
```

```

    if self.free_bound:
        pad_width = [[0, 0]] + [[1, 0] for _ in range(self.d)]
        self.lat = np.pad(self.lat, pad_width)

def align(self):
    self.lat = np.ones(self.shape)
    if self.free_bound:
        pad_width = [[0, 0]] + [[1, 0] for _ in range(self.d)]
        self.lat = np.pad(self.lat, pad_width)

# Hamiltonian term due to neighbours interaction and magnetic field
def energy(self, H=0):
    E_J = np.zeros_like(self.lat)
    for i in range(1, self.d + 1):
        E_J += np.roll(self.lat, 1, i) + np.roll(self.lat, -1, i)
    E_J *= -self.lat
    E_H = -H * self.lat
    return E_J + E_H

# we return the energy because it is a useful stat
def _update_chess(self, mask, T, H=0):
    E = self.energy(H)
    d_E = - 2 * E
    p = np.exp(-d_E / T) * mask
    self.lat[p > np.random.random(self.lat.shape)] *= -1
    energy = np.sum(E, axis=tuple([i for i in range(1, self.d + 1)]))/self.tot_sites
    return energy

def update(self, T, H=0, energy=False):
    mask = np.sum(np.indices(self.lat.shape), 0) % 2      # create multi-dimensional
    E = self._update_chess(mask, T, H)                      # chessboard
    self._update_chess(1 - mask, T, H)
    if energy:
        return E/2

def evolve(self, n, T, H=0):
    for _ in range(n):
        self.update(T, H)

# returns and ensemble of shape (K) containing lattices magnetisations
def magn(self):
    return np.sum(self.lat, axis=tuple([i for i in range(1, self.d + 1)]))/self.tot_sites

def stats_equ(self, n, T, H=0):
    Es, Ms = [], []
    for _ in range(n):
        M = np.average((self.magn()))

```

```

        E = np.average(self.update(T, H, energy=True))
        Es.append(E), Ms.append(M)
    return Es, Ms

# choose n >> tau
# first the system runs evolve() t0 times to reach equilibrium
def time_decorr(self, n, T, H=0, tau=0, t0=200):
    magns, acorrs, decorr_times = [], [], []
    self.evolve(t0, T, H)
    for _ in range(n):
        self.update(T, H)
        magns.append(self.magn())
    m_mean = np.average(magns, axis=0)
    d_m = magns - m_mean
    m_var = np.var(magns, axis=0)
    tau = n // 10 if tau==0 else tau    #default tau
    for t in range(tau):
        acorr = np.average(d_m[:n-t] * d_m[t:], axis=0) / m_var
        if np.any(acorr < 0.1):  # error becomes greater than the expected correlat
            if t < 4:
                return 1, 0
            else:
                break
        acorrs.append(acorr)
    acorrs = np.array(acorrs).T
    for series in acorrs:
        exp_fit = np.polyfit(range(len(series)), np.log(series), 1)
        decorr_times.append(-1 / exp_fit[0])
    decorr_time = np.average(decorr_times)
    error = np.std(decorr_times)
    return decorr_time, error

# calculates M, Chi, E, and C with corresponding errors
def stats(self, n, T, H=0, t0=200, abs=True):
    magns, E = [], []
    self.evolve(t0, T, H)
    for _ in range(n):
        magns.append(self.magn())
        E_spec = self.update(T, H, energy=True)
        E.append(E_spec)
    magns = np.abs(magns) if abs else magns
    Ms = np.average(magns, axis=0)
    Chis = np.var(magns, axis=0) * self.tot_sites / T
    Es = np.average(E, axis=0)
    Cs = np.var(E, axis=0) * self.tot_sites / (T ** 2)
    stats = np.average([Ms, Chis, Es, Cs], axis=1)
    errs = np.std([Ms, Chis, Es, Cs], axis=1)

```

```

    return stats, errs

# computes averages for temperatures in a range T_range
def crit_data(self, T_range, H=0, n=1000, t0=400):
    Ms, M_errs = [], []
    Chis, Chi_errs = [], []
    Cs, C_errs = [], []
    for T in T_range:
        stat, err = self.stats(n, T, H=H, t0=t0)
        Ms.append(stat[0]), M_errs.append(err[0])
        Chis.append(stat[1]), Chi_errs.append(err[1])
        Cs.append(stat[3]), C_errs.append(err[3])
    return [Ms, Chis, Cs], [M_errs, Chi_errs, C_errs]

def hysteresys(self, n, T, H_max, H_steps, cycles):
    magns = []
    Hs = np.linspace(-H_max, H_max, H_steps)
    Hs = np.concatenate([Hs[:-1], np.flip(Hs[1:])])
    Hs = np.concatenate([Hs for _ in range(cycles)])
    Hs = Hs[H_steps//2:]
    for H in Hs:
        self.evolve(n, T, H)
        magns.append(self.magn())
    return Hs, magns

```