Metropolis–Hastings simulations of ferromagnetic Ising model

BGN: 6915X [1)]

[1)]University of Cambridge, Cavendish Laboratory, 19 J.J. Thompson Avenue, Cambridge CB3 0HE, UK

May 12, 2021

Abstract

The Ising model is an integral part of statistical physics for describing the magnetization of a material. With an increase in the dimension of the system, the area of known theoretical results narrows down. 2D Ising model is exactly solved without an external field, while cannot be exactly solved with external field. In this work, we have used Metropolis-Hasting numerical simulation method for the Markovian dynamics of the 2D Ising model. We compared our numerical results with known theoretical results, such as the critical temperature and its finite-size scaling relation. An important discrepancy was found that heat capacity scales differently at different boundary conditions. Time-autocorrelation functions were calculated and relaxation times were estimated at different temperatures. Additionally, the case of a non-zero external field was also investigated. Magnetic susceptibility was calculated in two different ways. The magnetization of the system was calculated with and without an external field and the phase transition was demonstrated.

1 Introduction

In physics, very few many-body systems can be solved. The word "solved" is used in a general sense: find dynamics of the system, find out stationary solutions, properties of the system in equilibrium (if ever exists), etc. For example, the equations of motion of two gravitationally interacting objects can be solved exactly. However, when a third object is added to the system, the equation does not have a closed-form solution in general. The transition from two to three objects massively complicated the equations.

Despite this, only physicists have the courage to move on and try to solve systems of a massive number of particles: gasses and fluids. And there are very successful models in fluid dynamics and statistical physics explaining a complex phenomenon in these systems. This is archived only by making the right assumptions and approximations to the system, such that it's not too much different from reality and at the same time is possible to solve with available mathematical tools.

An example of such a many-body interacting system is the Ising model, which we investigate in this review. It is a simple lattice model of interacting magnetic spins, a mathematical model of ferromagnetism. The success of this model is the existence analytical solution, despite having nontrivial interactions. The current status of the Ising model with nearest neighbour interactions is the following: for the 1D Ising model, i.e. spins in a chain, the partition function is calculated, Markovian dynamics is also solved, both with external magnetic field [1, 2, 3, 4, 5]. In two dimensions, the partition function is calculated only in zero external fields by Onsager[1]. In three dimensions, despite a lot of effort, no exact analytical solution is known.

We see that in 2D, when we complicate the situation a little, e.g. add an external field or next nearest neighbour interactions, the system ceases to be solvable. In this kind of situations, it is important to have computational tools, which can simulate the system, or numerically solve

equations of dynamics. In the following sections, we will simulate the system and compare numerical results with Onsager's analytical results. We will also investigate the system in the external time-dependent magnetic field.

In the next section, we will shortly review the required theory and algorithm of the numerical method we have used. The following sections cover the results of the simulation, each of them is self-contained and present its own method of analysis. In the last section, we summarise the results and the review is concluded.

2 Background theory and algorithm of simulation

We consider the two dimensional lattice of interacting spins, which has the following Hamiltonian:

$$\mathcal{H} = -J \sum_{<ij>} s_i s_j - \mu H \sum_{i=1} s_i \tag{2.1}$$

where s_i represents the spin at lattice site i, $s_i = \pm 1$, J is spin-spin interaction energy, $J > 0$ for ferromagnetism, H is the external magnetic field strength, μ the magnetic moment. The sum $< ij >$ is over nearest neighbour sites only. The configuration of the system is fully described by state vector $\{s_i\}$, i.e. spins at all sites. In total, there are 2^N configurations where N is the number of sites. The energy and magnetisation for each state are defined as

$$E = \mathcal{H}(\{s_i\}) \qquad \text{and} \qquad M = \frac{1}{N^2} \sum_i s_i \tag{2.2}$$

We assume that the system is coupled to the thermostat of temperature T. Statistical mechanics state that in equilibrium, the probability of each configuration is given by Gibbs distribution [6]:

$$P(\{s_i\}) = \frac{1}{\mathcal{Z}} \exp\left(-\frac{E(\{s_i\})}{T}\right) \tag{2.3}$$

where $\mathcal{Z}$ is a normalization constant (well known as partition function). The interaction with the thermostat will induce spontaneous transition between states. These transition rates are very specific: the corresponding steady state must have the probabilities of Gibbs distribution (eq. 2.3). Otherwise it will not model the system in equilibrium [6].

In formal sense, we have a normalized probability distribution (eq. 2.3) (as partition function $\mathcal{Z}$ is hard to compute) and we want to draw samples from this distribution. This is exactly the settings of Metropolis–Hastings (MH) algorithm [7, 8]. The protocol for an ising model is the following:

0. start from some configuration $\{s_i\}$, possibly random
1. pick a random site on the lattice s_α
2. Find the energy ΔE required to flip the spin at the selected site:

$$\Delta E = E(\{s_1, s_2, \dots, -s_\alpha, \dots\}) - E(\{s_1, s_2, \dots, s_\alpha, \dots\}) \tag{2.4}$$

3. flip the spin with probability

$$p = \min\left\{1, \exp\left(-\frac{\Delta E}{T}\right)\right\} \tag{2.5}$$

4. register resulting state as a sample, *even is the spin has not flipped* in the previous step.

Repeat steps $1 - 4$. It is guaranteed that after sufficient steps samples come from equilibrium distribution 2.3 [7]. More details about Markovian dynamics can be found in [3, 4, 5].

We will use $J = H = \mu = 1$, i.e. measuring energy in units of J, the magnetic field in units of J/μ. Additionally, k_b Boltzmann constant set to 1, i.e. temperature in units of energy.

2.1 Implementation details

We will consider two dimensional square lattices with same interaction J in x and y direction, i.e. $J_x = J_y = J$. It is not possible to simulate infinite lattice, so we restrict lattice size to $N \times N$ and impose periodic boundary conditions unless otherwise stated.

The simulation engine was implemented with C++ language to achieve the best performance. With language bindings provided by `pybind11` [9], the code is accessible to Python. All analysis is done with data analysis tools of Python. Because of C++ we achieved $\sim$ 1second runtime on single-core for 10^7 interations(steps) of the algorithm. Python allows us to run simulations with different parameters in parallel so we have achieved additional speedup. The codebase is available on GitHub [10] and code usage is given in the Appendix **??**

3 Relaxation time of the system

In this section, we will present our method to measure relaxation times in units of the number of iterations(steps). Additionally, we will argue how it scales with the size of the lattice.

First, we present our method of estimation of the relaxation time. Basically, we have estimated the time required for some quantity of the system to come close to its stationary value. In our case, it was average magnetisation and the energy of the system. Note that the initial state is important in our estimation method, as if the initial value of the quantity is close to its stationary value, it would be hard to see the relaxation period. Therefore we will use both "constant" (i.e. all spins are in the up direction) and "random" (i.e. spins are directed uniformly randomly up and down) initial states.

The algorithm is the following:

1. set pointer `start` to the beginning of the series `s[t]`,
2. calculate average `m` of the series from pointer `start` to the end. WLOG[1] $\texttt{m} > 0$,
3. update pointer `start` to the smallest time s.t. $\texttt{s(t)} \geq \texttt{m}$,
4. then repeat 2 to 3 until convergence.

By this method, we have calculated relaxation time for each realisation at different temperatures and different initial conditions. The results are presented on figure 1. Note that at low temperatures, where the at the stationary phase the magnetisation is close to its extremal value ($M = +1$ or -1), our estimation method may not give reliable result, thus in that region, we will consider the estimations from random initial state (see figure 1). Similarly, for high-temperature regions, the stationary magnetisation will be close to zero, thus we consider the constant initial state. Thus the unusual peak at $T = 0.6..0.8$ region was obtained from simulations from constant initial state, and stationary state is close to the constant initial state.

The relaxation time is a well-defined quantity, however, the developed method gives us realisations of some random variable which is the noisy variant of the relaxation time.

Relaxation time is characteristics of the coupling of the system and the reservoir. It is related to the flipping rate of each spin in the lattice, which is temperature-dependent: the stronger the coupling, the flipping rate is higher. However, in the Metropolis Hasting simulations the flipping rate of a single site is proportional to the iteration rate divided by the number of spins:

$$\begin{matrix}\text{flipping rate} \\ \text{of single site}\end{matrix} \propto \frac{1}{\text{number of sites}} \tag{3.1}$$

Thus relaxation time divided number of lattice sites should be approximately invariant of the lattice size. We use this fact to predict approximate relaxation time for arbitrary lattice size at arbitrary temperature. On figure 1 we fitted qualitative relaxation time dependence on temperature and used it in calculations in next sections.

[1] without loss of generality

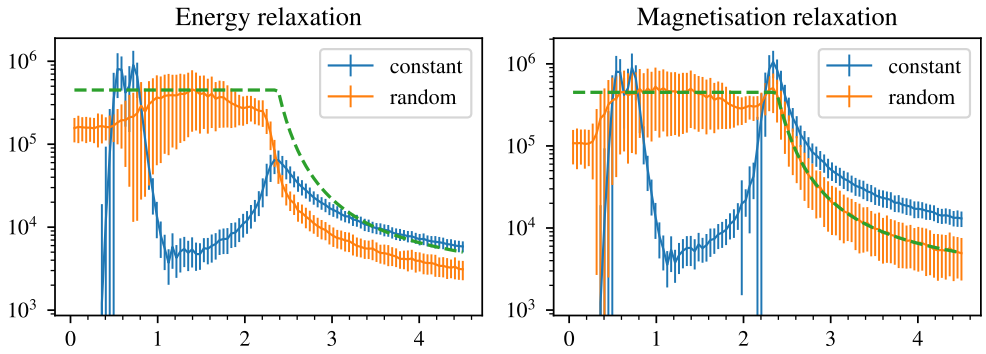


Figure 1: This figure present energy (left) and magnetisation (right) relaxation time v.s. the temperature of the system. On each figure, we have plotted the estimated value for two initial conditions. The dotted line is the fitted function for the relaxation time estimate, which is used for our future calculations. In above simulations lattice size was taken 32×32 ($N = 32$), 10^7 iterations, 0.5σ errorbars are given for 10^3 runs at each temperature.

4 Time autocorrelation function of magnetization

Metropolis-Hasting return us a series of magnetisations of the system. The autocovariance of the series is defines as $A(\tau) = \langle M'(t)M'(t+\tau)\rangle$, where $M' = M - \langle M\rangle$ and M is the magnetisation in (2.2). The autocorrelation is given by $a(\tau) = A(\tau)/A(0)$.

$$a(\tau) = \frac{A(\tau)}{A(0)} = \frac{\langle M'(t)M'(t+\tau)\rangle}{\langle M'^2\rangle} \tag{4.1}$$

For the system in stationary regime the quantity in eq. (4.1) is independent of t as the distribution of the series is invariant under time translations.

Having time series of $M(t)$ we can easily evaluate the autocorrelation, and thus find the decorrelation time, i.e. the time lag τ_e over which the autocorrelation falls by e. The results are plotted on figure 2.

This means that the magnetisation of the system is highly correlated between two time-points $\mid t_1 - t_2 \mid < \tau_e$. We may assume other quantities of the system (e.g. energy) have similar decorrelation time. Thus for estimating statistical moments of some quantity, the samples must come from a much larger time interval compared with τ_e. It is convincing that the graphs on figs. 1 and 2 have similar tails above critical temperature: Note that the scale of the graphs is logarithmic and decorrelation/relaxation time decreases faster than exponential.

5 Mean magnetisation and phase transition

The 2D Ising model is very successful, as despite its simplicity it undergoes a second-order phase transition. The ordering parameter is the magnetisation of the system. Although for positive temperatures the mean magnetisation is zero, however below some critical temperature T_c symmetry breaking occurs and magnetization becomes positive or negative with equal probability. The T_c depends on the size of the lattice. Onsager's famous result for the critical temperature in case of the infinite lattice is [1]

$$T_c = \frac{2}{\ln(1+\sqrt{2})} \approx 2.269 \tag{5.1}$$

The dependence of T_c of lattice size is investigated in the next sections.

At low temperatures, from a random initial state, the system mostly converges to the extremal magnetisation state ($M = \pm 1$). Relatively rarely system stuck in the metastable states. To see

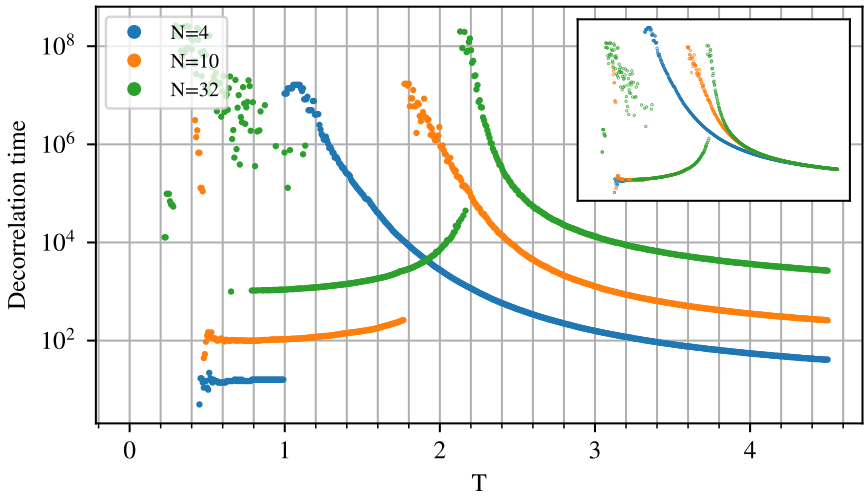


Figure 2: Above are presented decorrelation time at different temperatures for 3 different lattice sizes (N^2 lattice sites). Inset demonstrates the same data with normalized decorrelation time according eq. (3.1) and we see that away from critical temperature they behave similarly. The critical temperatures are $T_c^{(4)} = 1.05$, $T_c^{(10)} = 1.78$, $T_c^{(10)} = 2.13$. MH simulation was done with 10^9 steps with periodic boundary conditions. Artefacts below $T = 1$ are due to random errors in the data and estimation method of the decorrelation time, and can be discarded.

the statistical weight of these events see histogram on figure 3. The existence of metastable states can be explained easily by considering a lattice, with positive spins on the top half, and negative spins on the bottom half, such that there is a horizontal flat boundary between two halves; in this configuration, any spin-flip requires energy and low-temperature weak fluctuations are not enough to leave the current state. In other words, there is a potential barrier between this state and state with all the spins aligned (which has lower energy) which the system can't overcome. These metastable states vanish above $T = 1.5$ as fluctuations become sufficient. However, they still "lose" the interaction forces and the system converge to the ordered phase.

On figure 4 are shown the states of the system at different temperatures. The long-range correlations can be clearly seen at low temperatures.

6 Heat Capacity and critical behaviour

The critical behaviour of the system was investigated by measuring heat capacity. The Fluctuation-dissipation theorem connects specific heat capacity with the fluctuations of the energy of the system:

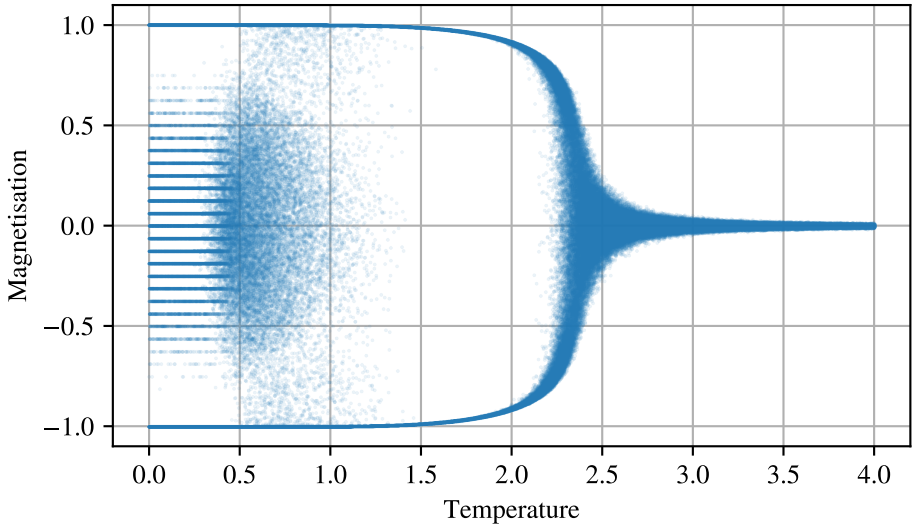
$$C = \frac{1}{NT^2}\left[\langle E^2\rangle - \langle E\rangle^2\right] \tag{6.1}$$

One may try to measure heat capacity from the measurements of the energy versus temperature, then differentiating it. But much accurate and faster method is to do it by the above equation.

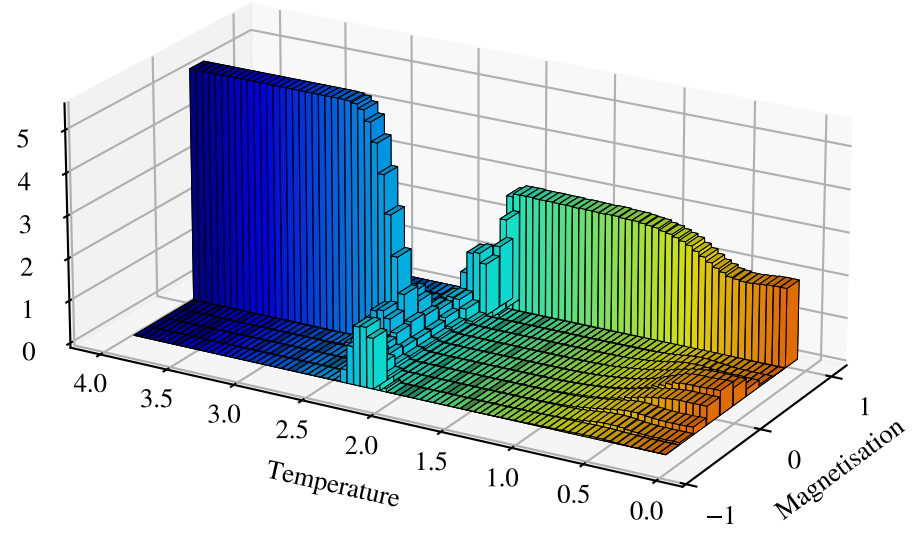
Heat capacity has a maximum at the critical temperature. These maxima were found at different lattice sizes and shown on figure 5. Conventional finite-size scaling analysis gives the following dependence of critical temperature on lattice size [12, 13]:

$$T_C(N) = T_C(\infty) + \frac{a}{N^{1/\nu}} \tag{6.2}$$

The measurements were made with both *free* and *periodic* boundary conditions. Surprisingly, we get different dependence on the lattice size. It was expected that they will be slightly different, but we got different signs for the parameter a, i.e. for periodic boundary conditions critical temperature for finite lattice is higher then Onsager's result for infinite lattice (eq. 5.1) which



(a) Above demonstrated the result from each run.



(b) 3d histogram of the points on the top figure. Here we cut the M axis leaving out high bars around $M = -1$, such that the inner region is visible.

Figure 3: Magnetisation was calculated only using the data after relaxation. Phase transition at $T_c \approx 2.4$ is clearly seen. Simulation has started from random orientations of the spins, with lattice size $N = 32$, 10^7 steps and 600 000 independent runs in total with periodic boundary conditions. This was also demonstrated as a video and more features can be seen [11].

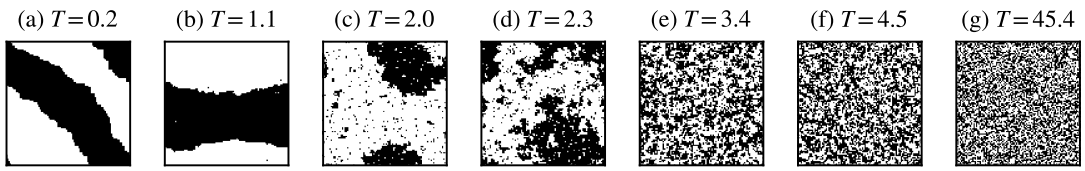


Figure 4: These are captures of the state in the stationary regime at different temperatures. At low temperatures intentionally the metastable states are shown. It can be seen that on (b) boundaries are less sharp compared to (a). On (c) we start to see fragmentation, (d) is close to the critical temperature. After (d), the next images become close to just random orientations of the spins and the spatial length of correlations decrease with temperature. Lattice size is $N = 100$, i.e. 10^4 sites, with periodic boundary conditions.

is counter-intuitive. However we can rationalize that for finite lattice $T_c^{(\text{periodic})}$ is larger then $T_c^{(\text{free})}$: the phase transition is a interplay of fluctuations and interactions. At ordered state interactions dominate, while at disordered state fluctuations dominate. When we change the boundary conditions to periodic, more interaction pathways are created (e.g. now spin on the left boundary can interact with the spin on the right boundary). This effectively increases interactions and suppress fluctuations. Thus we expect that phase transition undergoes at a slightly higher temperature. But at this stage, we can't explain why the sign of a is different.

The finite-scaling relations and in particular the eq. (6.2) are asymptotic formulas, i.e they are valid at sufficiently small limits of $1/N$ (large lattices). We were tracking the residuals of the fitted curve and eliminated small-N datapoints until the fitting error was unbiased and close to normal distribution. On fig. 5 only points with orange background were used for estimation parameters.

For free boundary conditions we got $T_c^{(\text{free})} = 202829 \pm 0.0005$ (overestimated) and for periodic boundaries $T_c^{(\text{periodic})} = 2.2624 \pm 0.0006$ (underestimated). The errors are very tight and are biased from theoretical value (eq. 5.1). This can be explained by the asymptotic nature of the formula (6.2): for large values our results have errors but function $T_c(N)$ does not vary much, variation of the function occurs at small N and we argued above that the finite-size scaling is not valid for small N. Another source of error may come from pseudo-random number generators: these are not purely random and our results highly rely on them. The numerical precision errors are not significant here, as everything was done using integers (except the flipping part of the algorithm eq. (2.5)).

7 Behaviour in the external field

So far we were experimenting in zero external field regime. In this section we review the behaviour in the external field. However, an important quantity, which is related to the external field, can be measured without introducing external field. The magnetic susceptibility is defined as

$$\chi = \frac{\langle M \rangle}{H} \tag{7.1}$$

where M is the magnetisation per site (eq. 2.2), $\langle \dots \rangle$ stands for ensemble average, H is the external field. Fluctuation-dissipation theorem also connects magnetic susceptibility with the fluctuations of the magnetisation of the system:

$$\chi = \frac{1}{NT} \left[\langle M^2 \rangle - \langle M \rangle^2 \right] \tag{7.2}$$

Note the similarity to the equation 6.1. The results of this method are shown on figure 6. The eq. (7.1) uses ensemble average, which is a theoretical concept. Below the critical temperature symmetry breaking occurs and the system shows different susceptibility at positive or negative branches on fig. 3. Theoretically, if time interval were sufficiently large, there is a nonzero probability to jump between branches at positive temperatures. Ensemble average also accounts for this hopping, however, in the numerical simulation the time is limited and these low-probability events cannot be seen. Therefore we see the highly dispersed values of magnetic susceptibility below the critical temperature.

We have measures magnetic susceptibility by measuring mean magnetisation at different external field values, then taking the slope. It is more efficient to do by varying the external field over time. The inverse frequency (period) of the external field must be sufficiently large than the relaxation time of the system, such that we can assume that system is in equilibrium at every point of time (adiabatic approximation). However, this method is limited by the relaxation time of the system: it increases at lower temperatures (see fig. 1) and this method becomes computationally more expensive. The results of both methods are shown on fig. 6.

Even in adiabatic limit, the system shows hysteresis effects: the system tends to preserve its magnetisation while the field is changing. E.g. at low temperatures, when magnetisation is -1, for a range of values of the external field, it preserves its magnetisation. When the field is higher

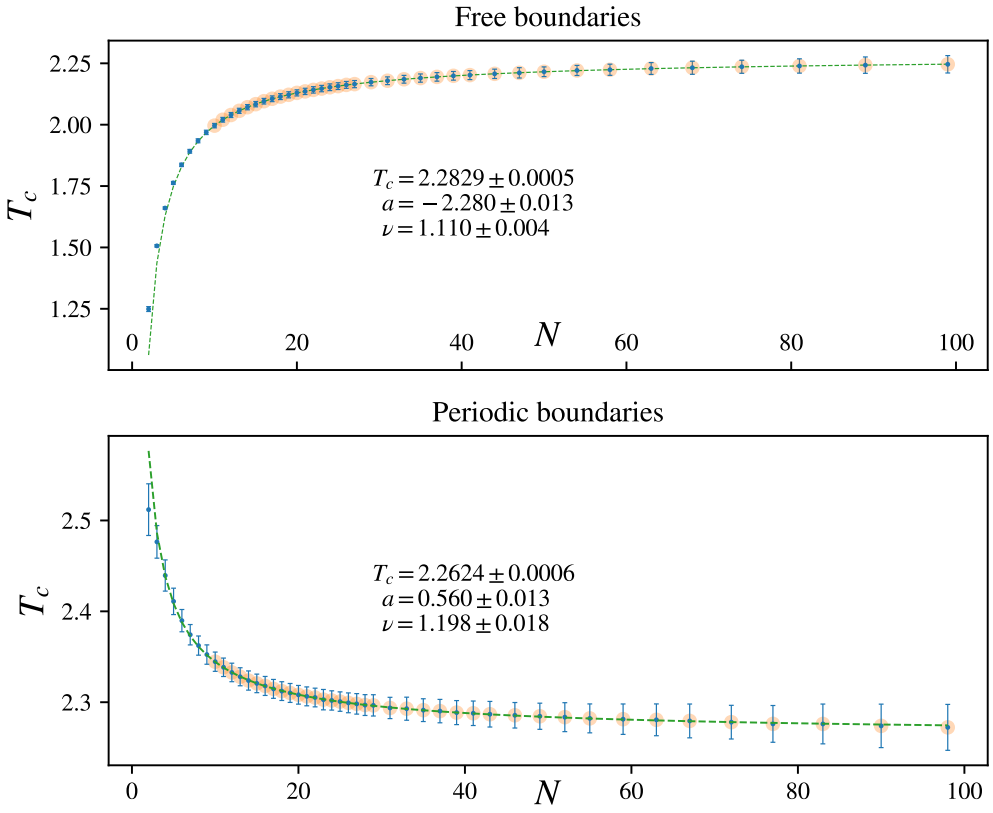


Figure 5: Above are shown the dependence of the critical temperature of the lattice size N. The critical temperature was found from the maximum of C(T) specific heat capacity: 4000 simulations were made around this maximum (over the range $T_c \pm 0.05$), then a parabola was fitted. The vertex of the parabola was taken as the peak of the curve. Simulations were done at free and periodic boundary conditions. Fitted parameters of eq. (6.2) are shown on the plot. The points with the orange background are used in the curve-fitting algorithm (see main text). The error bars are exaggerated (50σ) and just for demonstrating how the errors scale with temperature. Particularly, errors increase with temperature as well as with lattice size (in both cases the fluctuations increase).

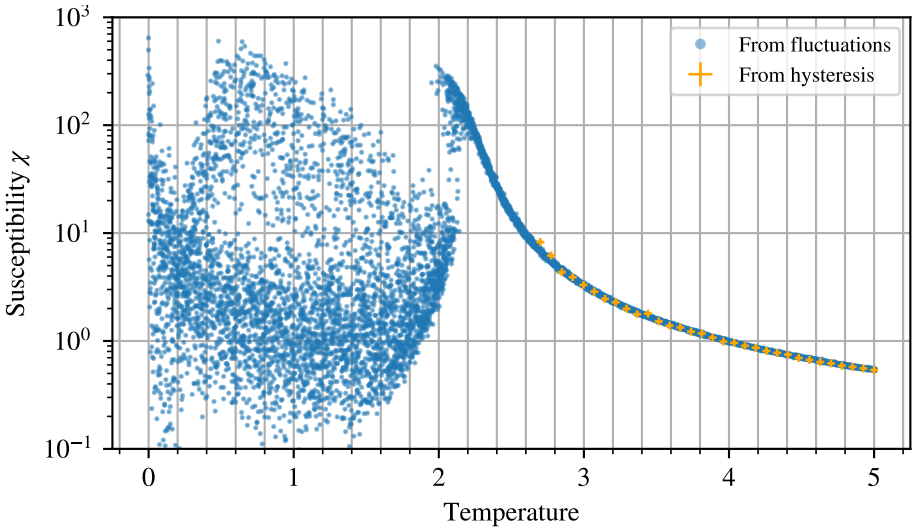


Figure 6: Magnetic susceptibility was measured in two different ways. The blue dots were calculated from fluctuations (eq. 7.1). The orange crosses were found from the maximum slopes of hysteresis curves (see figure 7).In the region where both methods are calculated the results are consistent. The lattice size is $N = 32$, with periodic boundary conditions, 10^4 runs in total.

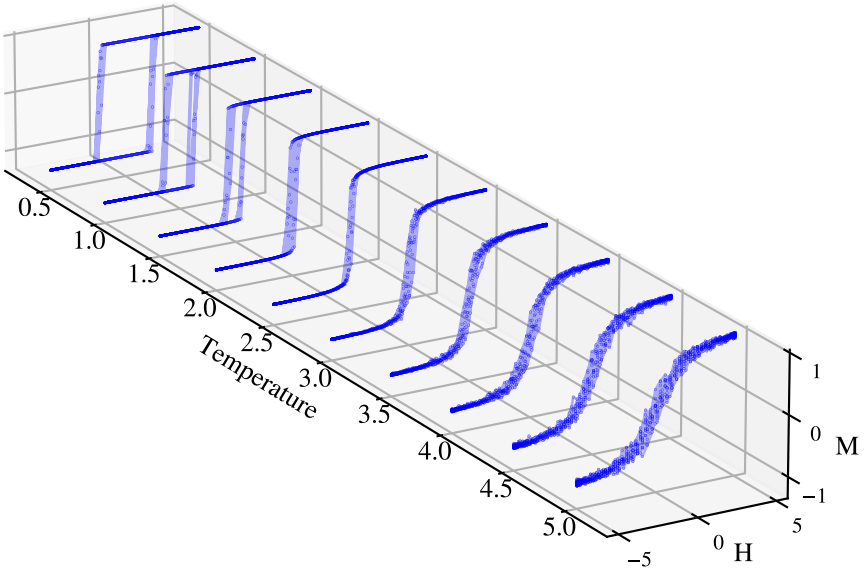


Figure 7: Hysteresis curve are plotted at different temperatures. Magnetic field was changed periodically in $(-5..5)$ interval, s.t. period 100relaxation time, i.e. adiabatic approximation can be made. With decreasing temperature, the slope of the curve increases until critical temperature. Below that we start to see hysteresis effects. In this graph the system size was dramatically decreased to $N = 4$ to be able to preserve adiabaticity, as relaxation time increases with system size.

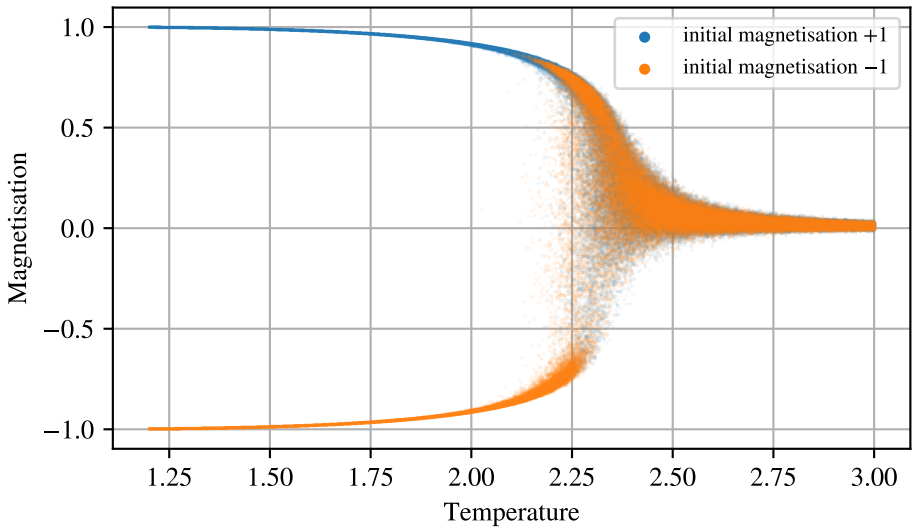


Figure 8: Almost same calculation as on fig. 3. Here the external filed is $H = 0.003$. Blue and orange points are results of simulation which started with all spins up (+1) and down (-1) respectively. Note that with the external field this becomes a first-order phase transition.

than some critical value, magnetisation jumps to +1. This can be seen from hysteresis curves on fig. 7. The same effect can be seen at static magnetic fields (see fig. 8). Note that e.g. $T < 2.25$ system started from magnetisation -1 remains there despite the external field and energetically favourable state with magnetisation +1. Again this can be explained by insufficient fluctuations to overcome the energetic barrier between the states. Because of the external field, this jump occurs at a lower temperature than T_c and phase transition becomes first-order in nature which can be clearly seen by comparing figures 3a and 8.

8 Conclusion

In this work, we have studied the out-of-equilibrium dynamics and correlation times, the critical behaviour and the effects of the external field. The relaxation time of the system increases faster the exponential close to the critical points. Theoretically, at the critical point, decorrelation length diverges to infinity, however, it is limited by the size of the lattice. And generally, the quantities that expected to diverge at the critical point in infinite lattice limit, are expected to be finite for finite-size lattice.

Finite-size scaling relations provide the behaviour of such quantities close to critical point and show how they scale with system size [12, 13]. E.g. the magnetisation $M \sim N^{-1/\nu}$ (eq. 6.2) for which we found $\nu = 1.110 \pm 0.004$ in case of free boundary conditions and $\nu = 1.198 \pm 0.018$ for periodic boundary conditions. ν known as a critical exponent. Interestingly, proportionality constant of (6.2) has different sign for different boundary conditions (fig. 5). We have not given complete explanation for this phenomenon.

In the study of the average magnetization, it turned out that in significant cases the system converged to a metastable state. Mean-field methods do not reveal such metastable states. We have rationalized their existence and estimated the distribution of these states.

Fluctuation-dissipation theorems (FDT) provide efficient way to calculate e.g. magnetic susceptibility (7.1) or heat capacity (6.1) [6]. In section 7 we have calculated the former using explicit method and demonstrated the efficiency of FDTs.

In presence of the external field, we have shown the ferromagnetic behaviour of the system: Below phase transition temperature, even at slow varying fields (compared to relaxation time), the system shows hysteresis effect, i.e. magnetisation is stable against small external fields. Additionally, first-order phase transition was demonstrated.

In summary, the numerical results are well fitted to the known analytical results and behaviour qualitatively consistent with approximate solutions. There remained some open questions about the scaling of heat capacity with system size, which requires further explanation. The simulation engine was written on one of the fastest programming languages. There are other, more efficient simulation algorithms, such as the Wolff cluster algorithm [14], which requires fewer iterations to converge to the stationary state. However, MH algorithm is much more related to statistical physics and it is easier to make connections to the theory.

References

[1] L. Onsager, "Crystal statistics. i. a two-dimensional model with an order-disorder transition," *Physical Review*, vol. 65, no. 3-4, p. 117, 1944.

[2] G. Grinstein and D. Mukamel, "Exact solution of a one-dimensional ising model in a random magnetic field," *Physical Review B*, vol. 27, no. 7, p. 4503, 1983.

[3] R. J. Glauber, "Time-dependent statistics of the ising model," *Journal of mathematical physics*, vol. 4, no. 2, pp. 294–307, 1963.

[4] S. Heims, "Master equation for ising model," *Physical Review*, vol. 138, no. 2A, p. A587, 1965.

[5] P. A. Martin, "On the stochastic dynamics of ising models," *Journal of Statistical Physics*, vol. 16, no. 2, pp. 149–168, 1977.

[6] L. Landau and E. Lifshitz, *Statistical physics, part 1, Volume 5 (Course of Theoretical Physics, Volume 5)*.

[7] W. K. Hastings, "Monte carlo sampling methods using markov chains and their applications," 1970.

[8] C. Robert and G. Casella, *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

[9] W. Jakob, J. Rhinelander, and D. Moldovan, "pybind11 — seamless operability between c++11 and python," 2016. https://github.com/pybind/pybind11.

[10] A. Matevosyan, "Ising-model." https://github.com/ashmat98/ising-model, 2021.

[11] A. Matevosyan, "Magnetisation at different temperatures. video demonstration (25 sec).." https://github.com/ashmat98/ising-model/blob/main/magnetisation.mp4, 2021.

[12] V. Privman, *Finite size scaling and numerical simulation of statistical systems*. World Scientific, 1990.

[13] K. Binder, "Applications of monte carlo methods to statistical physics," *Reports on Progress in Physics*, vol. 60, no. 5, p. 487, 1997.

[14] U. Wolff, "Collective monte carlo updating for spin systems," *Physical Review Letters*, vol. 62, no. 4, p. 361, 1989.

Appendices

A `README.md` from repository

See next page.

Ising-model

This engine runs Metropolis-Hasting algorithm on 2D Ising model. It provides base framework for the extension to more complex models with extended interactions. At this stage, only 4-neighbour interactions are accounted.

The project structure is:

```
ising-model
 ├ cpp                      /* C++ part, simulation engine */
 │ ├ base.cpp
 │ ├ base.h                 /* base class of the engine */
 │ ├ CMakeLists.txt         /* configuration for CMake */
 │ ├ ising-model.cpp        /* Language bindings using PyBind11 */
 │ ├ main.cpp               /* sample usage of the engine inside C++ */
 │ ├ metropolis-hasting.cpp
 │ └ metropolis-hasting.h   /* Metropolis-Hasting algorithm */
 ├ python
 │ ├ dumps                  /* simulated data, which is not uploaded to
 │ │                           the github */
 │ │ └ ...
 │ ├ figs                   /* generated figures */
 │ │ └ ...
 │ ├ Results.ipynb          /* results and code for images */
 │ ├ ****.ipynb             /* other jupyter notebooks for running
 │ │                        simulator for specific task */
 │ ├ parallel.py            /* functions that run the engine for specific
 │ │                           task, can be run in paralel. */
 │ ├ style.py               /* style for plots */
 │ └ utils.py               /* utility functions */
 ├ report.pdf               /* The report on this project */
 ├ CMakeLists.txt           /* configuration for CMake */
 └ README.md                /* This file */
```

Compilation

1. Install conda environment for convenience.
2. Run these commands in the shell

```
$ conda install pybind11 cmake
$ git clone ...
$ cd ising-model
$ mkdir build
$ cd build
$ cmake cmake -DCMAKE_BUILD_TYPE=Release ..
$ make
```

3. This will create the python module. E.g. on Windows something like `ising_model.cp38-win_amd64.pyd` or e.g. `ising_model.cpython-37m-x86_64-linux-gnu.so` on Linux. This can be imported to Python (see below).

Usage

Sample usage in `C++` ```c++ SimulateMH engine(32, 32, 1, 1); engine.set_T(10); engine.make_steps(stps); auto E = engine.get_sampled_E(); auto M = engine.get_sampled_M(); etc.

Sample usage in `Python`:

```python
from ising_model import SimulateMH
BC = SimulateMH.BoundaryCondition
engine = SimulateMH(Nr=32, Mc=32, frequency_to_store=1, H=1, omega=0,
                    bc=BC.Periodic, SEED=42)
engine.random_init()
# or engine.constant_init()
engine.make_steps(10**8) # ~10 seconds runtime
M = engine.get_sampled_M()
E = engine.get_sampled_E()
etc.
```

Paralelisation usnig `multiprocessing.pool`:

```
 from multiprocessing import Pool
import numpy as np
from parallel import to_run, BC

pool = Pool(4)
Ts = np.linspace(0,4,100)

runs = len(Ts)

res = pool.starmap(to_run,
    zip(*(np.arange(runs),
          [10_000_000]*runs, # steps
          Ts, # temperature
          [32]*runs, # lattice size Nr
          [32]*runs, # lattice size Nc
          [1]*runs, # frequency
          np.random.randint(0,10**8, runs), # seed
          [BC.Periodic]*runs, # bc
          [False]*runs, # return_engine
          ["random"]*runs, # init
          [0]*runs, # H field
          [0]*runs # H omega
          ))
)
```

So the above code will run 100 independent simulatins with given parameters (in this case the temperature changes).