Worksheet 6: Finite-Size Scaling and the Ising Model

Patrick Kreissl Kai Szuttor

January 21, 2020

Institute for Computational Physics, University of Stuttgart

Contents

1 General Remarks

- Deadline for the report is **Tuesday, February 4th, 2020**
- On this worksheet, you can achieve a maximum of 20 points.
- The report should be written as though it would be read by a fellow student who attends to the lecture, but does not do the tutorials.
- To hand in your report, send it to your tutor via email
  - Patrick (`pkreissl@icp.uni-stuttgart.de`)

- Kai (`kai@icp.uni-stuttgart.de`)

- Please attach the report to the email and use PDF format for the report (we will *not* accept MS Word doc/docx files!). Include graphs and images in the report.
- If the task is to write a program, please attach the source code of the program, so that we can test it ourselves.
- The report should be 5–10 pages long. We recommend using LaTeX. A good template for a report is available online.
- The worksheets are to be solved in groups of two or three people.

On this worksheet, you will combine all the methods and skills that you have obtained during the term and use them to compute the critical temperature T_c and the critical exponents β_m and ν of the two-dimensional Ising model.

All files required for this tutorial can be found in the archive `templates.tar.gz` which can be downloaded from the lecture's homepage.

As on the previous worksheet, we will perform simulations of the two-dimensional Ising model on a $(L \times L)$ square lattice. $\sigma_{i,j}$ denotes the spin at lattice position (i, j).

The (total) energy of the system is defined by

$$E = \frac{1}{2} \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} E_{i,j} \tag{1}$$

where

$$E_{i,j} = -\sigma_{i,j}(\sigma_{i-1,j} + \sigma_{i+1,j} + \sigma_{i,j-1} + \sigma_{i,j+1}) \tag{2}$$

The system uses periodic boundary conditions, *i.e.*

$$\sigma_{-1,j} = \sigma_{L-1,j} \qquad \sigma_{L,j} = \sigma_{0,j}$$
$$\sigma_{i,-1} = \sigma_{i,L-1} \qquad \sigma_{i,L} = \sigma_{i,0}$$

As an observable, we are interested in the (mean) *energy per site* e, which is defined by

$$e = \left\langle \frac{E}{L^2} \right\rangle$$

and the (mean) *magnetization* m

$$m = \langle |\mu| \rangle \tag{3}$$

where

$$\mu = \frac{1}{L^2} \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} \sigma_{i,j} \tag{4}$$

Magnetization in the Ising model on an infinite lattice is unity at $T = 0$, decreases with increasing temperature and vanishes in a first-order phase transition at critical temperature T_c. At this temperature, the correlation length ξ diverges. This means that fluctuations on all length scales exist close to the critical point and thermodynamic observables diverge following a power-law dependence, for example of the type

$$\xi \sim t^{-\nu}, \qquad m \sim t^{-\beta_m}, \qquad t \equiv |1 - T/T_c| \,. \tag{5}$$

For $2D$, the values of T_c and the critical exponents β_m and ν are known analytically as well as the critical temperature:

$$\beta_\mathrm{c} = \frac{J}{k_\mathrm{B} T_c} = \frac{\ln(1+\sqrt{2})}{2}, \tag{6}$$

Note that for historical reasons the symbol β is used both for the inverse temperature and the critical exponent for magnetization. To clearly distinguish between them, in this worksheet, we will use β_m for the critical exponent for magnetization.

Simulations can only be performed on finite lattices, usually with periodic boundary conditions. By construction, these cannot handle infinite-ranged correlations close to the critical point. Therefore, the results obtained from finite-lattice simulations will systematically differ from the infinite limit.

2 Speeding up the Simulation

The Ising Monte-Carlo simulation from the last worksheet is written in pure Python. Unfortunately, when the size of the lattice L grows, the performance of the simulation is too low. To speed up the simulation, the calculation needs to be implemented in C++. We make use of Cython to provide a Python interface to the C++ implementation. All you need to do to compile the C++ source code and create the python module `cising` is executing the bash script `build.sh`.

Task (3 points)

- The template contains a skeleton of the Simulation code. Insert the missing pieces in `ising.hpp`.
- Explain, how a frequent recalculation of the energy and magnetization is avoided and at what point a full recalculation is still necessary.

Hints

- Use `ising_test.py` from the template to verify your implementation. It is necessary to write such tests for simulation codes. "If it isn't tested, it doesn't work".
- In some cases, good programming practice has been omitted in favour of simplicity. For instance, methods that do not change the state of an object should be declared `const` in C++ and the value type of the Ising model should be a C++ template parameter.

3 Determining Equilibrium Values and Errors

In this task, your job is to perform simulations of the two-dimensional Ising model for different lattice sizes L and measure the equilibrium values of the magnetization and the energy and their errors.

Task (3 points)

- Run simulations with $L \in \{16, 64\}$ for $T \in \{1.0, 1.1, 1.2, \ldots, 4.9, 5.0\}$.
- Determine equilibrium values and errors of m and e for all of these runs. Keep in mind that consecutive measurements may be correlated.
- Plot m and e (with error-bars) as a function of T for the different system sizes.
- Add the exact curve for $L = 4$ from the last worksheet.
- Take a look at `https://en.wikipedia.org/wiki/Square-lattice_Ising_model`. Plot the analytic result for the magnetization of an infinite lattice along your results.
- How do the curves depend on L?

Hints

- If you want to store simulation data to a file, remember the module `pickle`.
- To reduce the file size, you can use the module `gzip`. Open the file with `gzip.open`, then all following operations will work on a compressed file.

4 Finite Size Scaling

4.1 Determining T_c

Task (6 points)

- Implement the Binder parameter $U = 1 - \frac{1}{3}\frac{\langle \mu^4 \rangle}{\langle \mu^2 \rangle^2}$.
- Measure the Binder parameter for $L \in \{4, 16, 32\}$ and $T \in \{2.0, 2.02, 2.04 \ldots, 2.38, 2.4\}$.
- Plot U vs. T for the different values of L, the should all cross approx. in one point.
- For each possible pair of L values find the point of intersection ($b = L_1/L_2$, T^*). This should be done programmatically, not via ruler and pencil.
- Determine T_c to a precision of ± 0.02 by averaging T^*.

Hints

- Note that you need a pretty good accuracy of U to determine T_c. This may take quite a while (approx. an hour depending on your hardware),
- For interpolating the discrete data you can use `scipy.interpolate.interp1d`.

4.2 Estimating β_m

Task (4 points)

- Perform simulations at T_c with different $L \in \{8, 16, 32, 64, 128\}$
- Plot m against L in double logarithmic scale.
- From the theory, what is the scaling law of the magnetization, *i.e.* how does the magnetization depend on L?
- Try to estimate β_m from the curve, given that $\nu = -1$ for a two-dimensional Ising system.

4.3 The Master Curve

Note that the value of β_m determined in the previous task depends strongly on the value of T_c, so do not trust the value overmuch. In the following, you will learn about a method that is better suited to determine the value.

When plotting $ML^{\beta_m/\nu}$ against $tL^{-\nu}$, for different values of L and $t = |1-T/T_c|$ (*reduced temperature*), all data should fall on a single *master curve*. To get a better estimate for β_m, one can use the following procedure: make plots of mL^a against $tL^{-\nu}$ with different values of a and select one where all data points are on the same curve.

Task (4 points)

- Use all of your data from this worksheet and plot the master curve for the estimated value of β_m from the last task. Scale the x-axis to the range $[-20, 20]$.
- Try different values of β_m to plot the master curve. Provide the best-looking plot and your estimated value of β_m.

Hints The fit should be best in the core part of the curve, *i.e.* where $ML^{\beta_m/\nu} \approx 1.0$ and $tL^{-\nu} \approx 0.0$.

5 Bonus Task: Calculating the Magnetic Susceptibility from Fluctuating Magnetization

The magnetic susceptibility of a system can be obtained via

$$\chi = \left.\frac{\partial m}{\partial B}\right|_{B=0}. \tag{7}$$

It can also be calculated from fluctuations in the system's magnetization in the canonical ensemble as

$$\chi = \frac{\langle \mu^2 \rangle - \langle \mu \rangle^2}{k_\mathrm{B} T}. \tag{8}$$

Task (5 points)

- Derive equation 8. To do so, it is advisable to have μ defined by simple summation over all spins – **not** as a sum over lattice sites as given in equation 4.

- Magnetic susceptibility can also be computed by differentiating the magnetization with respect to the magnetic flux density, see equation 7. Plot χ using equation 8 and by differentiation method. Compare both and discuss advantages or disadvantages of both the approaches.