SI2530 Computational Physics: Monte Carlo Simulation of the Ising model

Fall 2020

1. Brief instructions

All material required for the lab is in the file `lab03_ising_2020.tar`, available from the course home page [1]. Download and then extract it with the command `tar -xvf lab03_ising_2020.tar`.

- Compile the C++ program `ising2d` by typing `make`.
- Run the program with the shell script `run.sh`. Edit the script to set parameters for the program and the file where it should save the simulation data. Note that the script will overwrite old data files when running – *take care not to lose your old data*! Review the general lab advice handout for tips on keeping track of files.

To analyze the simulation data from `ising2d`, two tools have been provided:

1. A shell script `analyze.sh` which is useful to quickly plot a selected variable versus temperature from several simulations. To use it, run `./analyze.sh COLUMN FILE1 FILE2 ...`, where `COLUMN` is the column *number* (see table 1) to plot and `FILE1` (etc.) are the simulation data files to plot the data from.

   For example, to plot the absolute magnetization versus temperature in files `data4` and `data8`, type `./analyze.sh 3 data4 data8`.

2. A Matlab script `analyze_adv.m` which has some additional computational functionality which is useful during the latter half of the lab. Instead of Matlab, which is commercial software, we will use the open source alternative Octave. The syntax of Octave is essentially equivalent to that of Matlab. To run a script using Octave you can either start the program first by typing `octave` and launch the script from its shell, or you can launch them both in one go by typing `octave --persist -q <scriptname>`.

   Edit the script to change which variable to plot, or to calculate new ones.

   To print the figure, uncomment the line that says `print...` in the script.

To plot the analytical (2D) solution alongside your data, edit either script and uncomment the required lines inside.

Finally, a Matlab script `ising.m` has been provided to show the Ising model in action on a 2D lattice. This should also be run using Octave as per the instructions above.

[1] `https://canvas.kth.se/courses/20050/files`

Column	Symbol	Description
1	T	Temperature
2	m	Average magnetization
3	$\bar{m}$	Average absolute magnetization
4	χ	Susceptibility (chi)
5	e	Average energy per spin
6	c	Heat capacity
7	g	Binder cumulant
8	L	System width (total size is $L \times L$)
9	-	Seed used for the random number generator

Table 1: The thermal averages of parameters output from the program into the different columns.

2. The 2D Ising model

2.1. Overview

The two-dimensional Ising model is a good test case for Monte Carlo simulations in statistical physics, since it is both exactly solvable and quite nontrivial. The analytical solution of the two-dimensional Ising model is very complicated, and the three-dimensional Ising model is still unsolved exactly. On the other hand, the models are highly suitable for simulation, and the solution of the Ising model by simulation is quite straightforward. In this lab you will simulate the Ising phase transition and compare your data with the exact solution.

The Ising model is defined by the Hamiltonian

$$H = -J \sum_{\langle i,j \rangle} S_i S_j - h \sum_i S_i \,, \tag{1}$$

where $S_i = \pm 1$ is a spin degree of freedom on lattice site i, J is a coupling constant (from now on we will set $J = 1$, which means ferromagnetic coupling – $J = -1$ would mean anti-ferromagnetic modeling), $\langle i, j \rangle$ denotes nearest neightbour summation and h is an applied magnetic field. The thermal properties are defined by the partition function

$$Z = \sum_{\{S_i = \pm 1\}} \exp\left(-H/T\right)$$

where T is the reduced temperature (Boltzmann's constant k_B has been set to 1).

The total magnetization of a system with N spins is defined as

$$M = \sum_{i=1}^{N} S_i \,.$$

2.2. The simulation program

The provided program `ising2d` goes through an input temperature interval $T_{\min}$ to $T_{\max}$ in intervals of size ΔT. At each temperature step it runs a warmup equilibration of N_{warmup} steps and then samples the Ising model over N_{sample} additional steps. It then computes the thermal expectation values at the current temperature for some parameters and outputs the data in columns. See table 1 for the order of columns.

Some of the parameters that the program outputs are:

$$
\begin{aligned}
m &= \langle M \rangle N^{-1} && \text{average magnetization per spin} \\
\bar{m} &= \langle |M| \rangle N^{-1} && \text{mean absolute magnetization} \\
e &= \langle H \rangle N^{-1} && \text{energy per spin} \\
\chi &= \frac{\partial m}{\partial h} = \left(\left\langle M^2 \right\rangle - \langle M \rangle^2 \right) N^{-1} T^{-1} && \text{susceptibility} \\
c &= \frac{\partial e}{\partial T} = \left(\left\langle H^2 \right\rangle - \langle H \rangle^2 \right) N^{-1} T^{-2} && \text{heat capacity}
\end{aligned}
$$

N here refers to the number of samples N_{sample}. Later in the lab we also describe and use the binder cumulant g.

Q1: Derive the above formulas for χ and c.

`ising2d` is a Monte Carlo simulator. At each update step it randomly selects a point on the lattice and tries to flip its spin. It then calculates the change made to the Hamiltonian by that flip: If the change is favorable, ie. the resulting energy is lower than previously, the flip is accepted. If the energy is higher, the program calculates a transition probability w based on how much the energy increases. It then generates a number and tests against the probability. If the result is positive the flip is accepted, otherwise it is rejected.

3. Running the 2D simulation

3.1. Sampling the Ising model

Use the script `run.sh` to run simulations for various system widths, e. g. $L = 4$, 6, 8, 10, 12, 16 or 20. The system width and output file is set by editing the script. To run the script, type `./run.sh`.

If you have the time, increase the number of sampling steps N_{sample} to 50000 and see how it impacts your data.

Plot your results and the exact solutions using either of the supplied analysis scripts described above.

Q2: Comment on the convergence of the simulation data towards the exact curves. Why do the simulation results show a finite magnetization also above the transition?

Q3: How accurately can one deduce the critical point T_c from the MC data in this plot? What goes wrong with m and why does $\bar{m}$ look better?

3.2. Finite size scaling analysis

We will now study a few different ways to use scaling to extrapolate results for a finite system size $L \times L$ to the thermodynamic limit, when $L \to \infty$. This will allow for an estimation of e. g. the critical temperature and the critical exponents.

3.2.1 Making a scaling ansatz

Close to a critical point, thermodynamic quantities are expected to obey scaling laws. For the critical point of the Ising model we start by making a finite size scaling ansatz in the absence of external fields ($h = 0$):

$$m(T, L) = L^{-\beta/\nu}\, \tilde{m}(L^{1/\nu}\,(T - T_c))\,.$$

Here $\tilde{m}$ is some unknown scaling function, and β, ν are critical exponents defined by

$$m \sim |T - T_c|^{\beta}\,, \quad \xi \sim |T - T_c|^{-\nu} \quad \text{as} \quad |T - T_c| \to 0\,.$$

In general one can now make a multi-parameter fit of the MC data to the scaling ansatz, but it is usually easier to follow the steps described below.

3.2.2 Determining the critical point

Let us first assume knowledge of the exact results $\nu = 1$, $\beta = 1/8$ and observe that $\tilde{m}$ is independent of L at $T = T_c$. Hence a plot of $mL^{1/8}$ versus T for different L should give curves which intersect at $T = T_c$.

Q4: Plot $mL^{1/8}$ versus T for the data from your simulations [2] and determine T_c from their intersection. Compare it to the exact value $T_c = 2/\ln(1 + \sqrt{2}) \approx 2.269$.

3.2.3 Determining the critical exponent

In general we do not know the values of β and ν. These are rather the things we want to calculate in the simulation. A useful quantity in these cases is the so called Binder cumulant

$$g = 1 - \frac{\langle m^4 \rangle}{3\langle m^2 \rangle^2} = \tilde{g}(L^{1/\nu}\,(T - T_c))\,,$$

where $\tilde{g}$ is some unknown scaling function. When g is plotted as a function of T, data curves for different sizes L will intersect at T_c. Why?

Q5: Plot the binder cumulant for some system sizes L and see if the curves intersect at the critical point T_c. Explain why the number 3 is used in the denominator (hint: for $T \to \infty$ the spins fluctuate independent of each other).

Q6: Determine ν by plotting g versus $L^{1/\nu}(T - T_c)$ for different choices of ν. The best value for ν is the one that makes the data curves for different sizes and temperatures collapse onto a common curve, that represents the universal scaling function $y = \tilde{g}(x)$. Discuss the obtained accuracy.

3.3. Adding an external magnetic field

Modify the program source in file `ising2d.cpp` to make it handle the case for $h = 1$. This term is added to the change in the Hamiltonian when a spin flip is attempted (see equation 1). Don't forget to recompile by typing `make`. Run simulations for a few different system sizes and see how the added magnetic field impacts the magnetization.

[2] For this task, it is suggested to use the Matlab script `analyze_adv.m`.

4. Monte Carlo dynamics

Run the Matlab script `ising.m` at different temperatures below, at, and under the critical temperature T_c, e. g. $T = 0.5\,T_c$, $1\,T_c$ and $3\,T_c$. The temperature is changed inside the script. This program shows the Ising model in action on a 2D lattice. Study the configurations generated by the simulation in real time. Run a few times for each temperature.

Q7: Discuss the behaviour of the generated configurations at different temperatures. Describe the difference between the configurations at various temperatures and pay special attention the size and lifetime of domains.

Q8: At the phase transition the correlation length ξ and correlation time τ are diverging. Can you observe this by studying the generated configurations?

5. Simulating the 3D Ising model

Modify `ising2d.cpp` to do 3 dimensions. You will have to:

1. Add an extra dimension to the lattice.
2. Modify the Hamiltonian change when a spin is flipped to account for the additional lattice neighbours.
3. Update the maximum allowed energy change to account for the additional neighbours that each point interacts with.

All lines that need to be modified are pointed out in the code. You just need to figure out how. Make sure to turn off the external magnetic field (set $h = 0$ again) and don't forget to recompile by typing `make`.

The critical temperature for the 3D Ising model is approximately known from MC simulations to be $T_c \approx 4.51142$. Note that T_c and the values of the critical exponents depend on the dimensionality of the system, and are thus not equal for the two- and three-dimensional models.

Run the 3D Ising model for a few selected system widths L. The 3D Ising model is computationally much more expensive than the 2D model, so don't start too long or large simulations.

Q9: Repeat the finite size scaling analysis (Questions 4–6) for the 3D model. Since the critical exponents are not given you have to change your approach slightly to find a way to determine them.

Q10: Try to figure out an argument to explain why T_c increases when you go from two to three dimensions.

6. Write a report

Write a short report ($\sim$ 5 pages, including necessary figures) that describes what you have done and includes answers to all questions. Convert all numberts into suitable SI units. You may write in pairs. Take the opportunity to ask for help if you get stuck on the theoretical parts. Send your report (preferably as a PDF document) by e-mail to `anatoly@kth.se`.