Renormalization - cluster algorithm

Kürti Zoltán

1 Implementing and testing the cluster algorithm

I implemented the cluster algorithm as described during the lectures in C++. To store the state of spins I used a vector of bools because the C++ specification guarantees that vector<bool> uses one bit per element. For larger L values this seemed to improve the speed of the calculation by about 10% on my machine compare to using an entire byte per spin. Finding the neighbors of a spin are solved using integer division and remainders after division by L. The entire project can be found at the same link as the previous homework: `https://github.com/KurtiZoltan/Renorm`.

For each simulation I prepared the system in the $m = 1$ state. To get near the equilibrium states I calculated the magnetization of the system before and after ten steps of the cluster algorithm. I repeated these steps until the absolute value of the magnetization increased after ten steps of the cluster algorithm. The idea is that until the system is considerably colder than the β parameter provided to the algorithm it is very likely that the absolute value of the magnetization will decrease after ten steps.

For validation I calculated the expectation value of $|m|$ for $L = 512$ and $\beta = 0.5$ using 32768 simulated magnetization values. The result was $m = 0.911315$ which confirmed the validity of my cluster algorithm implementation.

2 The distribution of magnetization values

The goal was to plot the distribution of magnetization values as the system goes through the critical point. The simulation is obviously finite so instead of going through the critical point the best approximation is to go through the susceptibility maximum. For $L = 128$ using 1024^2 magnetization values and 50 bins the result is figure 1.

As mentioned during the lectures, in the disordered phase the distribution of m values is centered on 0 with a single peak. Near the critical point this distribution widens and the second derivative of this distribution at 0 becomes negative. Deeper in the ordered phase the peaks of the distribution clearly separate. For larger L values the peaks would have been narrower.

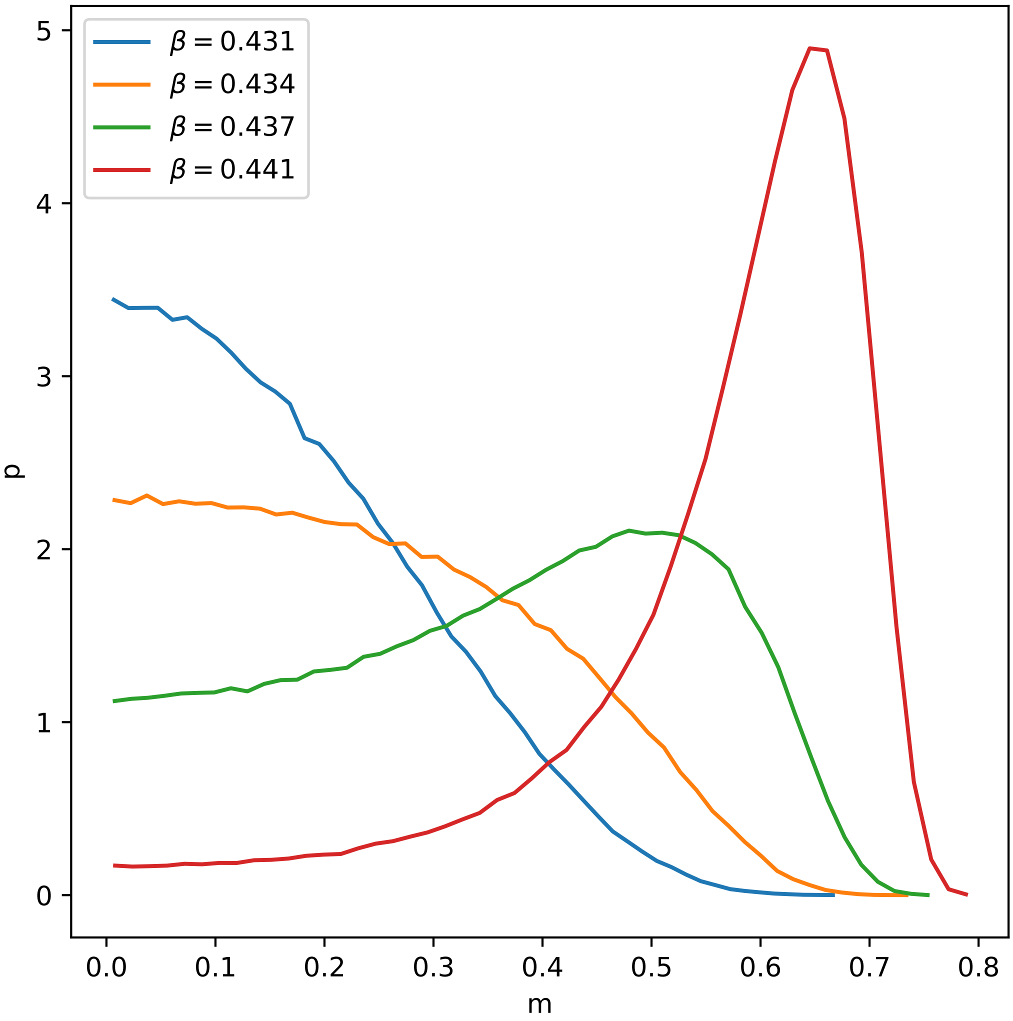

Figure 1: The estimated probability density of the magnetization for a system of size $L = 128$ near the susceptibility maximum.

3 Maxima of the susceptibility curves

For given L and β values the susceptibility was calculated based on 32768 magnetization measurements. The magnetization has a probability distribution and the magnetic susceptibility is connected to the standard deviation of this distribution. The naiv formula for empirical standard deviation is maybe only valid for independent random variables.

The measurements of magnetization are still correlated when using the cluster algorithm, I used blocking to estimate the standard deviation. (See src/cluster.cpp block() function, based on my tests for an uncorrelated set of data it keeps the variance unchanged.)

The L values used for the simulations ran from 32 to 160 with steps of 16. The β values for the initial scan were between 0.41 and 0.45. After the initial scan I selected all β values where the measured susceptibility was at least half of the maximum measured susceptibility. The minimum and maximum of these β values determined the new region of interest. I choose this region of interest to be 1.5 times the size of the interval between the minimum and maximum. I calculated susceptibility values in this region of interest and repeated the process. In total I calculated the susceptibility 200 times for each L size with this method and an additional 50 points with a similar algorithm but instead of using 32768 magnetization values I used 262144. This was to check if the length of each simulation was sufficiently long. Looking at the data by eye the results weren't different so I concluded that the length of the simulation was sufficient. The variance in the susceptibility continues to decrease as the simulation length increases, I was interested in systematic errors during this test.

In total I had 250 β and χ pairs for each L value. I fit a parabola to the peaks to determine χ_{max} and β_{max}. The result is figure 2.

4 Critical exponents

The final task is to estimate the value of critical exponents based on $\chi_{\text{max}}(L)$ and $\beta_{\text{max}}(L)$. Fitting the expected form on the data I got figure 3. From $\beta_{\text{max}}(L)$ ν can be extracted, and according to this simulation $\nu = 0.86$. Its exact value for the 2d Ising model is $\nu = 1$. From $\chi_{\text{max}}(L)$ the $\frac{\gamma}{\nu}$ ratio can be extracted. Based on the fit $\gamma = 1.65$ its exact value according to the literature is 1.75.

The difference between calculated and exact values of critical exponents can be caused by multiple errors. First, the fitted functional forms only apply for large L. Second, and this I think is the biggest source of error, the calculated susceptibility values have errors. Second moments are harder to approximate with Monte Carlo methods, longer simulation lengths should help. So far these were general sources of error to every Monte Carlo simulation trying to calculate these critical exponents. I have an observation that is specific to my simulation, my β_{max} estimation isn't very robust. For example for $L = 96$ it seems to be a bit off. Decreasing the number of points I use for the parabola fit could help for

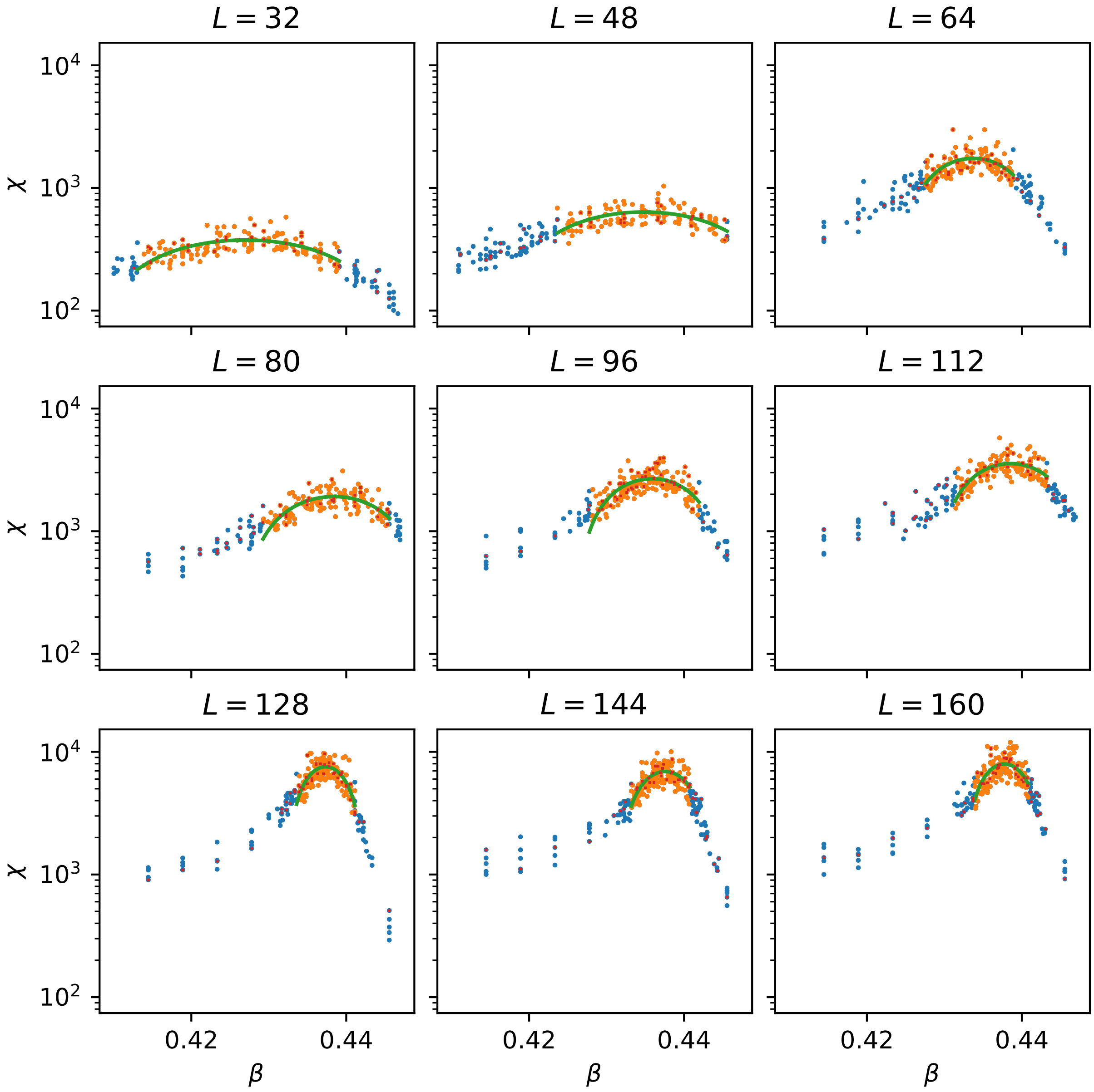

Figure 2: The temperature dependence of susceptibility for systems with different size. The orange dots represent the susceptibility measurements used to fit the parabola. The red dots are the control measurements calculated from 262144 magnetization values. The axis are the same for all sub figures.

that particular L but generally that would imply a shorter range in β used for fitting which would significantly worsen the estimation of linear and quadratic terms. Maybe there is a smarter choice of function to fit to get the argument of the maximum, or an entirely different process. Regardless, I would try to improve the estimation of $\beta_{\max}$ as

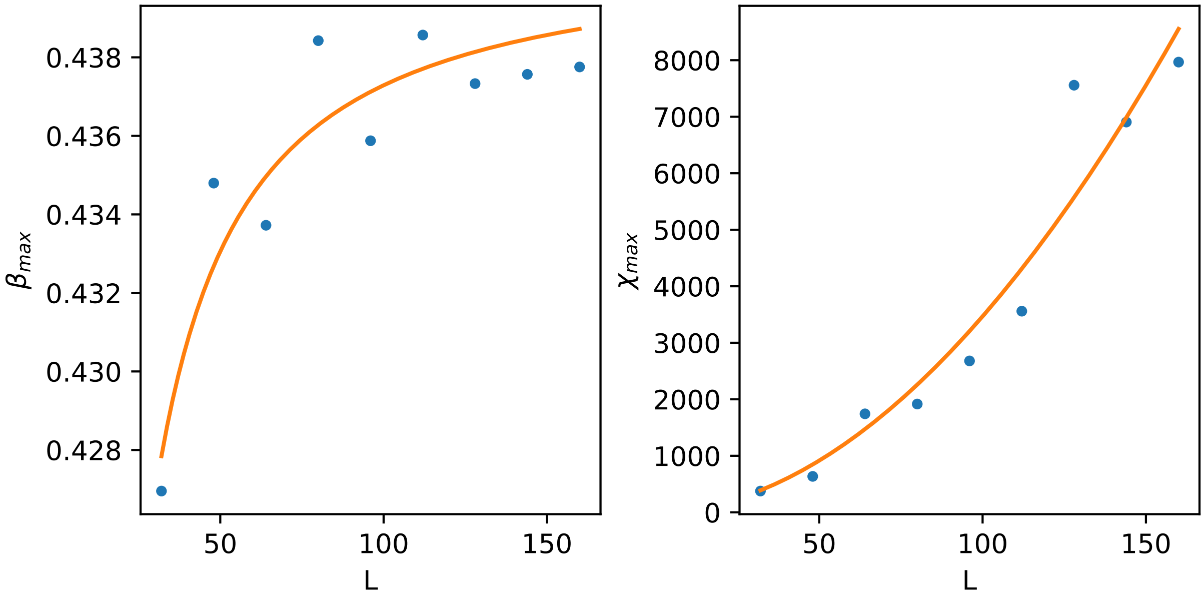

Figure 3: Fitting the theoretical large L dependence to the simulated data. $\beta_{\max}(L) = \beta_c - AL^{-\frac{1}{\nu}}$ and $\chi_{\max}(L) = BL^{\frac{\gamma}{\nu}}$ The fitted values are $A = 0.73$ $\nu = 0.86$ $B = 0.50$ $\gamma = 1.65$.

the next step. Also, I'm still not sure about the blocking part.