UMEÅ UNIVERSITY
Department of Physics
Peter Olsson

May 7, 2021
Monte Carlo, 7.5hp

Scaling analyses in critical phenomena

1 Data handling

In this lab you should improve the data handling in your simulations by storing data in some data files:

- Before the simulation is started (e.g. in `initialize_mc`) the simulation program should check if the data file is already there. If it is not the file should be created and the information needed for the calculations in the `summary` program should be put at the beginning of the file. (This is essentially the information in the `par` struct.)

  More information may be found in Chapter 8, "To organize large scale simulations".

- After each block is completed the simulation program should write the data associated with that block to the data file.

- You also have to write a `summary` program that reads the data files and writes the output in a way which is convenient for the plotting program. This program should take the file names from `stdin` and write results to `stdout`. This means that it should be possible to use it by writing

  ```
  ls data/256* | summary > L256.txt
  ```

  An partial program is found at the physics computer system, `/home/peol0002/MonteCarlo/summary.c` and is also available from a link on the course web page, `http://www.tp.umu.se/mc`.

2 Finite size scaling analysis

2.1 The magnetization

Do runs with the Wolff cluster update method for system sizes $L = 16$, 32, 64, 128, and 256. Set `nsamp=1000` (put it in the code for `CLU`) and run with at least `nblock=10`. Define $M = \sum_i s_i$, $m = \langle |M| \rangle / L^2$, and $t = T - T_c$. Choose the temperatures differently for different sizes such that you always have at least seven values with $-1 < tL^{1/\nu} < 1$. Plot four figures with different symbols for the different sizes:

1. m versus T (connect the points with lines)
2. $mL^{\beta/\nu}$ versus T, where you make use of the known values of β and ν from Table 4.1. (Again connect the points with lines.)
3. $mL^{\beta/\nu}$ versus $tL^{1/\nu}$ (no lines)
4. Zoom in figure 3 for $-0.25 < tL^{1/\nu} < 0.25$ and include error bars in the plot. Also draw a line from a second order polynomial through the data.

2.2 Binder's cumulant

Plot two figures with different symbols for the different system sizes:

1. Q_L versus T (with lines),
2. Q_L versus $tL^{1/\nu}$ (no lines).

3 For extra points

3.1 Spin-spin correlation (2p)

The most efficient way to determine the spin-spin correlation function,

$$g(x) = \langle s(x', y)s(x' + x, y) \rangle ,$$

is with a Fast Fourier Transform. Here we will instead do it with a simple raw calculation. To save time you should only examine a single randomly chosen row of spins in each measurement. Your program should do the following:

- Choose a row y by random.
- For each $x = 0, \ldots, L - 1$ accumulate
$$\sum_{x'} s(x', y)s(x' + x, y).$$
- Write the results from each block to a file.

Note that you now want different symbols for different temperatures.

Run simulations with $L = 256$ and $T = 2.16$, 2.20, 2.22, 2.24, 2.26, 2.2692, 2.28, 2.30, 2.32, and 2.36. At $T_c \approx 2.2692$ you should also run at $L = 1024$ and maybe also some bigger size (2048 and/or 4096).

First plot the raw data $g(x)$ vs. x for $L = 256$ and all the different temperatures. We will then have to plot data in the different regions, $T > T_c$, $T = T_c$, and $T < T_c$ in different ways:

3.1.1 For $T > T_c$

We could here determine ξ from $g(x)$ for the different temperatures and try to examine how ξ depends on $T - T_c$. We will instead do a quick test that just demonstrates that

$$\xi \sim \frac{1}{|T - T_c|},$$

by plotting $g(x)$ for different temperatures (on a log scale) versus $x/\xi \equiv x|T - T_c|$. The data should fall on straight lines with the same slope. (Cut off the data for large x; avoid plotting data which is only noise.)

3.1.2 For $T = T_c$

At T_c we expect $g(x)$ to decay algebraically. Plot $g(x)$ for $T = 2.2692$ versus x with log scale on both axes for two (or more) different sizes. Also draw a straight line with the expected slope through the data that is not affected by finite size effects.

3.1.3 For $T < T_c$

Below T_c $g(x)$ approaches a finite constant at large x. To extract the exponential decay, plot $g(x) - g(L/2)$ (on a log scale) versus x. Then make a second figure where the same data is plotted versus $x \times (T_c - T) \sim x/\xi$. Again, the data should fall on straight lines with the same slope.

3.1.4 The correlation length

Finally determine the slopes in the figures for $T < T_c$ and $T > T_c$, respectively. Use these slopes to determine the correlation length $\xi(T)$ (for $T \neq T_c$) and plot ξ versus T.

3.2 The 2D Ising model on a triangular lattice (2p)

The task is here to determine the critical temperature for the 2D Ising model on a triangular lattice. According to universality the critical exponents should be the same whereas quantities like T_c (of course) are different for different two-dimensional lattices.

Start from your program for the square lattice Ising model and change to a triangular lattice by adding links along one of the diagonals. Do simulations for a few different sizes and determine T_c with an uncertainity less than 0.001. To do this you need to get data with very good precision close to T_c. Avoid using too small sizes since finite size scaling is not quite OK for the smallest sizes. Very big systems, for which you don't have time to get good precision, are not good either.

A suggestion for getting a good determination of T_c is to do simulations with L = 16, 24, 32, 48, 64, 96, and 128 at several temperature close to T_c. One can then determine crossing points by focusing on data for L and $2L$ and determine $f(T, L) = y(T, 2L) - y(T, L)$, where y is a quantity that crosses at T_c, and for each $L = 16$ through 64 fit $f(T, L)$ to a second order polynomial and determine the temperature where this polynomial crosses zero. One should then find that these crossing points are about the same, with the possible exception of $L = 16$ which could be somewhat below (I think).