2. Computerübung, **Statistische Physik**

abzugeben am Donnerstag, 12.01.2012

Aufgabe C3 *Avalanches and hysteresis in condensed matter* (15 Punkte)

a) The basic model

Hysteresis in measurements of system properties is an interesting problem. While we investigated thermalization in assignment C2, this exercise will explore the opposite.

We initially imagine a fully magnetized ferromagnetic sample of condensed matter, described by an Ising-type next-neighbour-interacting spin lattice. As before, $S = \pm 1$ or ($\uparrow$ *or* $\downarrow$). We expose it to an adiabatically tuneable external field B. As a taste of a more realistic model, we assign an additional local field h_i to each spin, derived from the normal distribution

$$P(h) = \frac{1}{\sqrt{2\pi}R} e^{-h^2/2R^2},$$

where R and h are model parameters.

The energy function of the system thus looks like

$$\mathcal{H} = \sum_{i,j} J S_i S_j - \sum_i (B(t) + h_i) S_i.$$

The simulation process itself consist of a continuous increase in the external field strength until all of the spins are fully reversed. Occasionally, the flip of a spin at a certain field strength will induce one or more successive flips, that are part of an avalanche. Hence we want to explore some of the system properties.

Summary

- Create a 2-dimensional lattice datastructure of spins that are aligned in the down-direction. (Just like in C2 of the previous exercise.)
- Create a corresponding set of local fields that are normally distributed, with zero mean value.
- Set the external field B to a reasonable value that will sufficiently simulate the initial value $B \approx -\infty$.
- Simulate the adiabatic increase of the external field.
  1. Determine the unflipped spin with the least energy $\Delta\epsilon_{\text{min}}$.
  2. Set the external field B to -$\Delta\epsilon_{\text{min}}$.
  3. Flip that spin.
  4. Check for all neighbours, whether their energy is positive. If so, flip them as well and repeat step 4.
  5. Repeat steps 1-4 until all spins are reversed.

- Execute the procedure for values $R = \{1.4, 0.9, 0.7\}$ and $h = 1$.
- Determine the mean avalanche length and magnetization depending on the external field and display their temporal distribution graphically. Does the avalanche length depend on the parameter R or on the amount of unflipped spins?

Aufgabe C4 *Faster avalanches using priority queues* (9 Punkte)

The brute force way to search for the next spin to flip is very simple, yet, in a more complex lattice configuration, terribly slow. There are several more efficient ways to do it, including priority lists, stacks or smart ways of sorting the way the spin energy is checked. In this second part, you will enhance the model with a queued list of spin locations sorted by their respective energies. Finding the spin that is next to flip is way easier and updating the list is more efficient as there are less and less unflipped spins.

Summary

1. Create the spin lattice and the external fields h_i like before.
2. Define a list that points to the location of the next spin that would flip if it had n flipped neighbours. Initially, all elements of that array would logically point to the spin with the largest h_i.
3. From the number of spins pointed to by your array of possible next spins from step 2 (call it z), choose the spin with the largest value of the expression $\text{UP} - \text{DOWN} + h_i = 2\text{UP} - z + h_i$. Do not check spins that have fallen off the list by increase of the field. Note that by UP we mean the number of flipped spins (in $\uparrow$-direction) and by DOWN the number of unflipped spins next to the spin site the procedure operates on.
4. If the next spin on the list has exactly the right number of up-neighbours, flip it, remove it from the list and go back to step 3.
5. Move the pointer to the next spin on the sorted list. If you have fallen off the end, increment the external field $B(t)$ and start over at the top of the list.
6. Run this algorithm for the same parameters as in C3 and give an estimation, how much faster it performs.

Bonus (6 Punkte)

Vary the simulation over different values of h. Choose a fixed value of R and use one of the simulations C3 or C4 to check for average avalanche length, when h is not equal to one. What do you expect? Does the simulation concur? Try to find a parametrization that maximizes the avalanche length.

Formatting and delivery information

Though you are not confined to use a specific programming language, it is imperative that the source code you hand in is well documented and commented. In this manner we recommend the following languages:

- Java
- C/C++
- Fortran
- Mathematica

Of course you may use whatever high level language you may find useful, but nevertheless there are languages that are not really well suited for this type of assignment. Anyway, if we dont understand the way the code works and if it does not compile or run, you will not receive the maximal amount of points for that part of the assignment. For students not adept in higher level programming, Mathematica is a good choice, because the well-documented environment may speed up the overall learning curve.
You may either hand in your solution in exercise class, like usually, or via electronic mail to *fabian.transchel@itp.uni-hannover.de* with mat. number and your name. We prefer digital delivery of your solutions.

Problem assistance

Your homework tutors have basic knowledge about the problems and solution strategies. However, in case of more sophisticated questions refer to **Fabian Transchel**, room 013 of the quantum information group on the ITP ground floor. Consultation hours are Mon 14-16 and Wed 11-12:30. You may as well make an individual appointment, if you need further assistance.

Copying / Group work

We understand that the very nature of programming assignments give rise to the temptation of effortless copying your fellow student's source code. **While we appreciate teamwork, every student is required to hand in a personal version of solutions.** Be informed that we may find it useful to having you surveyed individually prior to the final allocation of homework points to check for authenticity of your work.