1. Computerübung, **Statistische Physik**

abzugeben am Donnerstag, 24.11.2011

Aufgabe C1 *Random number generation* (9 Punkte)

The generation and handling of pseudo random numbers is one of the basic tasks in numerical physics, nevertheless it seems useful to look at the process in more detail than calling an obscure built-in *random()* subroutine in the language of choice. In this first part of the sheet, you will design and implement a simple random number generator.
Generating random numbers seems easy. Instead of sampling from sources of (quantum) noise, one could simply hard code a lookup-table of predefined numbers that were determined randomly, e.g. through rolling a dice. However, the requirements of extensive numerical simulations will soon exeed any practical implementation of predetermined randomness. Thus, in essence, we not only have to create pseudo random numbers fast, but also on demand.
Since this is your first assigment regarding computational works, there is no need to demand innovative solutions and we will not judge the random number quality, because the evaluation of the Ising model in assigment two will allow for extensive testing of the quality of your pseudo randoms.

Summary

- Write an algorithm to create pseudo random numbers that does pass the following basic numerical tests:
- Check, whether your random number generator has a period. (How would you do that?) If it has (and it most likely does), it must not be smaller than 10^{10} samples, although good generators easily have better periods than 10^{50}.
- Do an incidence sampling test: Check whether all digits have equal probability up to a 5% variance. This is a very low standard, but will suffice to see interesting results in assignment two. Check for at least 10^5 samples.

Hints

- The R250 random number generator is a very prominent example of a simple and yet powerful algorithm that is widely used in Monte Carlo simulations in computational physics. When in doubt, implement this algorithm. However, use different initialization parameters and determine parameters for a period larger than 10^{15}.

Aufgabe C2 *The Ising model* (15 Punkte)

Sampling vast amounts of random numbers would be pointless if we do not plan to use them in a proper physical simulation. C2 deals with a simplistic

numerical simulation of the famous Ising model in two dimensions. In the two-dimensional Ising model, we consider a square lattice with 64x64 spins, that each have two internal states $S = \pm 1$ or ($\uparrow$ *or* $\downarrow$) respectively. A coupled pair (i, j) of neighboring spins contributes the interaction energy by $\epsilon_{ij} = -JS_iS_j$, where J is the coupling constant. If $J > 0$, the coupling is called ferromagnetic. (And anti-ferromagnetic, if $J < 0$.) We consider the simple case $J = 1$. (Also $k_B = 1$.) You can see that ϵ_{ij} will give 1 if the spins are parallel and -1 otherwise. Thus, the total energy of the system is

$$E_0(\mathcal{C}) = -J \sum_{(i,j)} S_i(\mathcal{C}) S_j(\mathcal{C}) = \sum_{(i,j)} \epsilon_{ij},$$

where $\mathcal{C}$ is the configuration of the spin lattice and (i, j) means the set of all neighbors in both directions. The boundaries are to be taken periodic. Before working on a dynamic simulation of the lattice, let us introduce the interesting part of the interaction: an external magnetic field. Let that field B be in $\uparrow$-direction. The energy formula now is given by

$$E(\mathcal{C}) = -J \sum_{(i,j)} S_i(\mathcal{C}) S_j(\mathcal{C}) - B \sum_i S_i(\mathcal{C}).$$

The first thing you want to do is to create a data structure that can store a configuration of spins. Once you established this array, fill it with randomly aligned spin values ± 1. We simulate the thermalization of the system using the Metropolis algorithm:

1. Pick a random spin site.
2. Calculate the energy $\Delta\epsilon$ for flipping the spin.
3. If $\Delta\epsilon < 0$, flip it. Else, flip with probability $e^{-\beta\Delta\epsilon}$
4. Repeat until the system is approximately thermalized. (Choose a reasonable threshold for the change of energy.)

Summary

- Implement the Metropolis algorithm on a 64x64 lattice.
- Thermalize the system for an ascending sequence of 100 external field values: $B = \{0, 0.02, ..., 2\}$. Use at least 10^2 Metropolis instances for each magnetization. What are the values of temperature and magnetization?
- How many cycles are necessary for the system to thermalize? Find a heuristic estimate.

Bonus (6 Punkte)

- Repeat C2 with real random numbers, for example obtained from the internet. Do the results differ from the previous ones?

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.