Becs-114.1100 Computational Science
Exercise round 10

https://mycourses.aalto.fi/course/view.php?id=4367

November 26, 2015

Problems

- Problem 1: 5 points
- Problem 2: 1 extra point *
- Problem 3: 3 extra points *

This is the last exercise round. You must obtain at least 15 points overall from the exercises to take the exam. Bonus for the exercise points is converted into exam points with some ratio.
*) The extra points are ONLY for passing the 15 points limit. They do not add to the exam points.

Problem 1: Ising model simulation

- The lecture material for lecture 10 (chapter 3) contains the description of the Ising model and the Monte Carlo simulation algorithm used here.
- For the initial computation of the energy, use: $Energy = -\sum s_i s_j$ with the sum being over all nearest-neighbour pairs (do not count $i = j$ or use the same pair twice). For updates of the energy during the Monte Carlo simulation, it's more efficient to update the previous value of the energy than to compute the whole sum since only a few terms in the sum will change. For magnetization, use $Magnetization = \sum s_i$, where the sum is over the sites (similarly, during simulation, it's easy to just update the term which changes). In result plots, divide these quantities by the lattice size ($L \times L$) to make them per site values.

Problem 1: Ising model simulation

- The equilibration time is estimated in order to discard this period from the beginning of the runs, so that the initial state would not have an effect on the measurements that are made (e.g., magnetization and energy). In this case, the equilibrium does not mean that the quantities themselves stay constant, but their averages remain unaltered within statistical fluctuations.
- Note that running long simulations (e.g., 50000 MCS) can take some time (especially when using Python). You might want to monitor the progress by printing the number of completed iterations at some intervals, so you have some idea how long the algorithm is going to run.

Problem 1: Ising model simulation

- In MyCourses, there is a simple 1D lattice implementation in Python and some ideas on how to use it for the Monte Carlo simulation. You don't have to use it, but it may be helpful. The lecture material contains more examples (in C).