Monte Carlo simulation methods **Homework 2, 18.10.2007**

Return the solutions (with program printouts) at the latest at the beginning of the 18.10. excercise session. You can also e-mail the solutions to Ahti Leppänen, <ahtilepp AT mail.student.oulu.fi>.

1. Let us consider a system of 2 real degrees of freedom, x and y, with a (dimensionless) energy function

$$H(x,y) = x^2 + y^2 + 5(x-y)^2 \qquad p(x,y) \propto e^{-H(x,y)}$$

   Write a Metropolis update algorithm for the system, i.e. update the variables x and y in turns, using the method described in page 42 of the notes. Measure $(x-y)^2$ and $(x+y)^2$ after every update step, and do at least ~ 10000 updates. What is the result for $\langle (x-y)^2 \rangle$ and $\langle (x+y)^2 \rangle$?

   For checking the results: it is easy to calculate the result analytically, using variables $a = x - y$, $b = x + y$.

2. RANDU is a notorious random number generator; it was the standard generator in IBM mainframes in 60's. It is defined by

$$I_i = (65539 I_{i-1}) \bmod 2^{31} \qquad x_i = I_i / 2^{31}$$

   Write a function which implements the generator, and a function which seeds it (naturally it can be the same function).

   Note: in practically all present-day computers the size of the integer variable is 32 bits. Thus, the multiplication in the generator would cause overflow. However, the result is modded by 2^{31}, i.e. only 31 low-order bits of the result remain.

   The easiest way to do the multiplication is to use `unsigned int` (at least in C or C++). If `irnd` is of type unsigned int, then the product `(65539U * irnd)` is quaranteed to have the 32 low-order bits correct. Then it is easy to mod the result by 2^{31}, i.e. the (integer) generator becomes

```
irnd = (65539U * irnd) % 2147483648U
```

   Another option is to use double precision floating point arithmetics, or the Schrage method described in the notes.

   a) Measure the cycle time of the generator (by brute force). Repeat it by using a couple of different seeds (i.e. values of I_0).

   b) Generate "2-tuples" $(x, y) = (x_{2i}, x_{2i+1})$ from successive pseudorandom numbers and plot these on a plane (dots, not connected by lines). Let us zoom in to the small sub-square $0 \le x \le 0.005$, $0 \le y \le 0.005$. Plot only those 2-tuples which fall within this square (but remember always generate numbers in pairs, and reject the pair if it is not within the square). Plot ~ 1000 points within the square.

      Do the test using seeds $I_0 = 1$, 4 and 32. What kind of pattern arises?