Basics of Monte Carlo simulations 2005. Exercise 1

To be handed in Mon 31.1, exercise session Thu 3.2. 10:15.

1. (4 p) Derive the probability

$$P_{hit} = \frac{2l}{\pi d}$$

for the Buffon's needle problem (see lecture notes).

If you want to have some fun, you can try the Buffon's needle java applet below http://stud2.tuwien.ac.at/∼e9527412/.

In exercises 2-5, we consider the following random number generators, given in Numerical Recipes second edition (whole book available in the web, www.nr.com): a) the lowest-order ($ia = 106, ic = 1283, im = 6075$) quick-and-dirty generator. b) The Park-Miller generator using the Schrage trick, but without the shuffle.

To make the results comparable, use 7 as the initial seed.

2. (6 p) Write a piece of code which calculates the repeat interval of the generators (by brute force i.e. going through the numbers before you get the original one back). Hint: you can use a "double" (C) or "double precision" (Fortran) variable to calculate the repeat interval; this data type almost always nowadays has a 15-digit mantissa, so it is enough to calculate the interval of any 32-bit integer (which only has up to 10 numbers).

3. (6 p) Generate a 2D distribution between (0-1 , 0-1) with the random-number generators, and plot them. Use 10000 points. Does the 2D distribution seem at least superficially OK?

4. (6 p) Generate again random points in 2D between 0 and 1, but print out only the ones lying below x=0.00001. Keep on doing this until you have obtained 10000 points, plot the result, and comment on it.

5. (6 p) Test of whether it is a good idea to attempt to improve on generators without much thought. Modify generator a) by changing the numbers a and c randomly to something which "feels" good to you. Then repeat the 2D test for the generator. Repeat this 5 times. Comment on the result.

6. (4 p) Look at the source code of the Mersenne twister (available on the course home page and the web). Find out what kind of algorithm is used to initialize it.

Hint: If you do not have a favourite graphing and fitting system, the program `xgraph` *available in the rock/science/tekno cluster in* `/usr/local/contrib/bin` *is enough to handle the graphical parts of this exercise.*