Computational Physics II

Prof. Ulrich Kleinekathöfer
Spring term 2021
Project 1, due February 24, 2021 at 11:55 pm
to be uploaded to https://moodle.jacobs-university.de

JACOBS UNIVERSITY

1a. Pseudo random number generator [50 points]

a) Write a random number generator for uniformly distributed numbers r_i between 0 and 1 by using the formula for a linear congruent generator:

$$I_{i+1} = (a \cdot I_i + c) \bmod m \,, \qquad r_i = I_i/m \,.$$

a, c and m are integer numbers. Choose a value for each of the three parameters which seems appropriate for a random number generator with a high periodicity, but be careful to avoid an overflow of the integer value range. (Wikipedia has a table of common choices, for instance, m = 2^{48} , a = 25214903917, c =11.)
b) Check the randomness of your linear congruent generator with $a = 57$, $c = 1$, $m = 256$, $r_1 = 1$ and of the Mersenne Twister generator implemented in Python via `random.random()`. To initialize a random sequence in Python, you need to plant a seed, or in Python say `random.seed(None)`, which seeds the generator with the system time.

1. Make a simple plot of r_i versus i which may not prove randomness, though it may disprove it as well as showing the range of numbers.

2. Make an x-y plot of $(x_i, y_i) = (r_{2i}, r_{2i+1})$. If your points have noticeable regularity, the sequence is not random. Random points should uniformly fill a square with no discernible pattern (a cloud).

3. A simple test of uniformity, though not randomness, evaluates the k-th moment of a distribution

$$\langle x^k \rangle = \frac{1}{N} \sum_{i=1}^{N} x^k \approx \frac{1}{k+1} + \mathcal{O}\left(\frac{1}{\sqrt{N}}\right)$$

where the approximate value is good for a continuous uniform distribution. If the deviation varies as $1/\sqrt{N}$, then you also know that the distribution is random since this assumes randomness. Make appropriate plots.

1b. Random walk in one dimension [50 points]

a) Suppose that the probability of a step to the right is $p = 0.8$. Numerically compute $\langle x \rangle$ and Δx^2 for different number of steps N = 4, 8, 16, 32, 64, 128 and 256. What is the qualitative dependence of Δx^2 on N?
b) An interesting property of random walks is the mean number $\langle D_N \rangle$ of distinct lattice sites visited during the course of an N step walk. Do a Monte Carlo simulation of $\langle D_N \rangle$ and determine its N dependence.

We can equally well consider either a large number of successive walks as in this problem or a large number of noninteracting walkers moving at the same time as in the next problem.

General remarks for all Projects

You will have to (i) analyze the problem, (ii) select an algorithm (if not specified), (iii) write a Python program, (iv) run the program, (v) visualize the data numerical data, and (vi) extract an answer to the physics question from the data.
Which checks did you perform to validate the code? State the results you got for these tests.
For each project you will submit a short report describing the physics problem, your way of attacking it, and the results you obtained. Provide the documented Python code in such a form that we can run the code. A Jupyter Notebook including the code and report is fine but not necessary.