Homework task 1 – Random walks in 2D

Deadline Nov 30, 2021

To solve the task you can use almost any programming language but I recommend to use Fortran *or*C/C++ *or* Rust *for the best performance together with* Gnuplot *for plotting. If you use* Mathematica *notebooks I used for demonstrations during my lectures as a starting point you have to solve also the bonus problem which is otherwise voluntary.*

The goal

To simulate and analyze various types of random walks on the lattice in the plane (all steps are of the same length $d = 1$, but the direction is randomly chosen from a certain set of prescribed possibilities).

Details

Using $N_w \geq 500$ random walks of n steps starting at the origin determine the mean Euclidean distance R after n steps together with its standard deviation. Plot the dependence of the mean distance (including error bars given by the standard deviation) on the number of steps n.

Check that for all walks this dependence is of the form

$$R(n) \doteq c\, n^{\alpha} \tag{1}$$

where $\alpha \in (0, 1)$. Determine the constant c and the exponent α by the least-square fitting (you can use e.g. the built-in function FIT in Gnuplot) of the logarithm of the equation (1)

$$\ln R(n) \doteq \ln c + \alpha \ln n\,. \tag{2}$$

for the following types of random walks on the square lattice:

1. **Simple random walk** – steps in any of the four directions (up, down, left, right) with the same probability.

2. **Random walk without immediate returns** – steps in three possible directions with the same probability, the first step is in any direction.

3. *Bonus problem:* **Self-avoiding random walk** – a step is allowed only to the neighbor lattice point which was not visited during the previous steps of the random walk. If a walk ends after k steps, then use it for averaging only for $n \leq k$.

If you are interested you can also try to find the average number of steps of the self-avoiding random walks in the 2D square lattice (it should be between 70-71 steps, see J. Chem. Phys. 81 (1984) 584) or you can analyze random walks on non-square lattices, e.g. a hexagonal one.