In-Class Work: Random Walks

At the beginning of class, I will walk you through the solution of step 2d of last class, some theory, and testing of the theoretical prediction with step 2e.

2. Random Walk in One Dimension

Next we will do important analysis on the random walk. To simplify the task (and yet being able to get the main concept) let us start with the random walk in one dimension.

2a. Write a program for a random walk in one dimension. In all following we assume $p = q = 0.5$, i.e. there is equal probability to jump to the right or to the left with jump size 1. Initialize with $x(t = 0) = 0$ (so x=0) and print t and x for each time step. Use `NSTEPS = 500` time steps. Look at the resulting $x(t)$ with `xmgrace` (or with python plotting).

2b. The next step is a preparation for 2c. Instead of printing every time step, print only once after $N =$`NSTEPS` time steps (i.e. after the time-loop) the resulting $x(N)$ and $(x(N))^2$. Increase `NSTEPS` to `NSTEPS=5000`.

2c. Now add a loop over simulation runs to your program from 2b. Each simulation run starts with $x = 0$ and gives you an x(`NSTEPS`) and an x^2(`NSTEPS`). First set `NSIMRUNS=10` and print for each simulation run x(`NSTEPS`) and x^2(`NSTEPS`). Next change your program such that it determines the average over `NSIMRUNS=10000` simulation runs of x and an x^2 and prints out the resulting averages $\langle x \rangle$ and $\langle x^2 \rangle$.

2d. Next no longer use the number of steps `NSTEPS` as constant but instead add a loop over $N=$`nsteps` to your program of 2c. Loop `nsteps` from 100 to 2000 in steps of 100. For each `nstep` print out $N=$`nstep` and $\langle x^2 \rangle$. Look at the resulting $\langle x(N)$ and also $\langle x^2(N) \rangle$. Please get me when you got this done, so that we can interpret your results with the class. Try if you can derive a theoretical prediction.

2e. In the following steps we will use `gnuplot` to fit the resulting $\langle x^2(N) \rangle$ simulation data. First save your data with `./classrndwalk2d.py > dat2d`, Then type the command

```
gnuplot
```

This will start a session in the graphics-tool `gnuplot`. To do a power law fit and to look at the comparison of the fitted line and your simulation data type

```
a=1;b=1;f(x)=a*x**b;fit f(x) "dat2d" using 1:3 via a,b;
plot "dat2d" using 1:3,f(x)
```