Statistische Physik im Gleichgewicht

WS 2023/2024 – Blatt 1

Dr. Milos Knezevic, Dr. Jeanine Shea Abgabe 30.10.2023, 12:00

Problem 1: Random Numbers (*3 Points*)

[C] For the following tasks, we suggest using the `python` package `numpy`. In particular, the functions contained within the package `numpy.random` and the function `numpy.histogram` should be helpful. For plotting, we recommend the functions contained within `matplotlib.pyplot`. `Numpy` is a powerful library, therefore it is a good idea to get acquainted with some of the functions that it offers.

Create functions that generate random numbers within the following distributions:

(a) uniform distribution with permissible values between $x = -10$ and $x = -4$,

(b) the standard normal distribution (i.e. with mean zero and standard deviation 1),

(c) Create discrete random numbers drawn from a Poisson distribution:

$$P_\lambda(k) = \frac{\lambda^k e^{-\lambda}}{k!},$$

using the mean $\lambda = 3$.

(d) Create and plot histograms for the random numbers generated in order to check if you have recovered the correct probability distribution.

Hint: Histograms can be generated with `numpy.histogram/ matplotlib.pyplot.hist`.

Problem 2: Random walk on a grid (*7 Points*)

[C] We consider a sequence of moves on a lattice due to random fluctuations, called a Random Walk. At every step from 0 to N a particle moves a distance Δ either up, down, left or right with equal probability. We use $\Delta = 1$ here. We can imagine this as a simple model of Brownian diffusion.

(a) Write the function `grid_rw`. This function should take the number of steps N as input and returns a two-dimensional `numpy` array with the x and y positions of every step of a random walk on a grid, as described above.

(b) Using this function, simulate 3 random walks with $N = 1000$ steps and plot the trajectories.

(c) For $\Delta = 1$, perform 1000 random walks each for 100 different numbers of steps, namely $N \in 10, 20, 30, ..., 980, 990, 1000$. For each N, calculate the mean square end-to-end distance $\langle |r(N) - r(0)|^2 \rangle$, where $\langle ... \rangle$ represents the ensemble average. Plot the mean square end-to-end distance as a function of the number of steps N, and also plot errorbars. Perform a polynomial fit in order to ascertain that a linear function describes the data well. Plot the best linear fit with your data.

Hint: A great help for fitting polynomials is the function `numpy.polyfit`. *Errorbars can be included with the function* `matplotlib.pyplot.errorbar`.

(d) Perform 1000 random walks each with $N = 1000, 2000, 3000$. This time plot the distributions of end-to-end distances $p(x(N) - x(0))$ and $p(y(N) - y(0))$.

Hint: Again, you can use `numpy.histogram`.

Feedback:

Roughly how much time did you spend on this problem set?