Final Project

Due: 12.3.25

Guidelines

The submission should include two files:

1. PDF file with the figures + text with your answers.

2. Your code.

Your grade will be based on correct execution of the problem, as well as the correctness and quality of your figures. Specifically, points will be rewarded (or reduced if lacking) on the following:

- The ability of your figure to tell a story: while a text should accompany each figure (see below) your figures should stand on their own. Namely, the reader should be able to understand your results without reading the text.

- Descriptions and conclusions: Each figure should include a description as well as conclusions. Describe the axes, the colors, marker shapes (if relevant), etc. Conclusions should give insight into the information and data presented in the figure. For example, if there's a peak, are you surprised there is a peak? What does the peak mean? What does the location of the peak mean? Or if there is a trend (linear, power-law, exponential, etc), discuss it. If you compare with analytical results, discuss this as well.

- Formatting your figures. Pay attention to:
  - Inward ticks.
  - Colors.
  - Plot lines vs. scatter points.
  - Correct scaling of font size inside the figure (e.g. labels).

- We will not accept figures saved as screenshots. You are required to save your figures in Python.

- You are required to execute the number of walks and steps written in the problem. If you do less points will be deducted.

Question 1 (33 Points)

In HW4Q2 you saw a random walk with probability of staying put. It was defined as follows

$$\begin{cases} S_n = X_1 + X_2 + ... + X_n \\ Pr(X_k = 1) = 1/4 \\ Pr(X_k = 0) = 1/2 \\ Pr(X_k = -1) = 1/4 \qquad\qquad k = 1, 2, ..., n \end{cases}$$

where S_n is the position of the random walker after n steps, $1/4$ is the probability to move left or right, and $1/2$ is the probability to stay put.

Simulate 1 million random walkers as defined above for 100 steps.

(a) Plot (on the same figure) 3 trajectories. For each trajectory use a different color. Add a legend. *(5 points)*

(b) From your simulations, estimate the probability to be at the origin after $n = 0, 1, ..., 100$ steps. Compare, using a plot, to the analytical result you got in HW4Q2. Are there any steps for which the probability to be at the origin is zero? Compare and contrast this to the simple symmetric 1D random walk. *(9 points)*
Plotting guide: If you plot all your data, it may look too crowded: different points visually overlapping, etc. To best visualize your data you will need to filter out some points. You can do this using:

```
indexes = [ .... ]
filtered_sites = sites[indexes]
filtered_probabilities = probabilities[indexes]
```

You can choose indexes manually. For example:

```
indexes = np.concatenate((np.arange(0, 5, 1), [6], np.arange(9, 100,
                                          4)))
```

(c) Recall that the probability to be at site j after n steps is given by:

$$P_n(j) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \lambda^n(\theta) e^{-i\theta j} d\theta$$

Use the characteristic function, $\lambda(\theta)$, that you found in HW4Q2 and numerically evaluate the probability to be at site j after $n = 20$ steps. Do this for all the js from your simulation. Plot your results and compare them to estimates coming from your simulation.

Use scatter points with blue circles for the numerical evaluation, and use scatter points with green crosses for the simulation results. Make sure to size the scatter points such that the crosses are inside the circles. *(9 points)*

(d) For $n \gg 1$ we can estimate $P_n(j)$ using the CLT. The CLT predicts that the exact probability distribution governing the position of the walk can be approximated by another distribution. Set $n = 20$ and follow these steps:

(i) What is that distribution? Generate 1 million samples from it. There is no need to plot. *(3 points)*

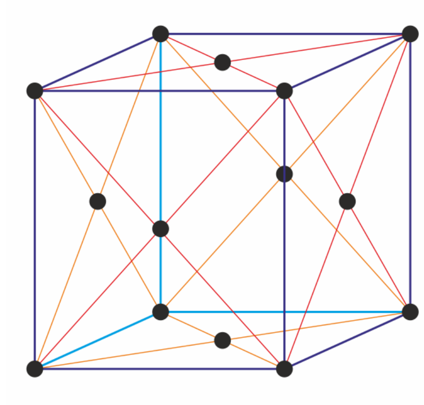
(ii) Bin the generated samples according to the sites in (c) (no plot yet, just make sure you are able to bin). You can do this using this function: *(2 points)*

```python
def bin_specific_sites(sites, new_data):
    bin_width = np.diff(sites)
    first_edge = sites[0] - bin_width[0]/2
    last_edge = sites[-1] + bin_width[-1]/2

    # Create bin edges array
    bin_edges = np.zeros(len(sites) + 1)
    bin_edges[0] = first_edge
    bin_edges[-1] = last_edge
    bin_edges[1:-1] = sites[:-1] + bin_width/2

    # Perform the binning
    counts, _ = np.histogram(new_data, bins=bin_edges, density=
                                            True)
    return bin_edges
```

(iii) Plot, in the same figure, the probability to be at site j after n steps as follows: the simulation estimations of $P_n(j)$ as empty blue circles, and the CLT approximation of $P_n(j)$ as orange crosses. *(5 points)*

Question 2 (34 Points)

Consider the face-centered-cubic (FCC) lattice:



Assume the lattice spacing is 1.

The MSD of a symmetric random walk on the FCC lattice is

$$\langle \vec{r}^2 \rangle = \frac{1}{2} n. \tag{11.1}$$

Simulate 1 million symmetric random walks on the FCC lattice for 1000 steps. Answer the following:

(a) Plot the MSD as a function of the number of steps (linear-linear) for the first 100 steps only. Perform a fit and compare to the analytical prediction Eq. (11.1). *(5 points)*

(b) Create a new figure: MSD as a function of the number of steps for a symmetric random walk on the FCC, BCC, and simple 2D, lattices. Use a linear-linear plot, and plot up to $n = 100$. Use your simulations from HW6Q2 for the BCC and simple 2D lattice (use different colors). Perform a fit for all three lattices (FCC, BCC and simple 2D). Discuss the similarities and the differences between the MSDs on the different lattices. *(5 points)*

(c) Recall that the generating function of the probabilities to be at the origin can be written as (see class 8)

$$P(\vec{o}, z = 1) = \left(\frac{1}{\pi}\right)^3 \int_0^\pi \int_0^\pi \int_0^\pi \frac{1}{1 - \lambda(\theta)} d\theta.$$

Numerically evaluate this integral using the following expression for the characteristic function:

$$\lambda(\theta) = \frac{1}{3}\left[\cos(\theta_y)\cos(\theta_x) + \cos(\theta_z)\cos(\theta_x) + \cos(\theta_y)\cos(\theta_z)\right].^{[1]} \tag{11.2}$$

Use your numerical evaluation to find the probability that this random walk returns to the origin. *(3 points)*

(d) Use simulations data to plot the probability to return to the origin as a function of the number of steps taken. In addition, draw the theoretical prediction you found in (c) as a horizontal line. Use a legend. *(6 points)*

(e) The probability to return to the origin by step n can be used to estimate the probability to eventually return to the origin. Plot the relative error in this estimate as a function of the number of steps n. Discuss the results obtained. *(4 points)*

(f) Calculate, for each random walk in your simulations, the number of distinct sites visited as a function of the number of steps taken. Plot the average number of distinct sites visited as a function of the steps. Use a linear-linear plot. Is the relation linear? Are you surprised? *(6 points)*

(g) Fit the data for large n and obtain the slope. From the slope, find a way to estimate the probability to return to the origin. Choose large enough ns for the fit such that the relative error is less than 3.5%. Why does the accuracy improve when fitting to large values of n? *(5 points)*

[1] As discussed in the review session, to comply with definitions given in class this characteristic function was calculated by setting to unity the distance between nearest neighbors on the FCC lattice.

Question 3 (33 Points)

Consider the overdamped harmonic oscillator. The Langevin equation for it is

$$\frac{dx(t)}{dt} = \frac{F_{\text{ext}}}{\xi} + \frac{\eta(t)}{\xi} = -\frac{kx}{\xi} + \frac{\eta(t)}{\xi},$$

where $F_{\text{ext}} = -\frac{dU(x)}{dx} = -kx$ and with $U(x) = \frac{1}{2}kx^2$. Note, that this equation is equivalent to the Orenstein–Uhlenbeck process process with:

$$x(t) \mapsto v(t)$$
$$\xi \mapsto m$$
$$k \mapsto \xi$$

Simulate one million particles with the following parameters:

- $\Delta t = 0.005$
- $t = 10$
- $k_B T = k = \xi = 1$

Choose any positive initial position, and answer the following:

(a) Plot (in the same figure) the trajectories of three particles. *(5 points)*

(b) Plot, in the same figure, the distribution of the positions at the following times, using a histogram (use different colors):

- After 10 time steps, i.e. at $t = 0.05$.
- After 50 time steps, i.e. at $t = 0.25$.
- After 200 time steps, i.e. at $t = 1$.
- The end of the simulation (i.e. 2000 time steps and $t = 10$).

Discuss how the positions changed. When did the biggest change occur? *(8 points)*

(c) Create a new figure: plot a histogram of the positions at the end of the simulation ($t = 10$) and at $t = 0.05$. In class you got the theoretical expression for this distribution,

$$p(x,t) = \frac{1}{\sqrt{2\pi\sigma_x^2(t)}} \exp\left(-\frac{(x - \langle x(t)\rangle)^2}{2\sigma_x^2(t)}\right)$$

where $\langle x(t)\rangle$ and $\sigma_x^2(t)$ are the mean and variance of the position at time t.

Plot the theoretical expression for $t = 0.05$ and $t = 10$ in the same figure, using plt.scatter. Use the theoretical predictions for $\langle x(t) \rangle, \sigma_x^2(t)$ (plug the relevant parameters in the analytical expressions). Does the theoretical expression agree with the simulation results? *(8 points)*

(d) Estimate from your simulations, as a function of time, the following:

- The average position.
- The variance of the position.
- The position autocorrelation function $\langle (x(t) - \langle x(t) \rangle)(x(t') - \langle x(t') \rangle) \rangle$. Take $t' = 5$ and plot for $0 \le t \le 10$.

Compare the average, variance and autocorrelation function to the analytical predictions. Use a plot. You may use plt.plot (rather than scatter with different markers) to compare the analytical and simulation result. If you use plt.plot use 'o' for the simulations and '-' for the analytical prediction. *(12 points)*