Estimating Distance from Origin in a 2D Random Walk

This report describes the approach for estimating the distance d from the origin after a 2D random walk with N steps, where each step moves in one of four directions (up, down, left, or right) with equal probability (0.25). The project aims to evaluate the average distance $\langle d \rangle$ after N steps and compare it to the theoretical expected distance, $d=\sqrt{N}$.
Below is the code used to simulate the random walk, calculate the average distance and uncertainty, and plot the results.

Code

```python
import numpy as np
import matplotlib.pyplot as plt
import os

# Parameters
N_steps = [10, 20, 40, 80, 160, 320, 640, 1280]
k = 10  # number of trials for averaging

# Random walk function
def random_walk_2d(n_steps):
    x, y = 0, 0  # starting point
    for _ in range(n_steps):
        direction = np.random.choice(["up", "down", "left", "right"])
        if direction == "up":
            y += 1
        elif direction == "down":
            y -= 1
        elif direction == "right":
            x += 1
        elif direction == "left":
            x -= 1
    return np.sqrt(x**2 + y**2)  # final distance from origin

# Theoretical values for sqrt(N_step)
theoretical_d = [np.sqrt(N) for N in N_steps]


for trial in range(1, 4):
    avg_d = []
    delta_d = []
    for N in N_steps:
        distances = [random_walk_2d(N) for _ in range(k)]
        mean_d = np.mean(distances)
        disp_d = np.var(distances, ddof=1)
        uncertainty_d = np.sqrt(disp_d)
        avg_d.append(mean_d)
        delta_d.append(uncertainty_d)

    plt.figure(figsize=(10, 6))
    plt.errorbar(N_steps, avg_d, yerr=delta_d, fmt='o', label=f'<d> (Trial {trial})', capsize=5, color='blue')
    plt.plot(N_steps, theoretical_d, 'r--', label='Expected d = sqrt(N)')
    plt.xlabel("N_step")
    plt.ylabel("Distance d")
    plt.title(f"Distance d from Origin after N Steps in 2D Random Walk (Trial {trial})")
    plt.xscale("log")
    plt.yscale("log")
    plt.legend()
    plt.grid(True)

    os.makedirs("plots", exist_ok=True)
    plt.savefig(f"plots/random_walk_trial_{trial}.png")
    plt.close()
```

Plots

The same calculation was repeated three times to illustrate the randomness.

Distance d from Origin after N Steps in 2D Random Walk (Trial 1)
Expected d = sqrt(N)
<d> (Trial 1)
Distance d
10^1
10^1
10^2
10^3
N_step

Distance d from Origin after N Steps in 2D Random Walk (Trial 2)
Expected d = sqrt(N)
<d> (Trial 2)
Distance d
10^1
10^1
10^2
10^3
N_step


Distance d from Origin after N Steps in 2D Random Walk (Trial 3)
Expected d = sqrt(N)
<d> (Trial 3)
Distance d
N_step
10^0
10^1
10^2
10^3