A Random walk is a series of movements made by an agent (in the case of our experiment a particle) and is a simple method of simulating Brownian motion. Brownian motion and the concept of the random walk were first scientifically examined by Albert Einstein in his 1905 paper, *Investigations on the Theory of the Brownian Movement*. Brownian Motion is described as small, random movement that particles continuously undergo.

Despite being random, certain properties of random walks follow patterns. Our simulation examines the results produced by tracking the Mean Squared Displacement (MSD) that particles travel inside a simulated vacuum.

**Objective & Impact of Professor’s Research**

Random walks have many applications to chemical engineering and biomedical engineering, the most notable application being the ability to simulate reactions in different types of medium. This has applications ranging from simulating pollutant concentrations in water to monitoring the release of a drug into a person’s bloodstream. There are even economic applications to random walks and Brownian motion, including asset value prediction. Our simulation aims to determine trends that arise from these random movements given bias in some direction within a homogeneous medium.

**Introduction**

A Random walk is a series of movements made by an agent (in the case of our experiment a particle) and is a simple method of simulating Brownian motion. Brownian motion and the concept of the random walk were first scientifically examined by Albert Einstein in his 1905 paper, *Investigations on the Theory of the Brownian Movement*. Brownian Motion is described as small, random movement that particles continuously undergo.

In the past 7 weeks, I have learned a great deal about matrices and Numpy, as well as machine learning that we used in a different project to predict the final resting position of a particle. These skills have furthered my interest in the field of data science, as well as the applications of Python and Machine Learning to fields outside of Computer Science.

**Methods and Process**

In our simulation, we executed an explicit 2D random walk, meaning that the particle is confined to only moving whole steps in 4 directions: up, down, left, and right. Before running the simulation, we first assign a bias in the positive X direction, meaning that it is some degree more likely to choose right over the other 3 directions. The formulas for the probabilities of each direction are:

\[
P(NSE) = \frac{(1-b)}{4} \quad P(W) = \frac{(1+3b)}{4}
\]

The sums of the probabilities is 1. After 100,000 steps, we take the natural logarithm of its MSD. MSD can be represented as

\[
MSD(t) = r^2(t) = K \times t^\alpha
\]

where K is the proportionality constant, t is time, r is displacement, and alpha is the diffusion coefficient.

**Skills Learned**

In the past 7 weeks, I have learned a great deal about matrices and Numpy, as well as machine learning that we used in a different project to predict the final resting position of a particle. These skills have furthered my interest in the field of data science, as well as the applications of Python and Machine Learning to fields outside of Computer Science.

**Results**

The slope of the logarithm of the MSD allows us to find the diffusion coefficient and examine its change over time. In a purely homogenous solution, alpha is 1, as shown on the graph to the right. Alpha approaches 2 as the bias approaches 1. When the Bias is higher, the mean displacement per step approaches one. Since the squared displacement is the square of this value, the logarithm of the squared displacement has a coefficient of two.

**Citations**


**Next Steps for You OR Advice for Future SHINE Students**

SHINE has furthered my interest in finding different ways to apply my skills in computer science to different fields. I plan on continuing research on random walks and applying my simulation to different porosity fields.

**Acknowledgements**

Thank you to Shayan Jalalmanesh and Shayan Rahimi for teaching me the skills required to perform this simulation as well as guiding me through debugging my code.

Thank you to Professor Sahimi as well, for selecting me to undertake this research.