Random Walk Simulation

Anirudh Prakash

Abstract:

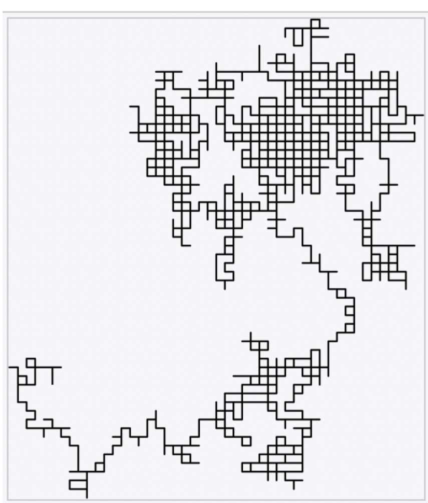
A random walk is a statistical model which describes the path followed by an atom across the lattice during its random motion. This can most easily be related to solid state diffusion where an atom moves from one lattice point to another in a completely arbitrary fashion, as long as the lattice or interstitial site to which its moving to is empty. Thus, in the most ideal case of random walk, we can consider all lattice points to be empty and simulate the random movement of the atom in the lattice. The first three sections of this report deals with the ideal case of random walk in one, two and three dimensional lattices. In the ideal scenario, there is an equal probability of the atom to move along any direction. The final section of the report deals with the biased random walk simulation where the motion of the atom is constrained by defining unequal probabilities for movement in different possible directions. In every scenario, we have plotted a graph for the movement of the atom on the x axis(1 dimensional case), x-y plane (2 dimensional case) and in the x-y-z space (3 dimensional case) to visualise the simulation. We have also plotted graphs for displacement vs time and mean squared displacement vs time to analyse the simulation in detail. For the biased 2 dimensional random walk simulation, we have defined the probability of the electron to move up, down, left or right to introduce some sort of restrictions and also analysed the simulation in detail by plotting graphs of the above mentioned parameters.

Introduction:

At temperatures above then absolute zero (0K), the atoms present at the different points possess some amount of energy known as thermal energy which is directly proportional to the temperature. This thermal energy is perceived in the form of kinetic energy of the atoms. Thus as temperature increases, the average kinetic energy of the atoms increases and the atoms start moving randomly from lattice point to another (or from one interstitial site to another) provided they are empty.
This random motion of the atoms across the lattice is termed as random walk or drunkard walk as it is very much related to the walk of a drunk person with reduced cognitive abilities.

Thus,

“Random walk is a stochastic process that describes a path that consists of a succession of random steps on some mathematical space”



Random walk in 2D space

<u>Brief History of Random Walk:</u>

The history of random walk dates all the way back to the early 19th century when Robert Brown observed the Brownian motion(random motion of particles suspended in a fluid).
Albert Einstein further studied brownian motion, providing a theoretical explanation of the random motion of particles. The English mathematician Karl Pearson first coined the term 'Random Walk' in 1905.
In the mid 20th century, Random Walk became a fundamental tool in statistical physics to model phenomena such as diffusion and the behaviour of particles in liquids and gases.

Mathematical Analysis of Random Walk

In a random walk, an atom moves step by step in random directions.

1. The Initial Position of the atom is taken as (x0,y0,z0). In the simulations done below, the initial position is taken as the origin (0,0,0).
2. New position after each step is given as:

   x1 = x0 +(-) a (if the atom moves along the x-axis)
   y1 = y0 +(-) a (if the atom moves along the y-axis)
   z1 = z0 +(-) a (if the atom moves along the z-axis)
3. In an unbiased random walk, the atom has equal probability of moving along different directions. In case of a 3D random walk, this probability is ⅙ (as total number of possible directions of movement is 6). Similarly for 2D random walk, it is ¼
4. In a biased random walk, the atom has unequal probabilities of moving in different directions determined by the biasing

probabilities. If the biasing probabilities for the six directions of movement in 3D random walk is [p1, p2, p3, p4, p5, p6], then:

p1+p2+p3+p4+p5+p6 = 1 ; they are mutually exclusive and exhaustive

5. Mean Square Displacement: measures how far the atom is expected to be from its starting point after ‘n’ steps on average.

$$MSD = (d1^2 + d2^2 + d3^2 + \ldots\ldots + dn^2)/n$$

Where n is the number of steps and d is the distance from the initial point. It is calculated using the distance formula:

$$d = \sqrt{(x - x0)^2 + (y - y0)^2 + (z - z0)^2}$$

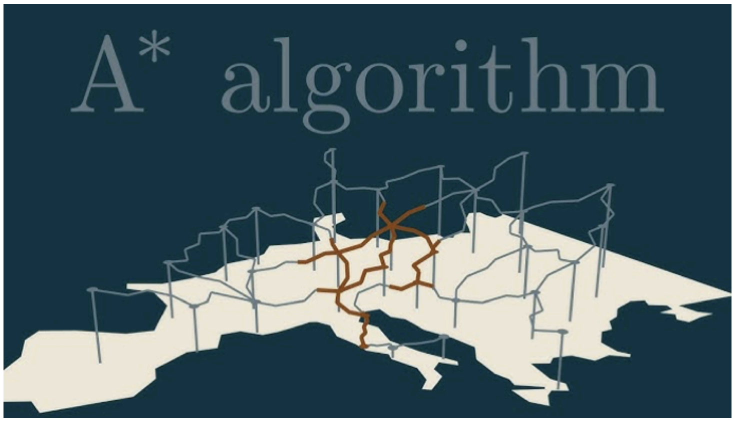
Where (x,y,z) is the position of the atom at that instant of time and (x0, y0, z0) is the initial position of the atom.

Applications of the Random Walk simulation:

In Sciences

In computer science, random walk theory is used to design complex search algorithms. In the A* search algorithm used commonly in graphs in applications like google maps, all possible paths (nodes) from the source to the destination are explored and the shortest path is determined

Reference: https://www.geeksforgeeks.org/a-search-algorithm/



In General Engineering

In civil engineering, random walk is used to simulate the movement of vehicles and roads and modelling the dispersion of pollutants in the atmosphere or water bodies.

Reference:
https://www.britannica.com/technology/civil-engineering/Construction

In Finance

In portfolio optimisation, asset returns are modelled as random walks so that investors can use optimization techniques to construct portfolios that balance risk and return.

Reference:
https://www.investopedia.com/terms/r/randomwalktheory.asp

In Materials Engineering

The most common application of Random walk simulation in materials engineering is the modelling of solid state diffusion where an atom jumps from one lattice site to another vacant lattice site/point defect/interstitial sites. In the real world scenario, there are a lot of factors affecting solid state diffusion, such as temperature and concentration gradient (Fick's Laws of diffusion) which affect the probability of the electron jumping along a certain direction. A virtual lattice can be constructed specifying the empty sites and diffusion modelling can be done using the random walk simulation by calculating the likelihood of electrons jumping in any direction for that particular temperature.

Random walk also finds application in simulating the path of the electron in an SEM beam after it strikes the sample. After striking the sample, the election follows a random zig-zag path which creates the interaction volume. With this method, we can even calculate the depth of interaction volume.

Reference:
https://www.uio.no/studier/emner/matnat/kjemi/KJM5120/v05/undervisningsmateriale/KJM5120-Ch5-Diffusion.pdf

DIFFERENT CASES OF RANDOM WALK SIMULATED BY THE CODES IN THE LAB:

1. **One Dimensional Unbiased Random Walk:** The atom is free to occupy lattice sites along the X-axis. The distance between two consecutive lattice points is taken as one unit. Since the random walk in this case is unbiased, the atom has equal probability of jumping left or right with the step size being one unit.
2. **Two Dimensional Unbiased Random Walk:** The atom is free to occupy lattice sites along the X and Y axis. The distance between two consecutive lattice points along any given axis is one unit. Since the random walk is unbiased, the atom has equal probability of jumping left, right, up or down with the step size being one unit.

3. **Three Dimensional Unbiased Random Walk:** The atom is free to occupy lattice sites along the X, Y and Z axis. The distance between two consecutive lattice points along any given axis is one unit. Since the random walk is unbiased, the atom has equal probability of jumping left, right, up, down, into the plane or out of the plane with the step size being one unit.
4. **Special Case of Unbiased Random Walk** where the number of steps is 5 times the number of lattice sites.
5. **Biased Random Walk:** The movement of the atom is restricted by biasing probabilities. The probability of the atom moving along different directions is different.

METHODS:

Flowchart: *Flowchart has been uploaded as a separate file.*

Parameters under which the code has been run and analyzed:

1. **nsteps:** Number of jumps the atom makes in one random walk simulation
2. **init_pos:** The coordinates of the initial position of the atom. In every case, it is the origin.
3. **ntrials:** The number of times (trials) the random walk simulation is performed to average the results (Displacement vs Time and Average Trajectory)
4. **probabilities:** The biasing probabilities for each direction of movement. This is done to restrict the movement of the atom in a certain direction.

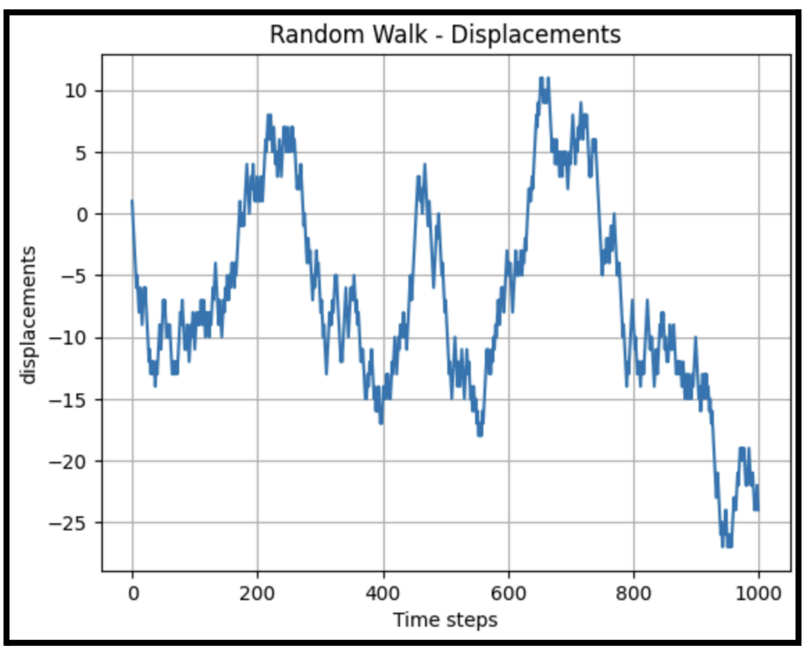
RESULTS AND DISCUSSIONS:

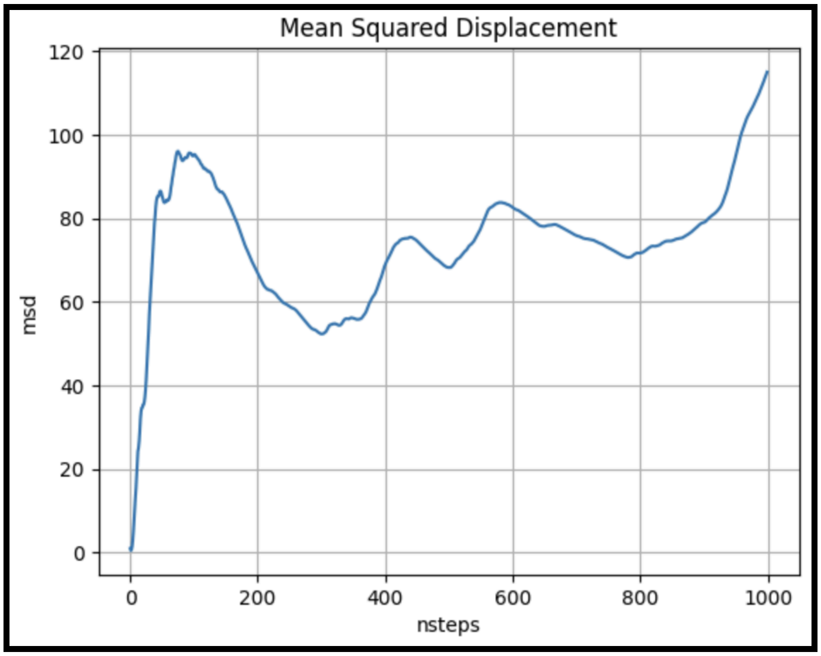
I. Displacement vs Time Steps Graph and Mean Squared Displacement vs Time Steps Graph

The displacement vs time steps graphs always exhibited numerous sharp peaks in unbiased random walk simulations (1,2 and 3 dimensional). The graph did not follow a regular increasing or decreasing pattern. It is characterised by sharp increases and decreases.

Displacement vs Time curves for:

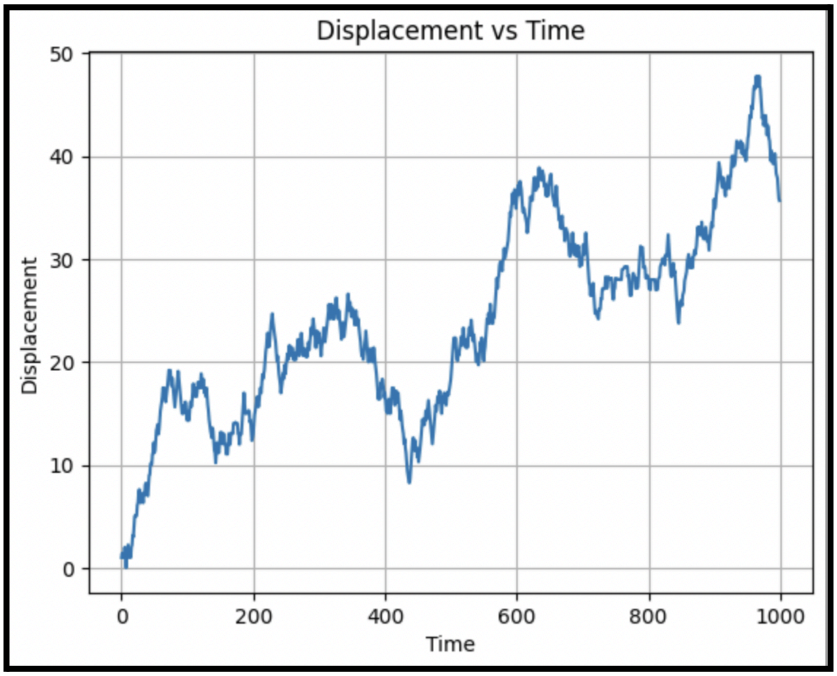
i. *One Dimensional Random Walk*

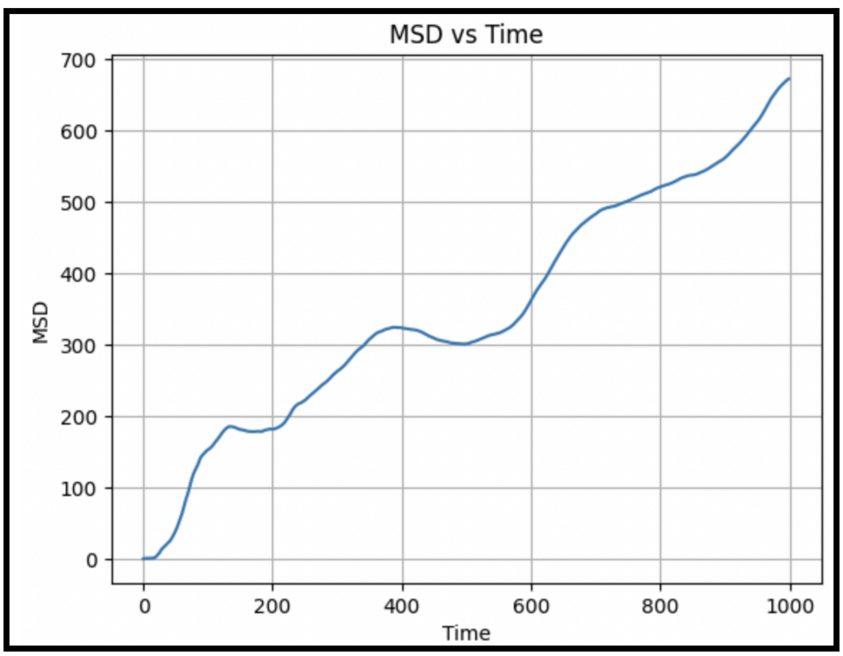




Since the particle can move only left or right, there is no strong tendency for large displacements. With each step, atom moves closer or drifts away from the origin. This is why the atom is seen crossing the origin several times.

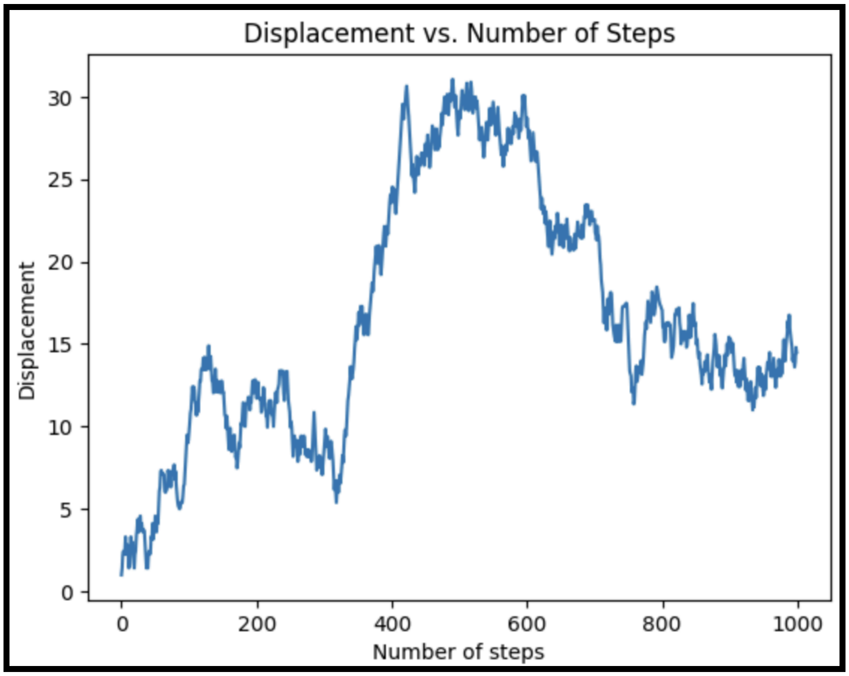
ii. *Two Dimensional Random Walk*

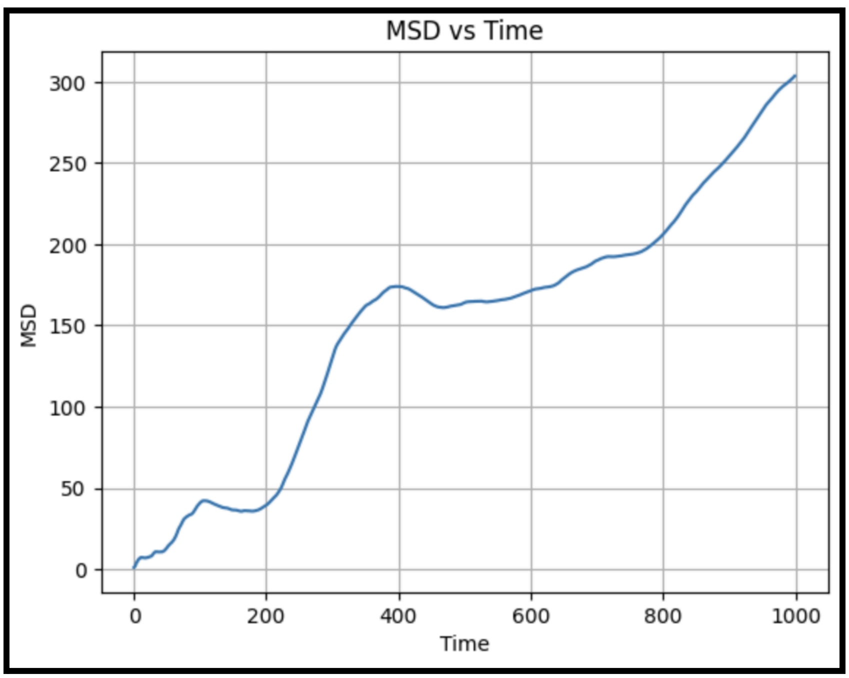




Since the atom has more options to move (up, down, left and right), it tends to have larger net displacements. It is for the same reason that the atom progressively moves away from the origin.

iii. *Three Dimensional Unbiased Random Walk*





The atom now has more freedom for movement (left, right, up, down, forward and backward) so the atom is even less likely to return ti the origin in this case.

In all the above cases, the MSD vs Time Steps plot exhibits an *overall* shape somewhat resembling that of the Displacements vs Time Steps plot. The MSD plot is much smoother and is not spiky unlike the Displacement Plot.

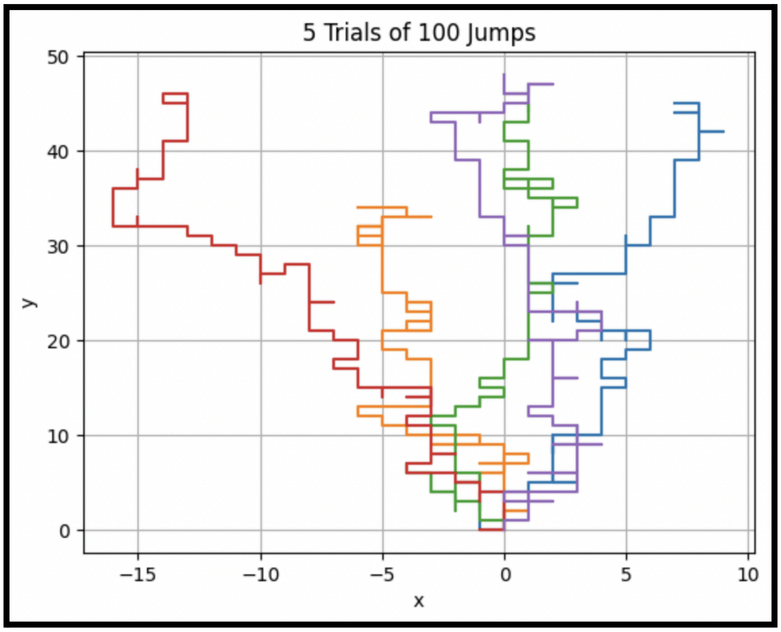
The MSD plot is smoother as it is calculated using a cumulative sum of squared displacements (cumulative averaging process).

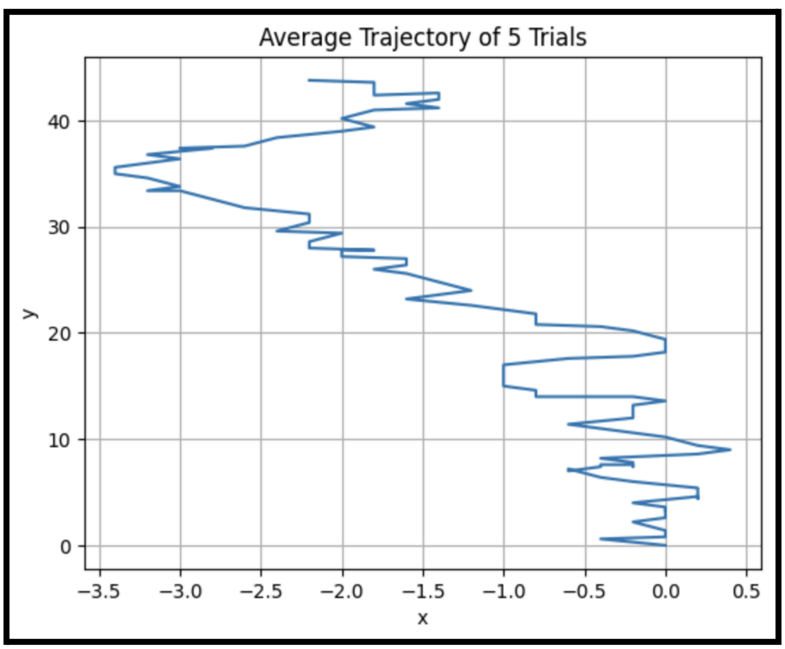
msd = np.cumsum(displacements**2)/np.arange(1,nsteps+1)

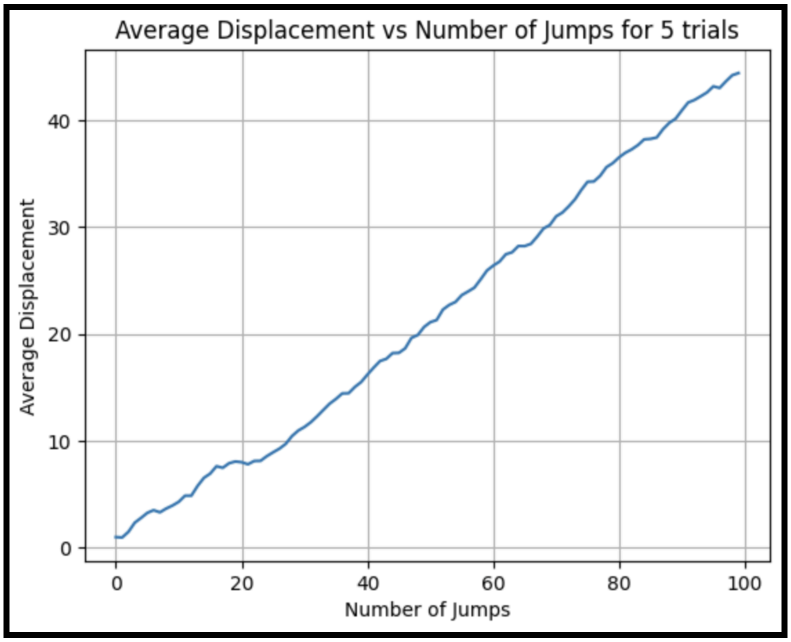
The squared displacements are averaged over a growing number of steps, eliminating all fluctuations or ‘spikes’. Squaring the displacements removes any negative values, further reducing the jagged behaviour.

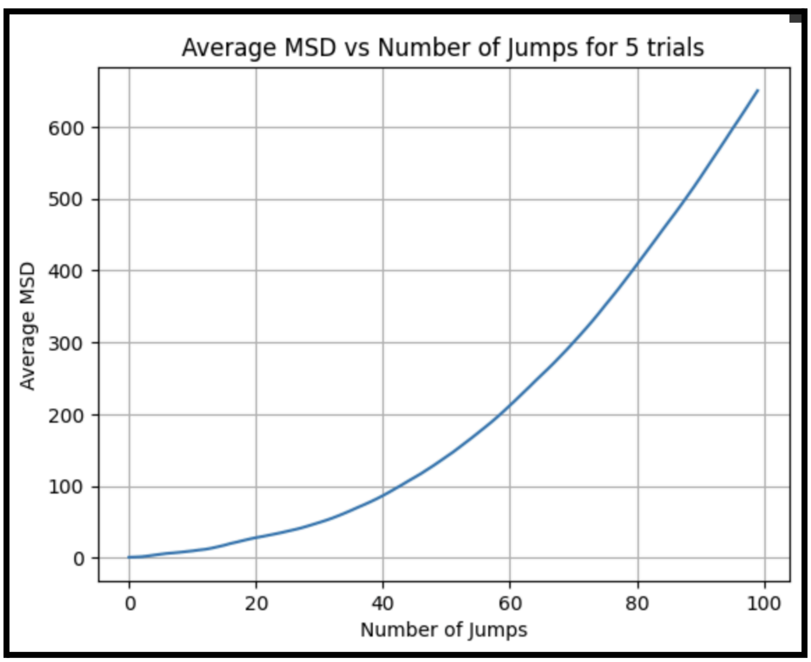
III. Variation in average distance vs time-steps and average mean square displacement vs time steps with increasing number of trials

i. *5 Trials for 100 Jumps each*

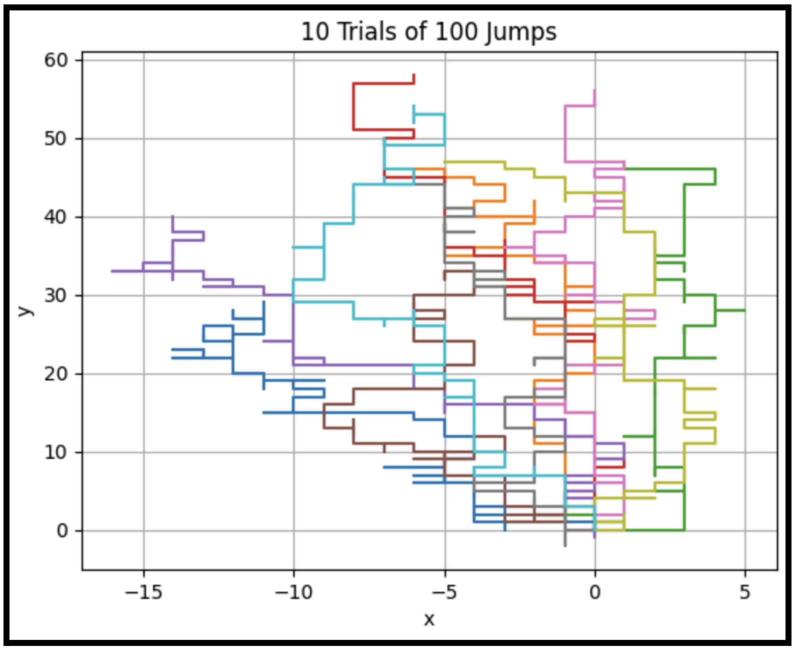


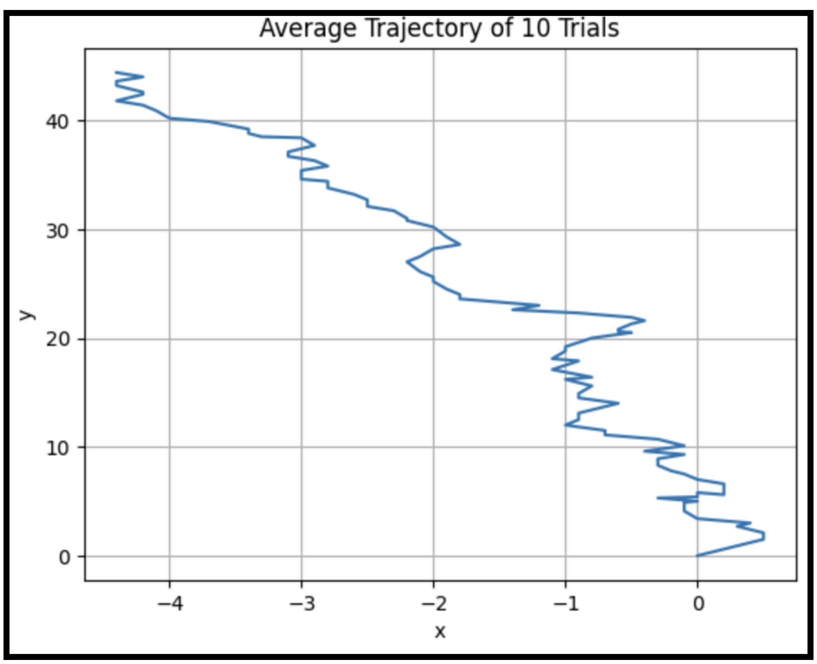


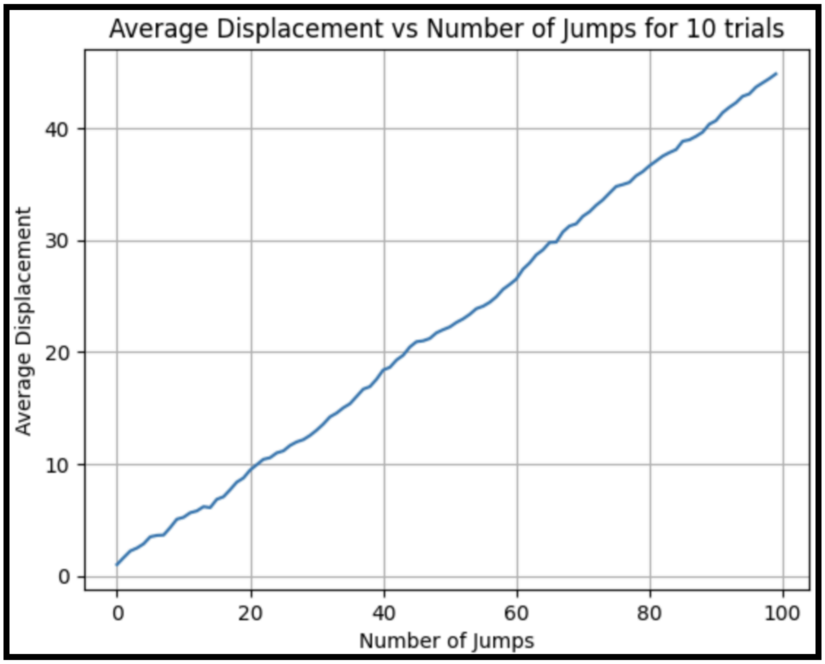

Average Displacement vs Number of Jumps for 5 trials
Average Displacement
Number of Jumps
0
10
20
30
40
0
20
40
60
80
100

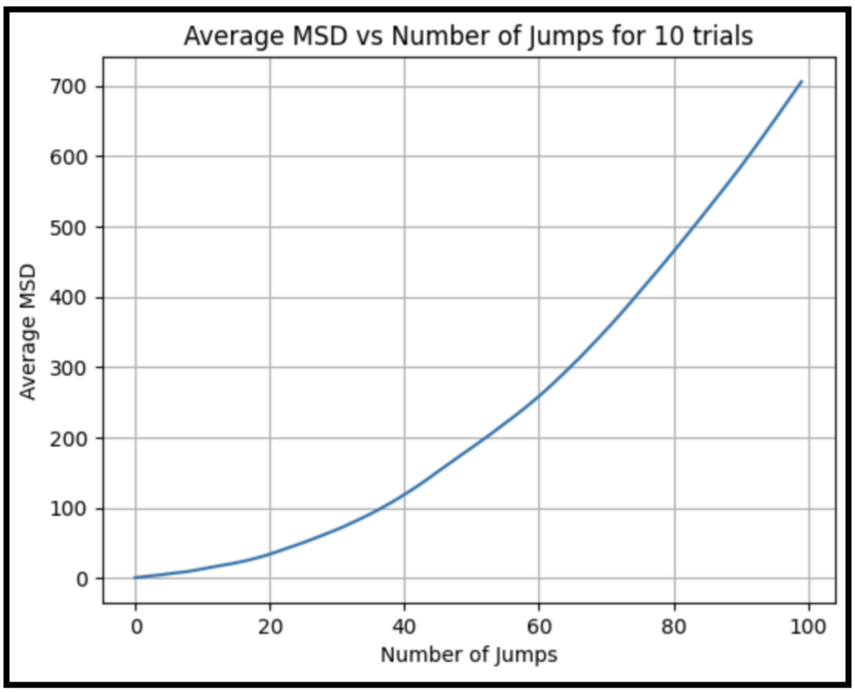

Average MSD vs Number of Jumps for 5 trials
Average MSD
Number of Jumps
0
100
200
300
400
500
600
0
20
40
60
80
100

ii. *10 Trials for 100 Jumps each*

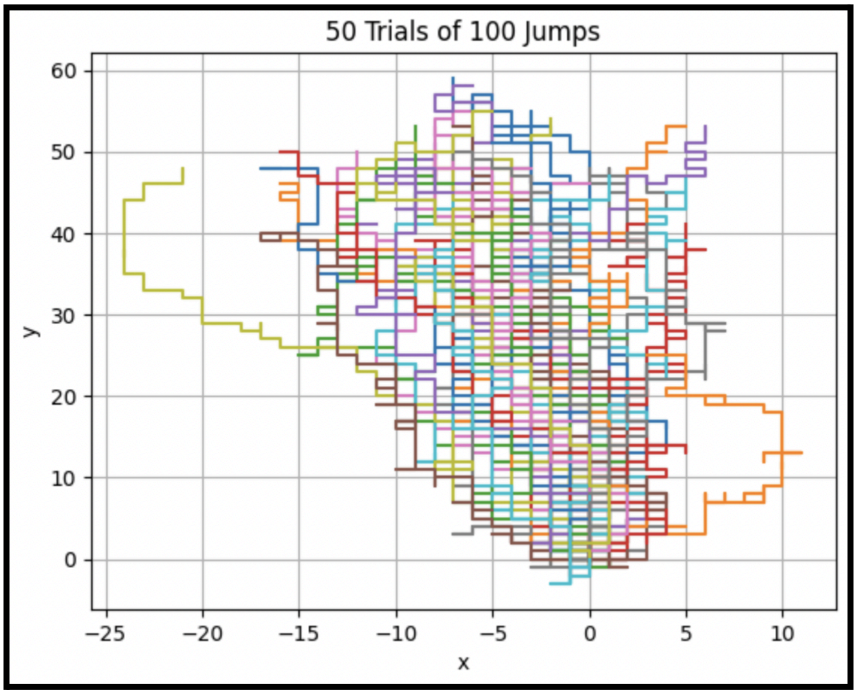


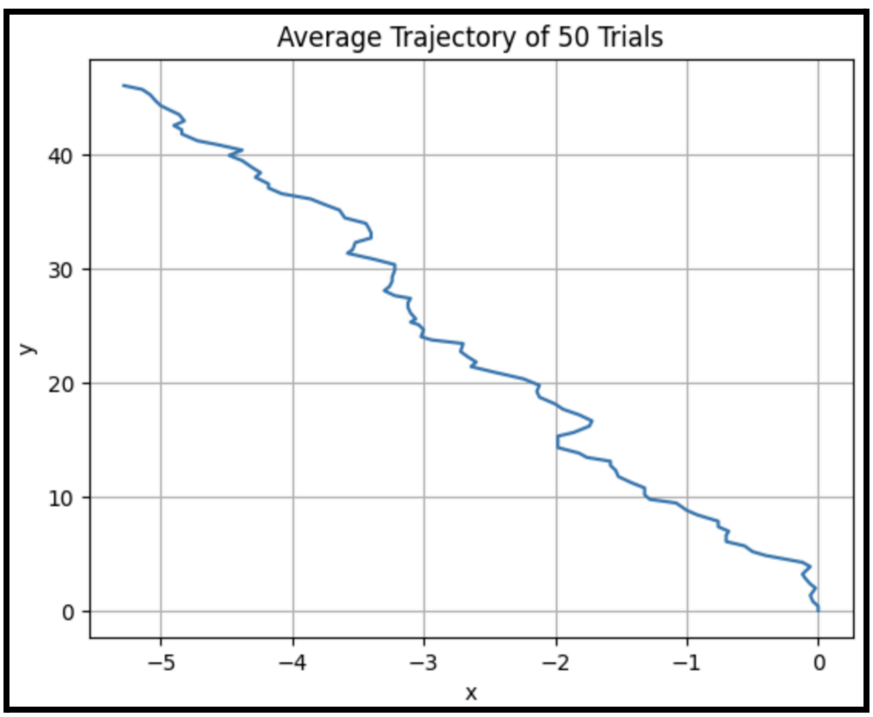


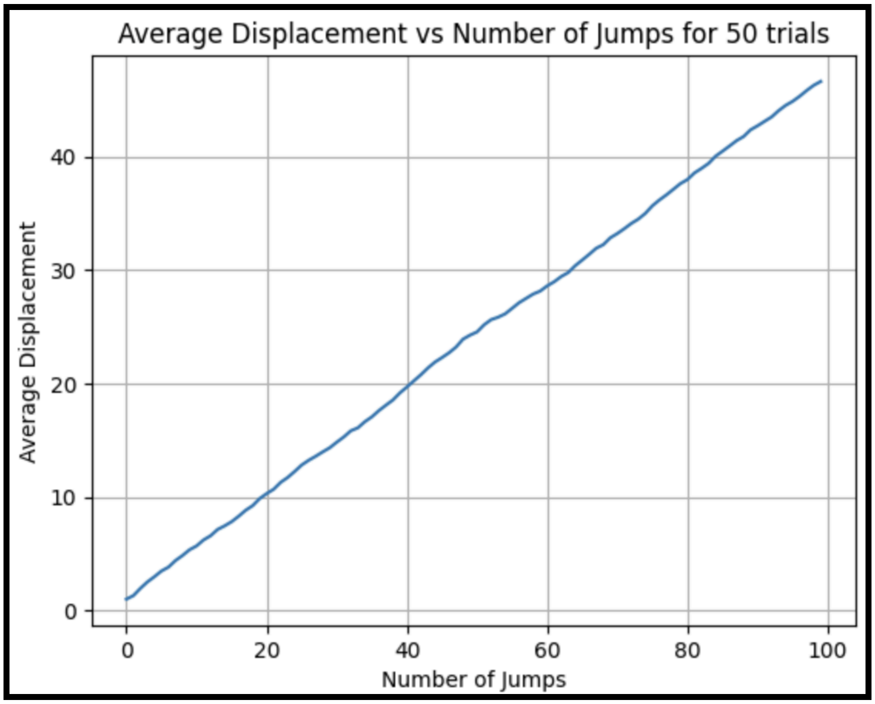

Average Displacement vs Number of Jumps for 10 trials
Average Displacement
Number of Jumps
0
10
20
30
40
0
20
40
60
80
100

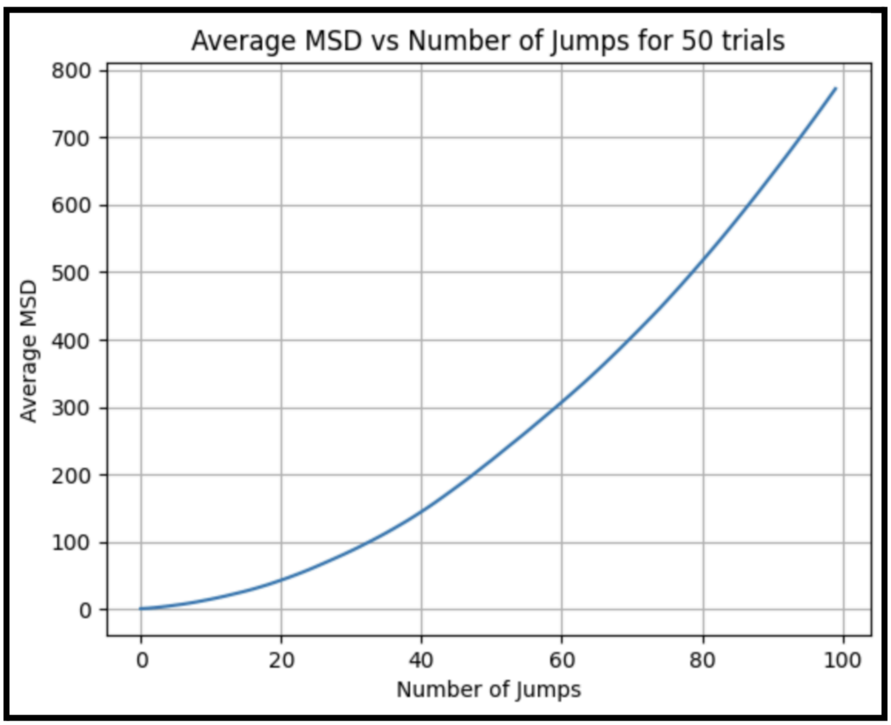

Average MSD vs Number of Jumps for 10 trials
Average MSD
Number of Jumps
0
100
200
300
400
500
600
700
0
20
40
60
80
100

iii. *50 Trials for 100 Jumps Each*

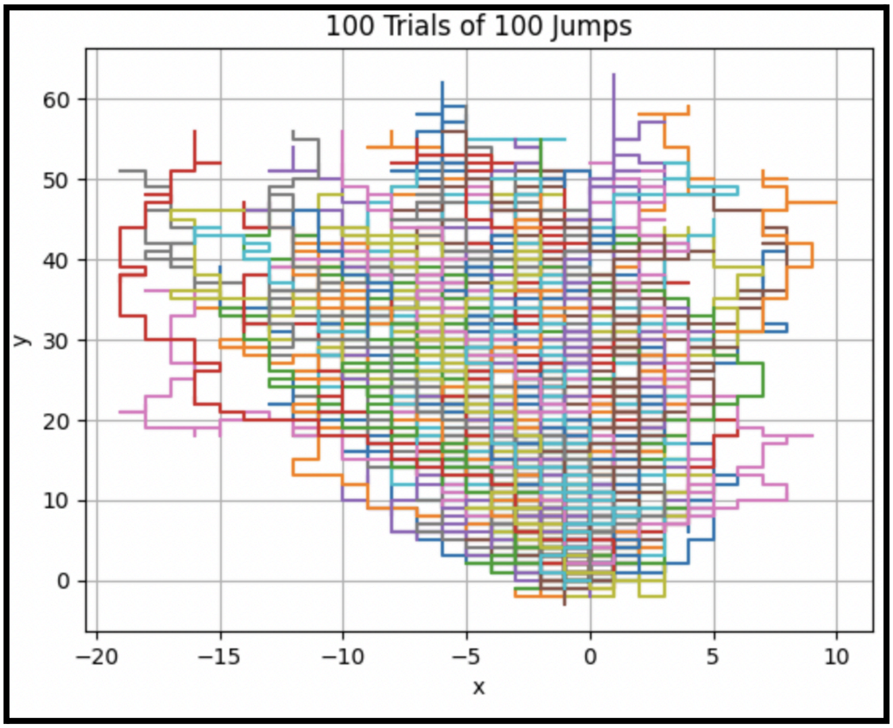


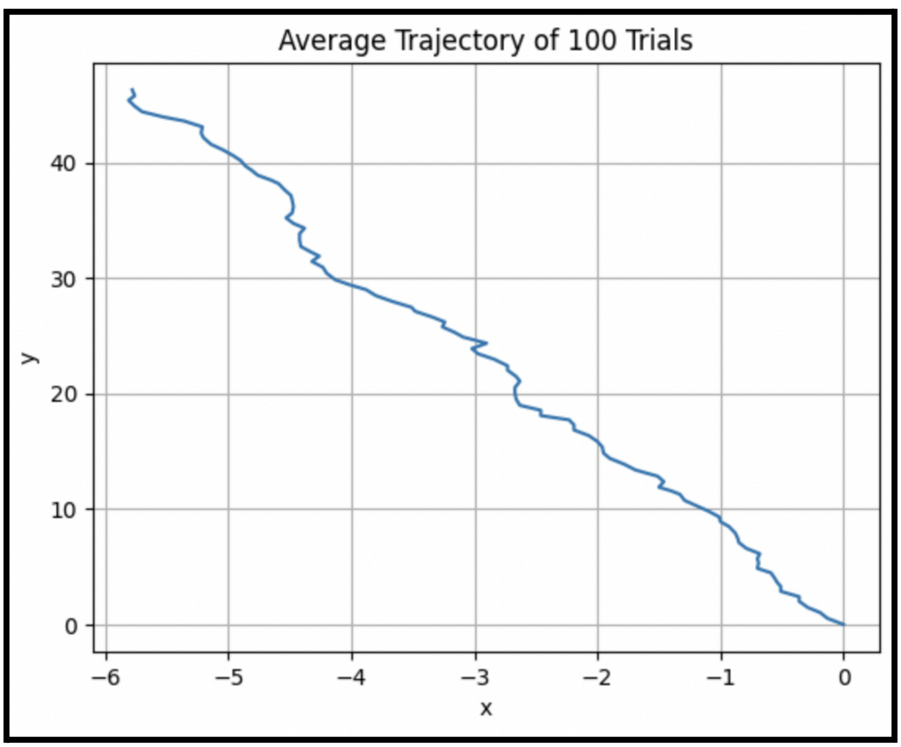


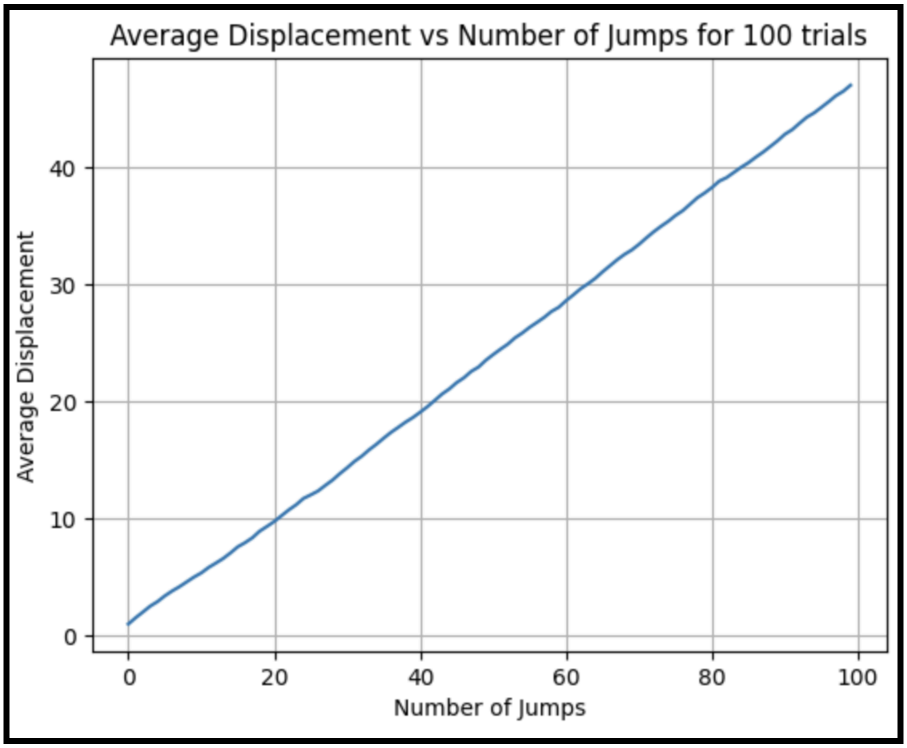

Average Displacement vs Number of Jumps for 50 trials
Average Displacement
Number of Jumps
0
10
20
30
40
0
20
40
60
80
100

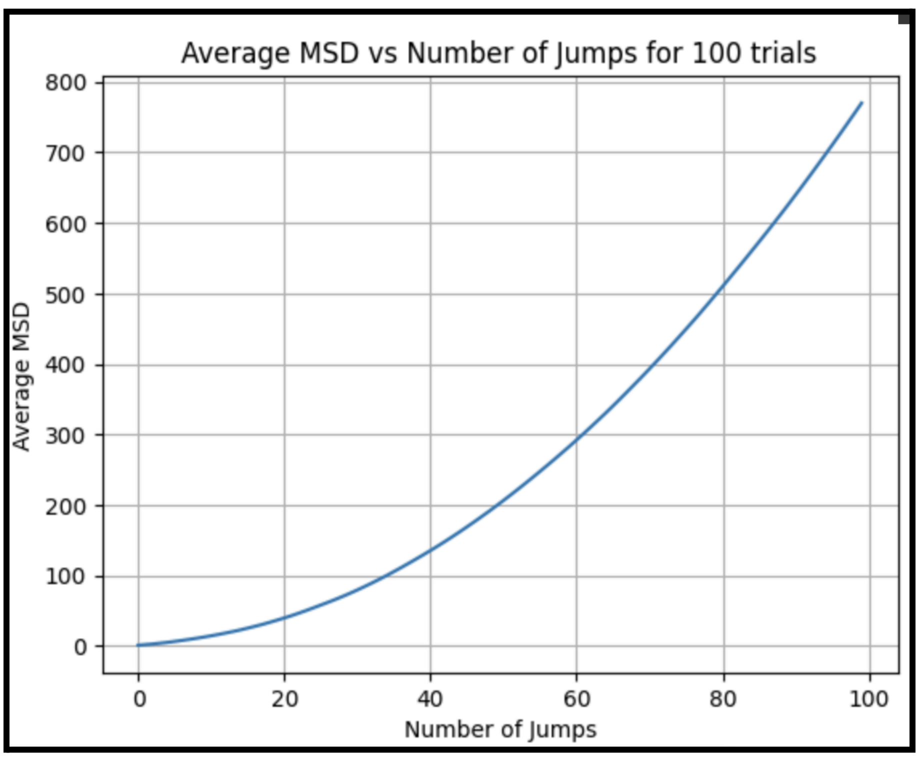

Average MSD vs Number of Jumps for 50 trials
Average MSD
Number of Jumps
0
100
200
300
400
500
600
700
800
0
20
40
60
80
100

iv. *100 Trials for 100 Jumps each*

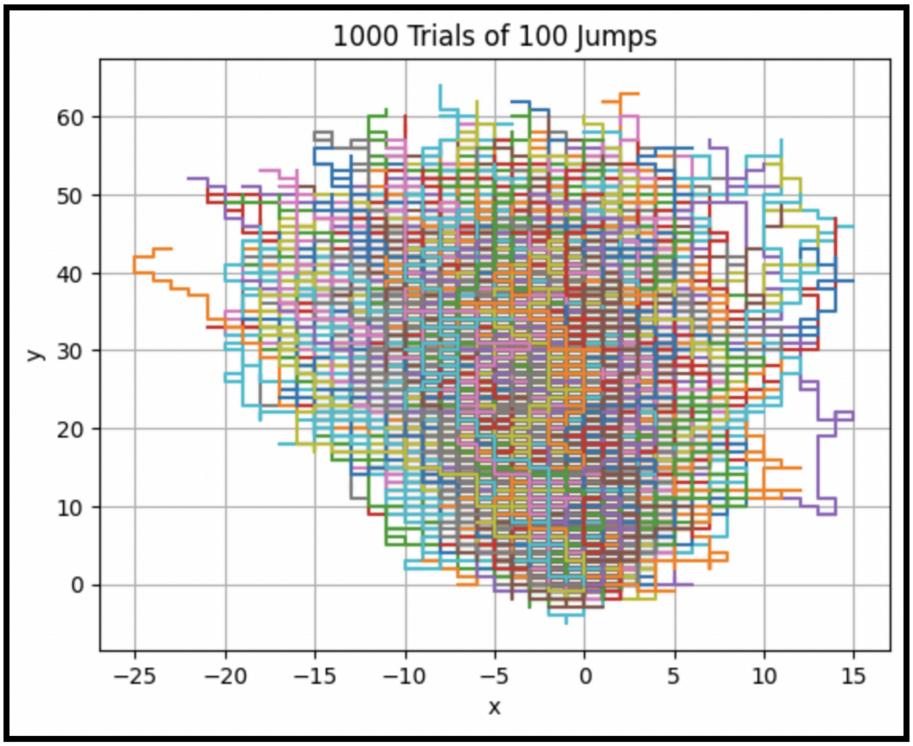


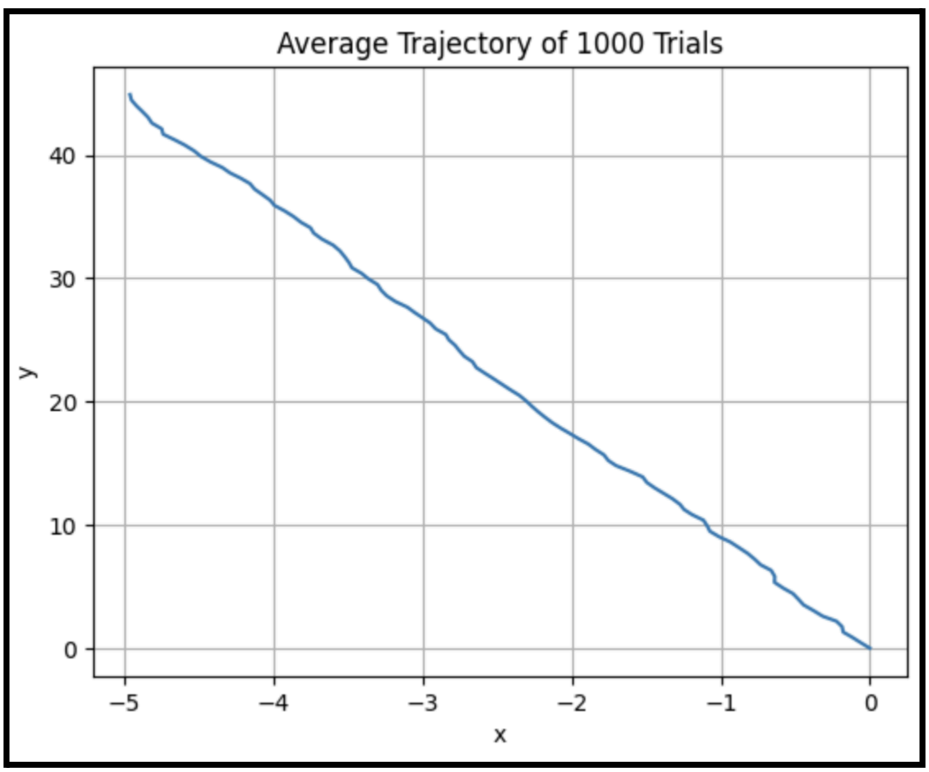


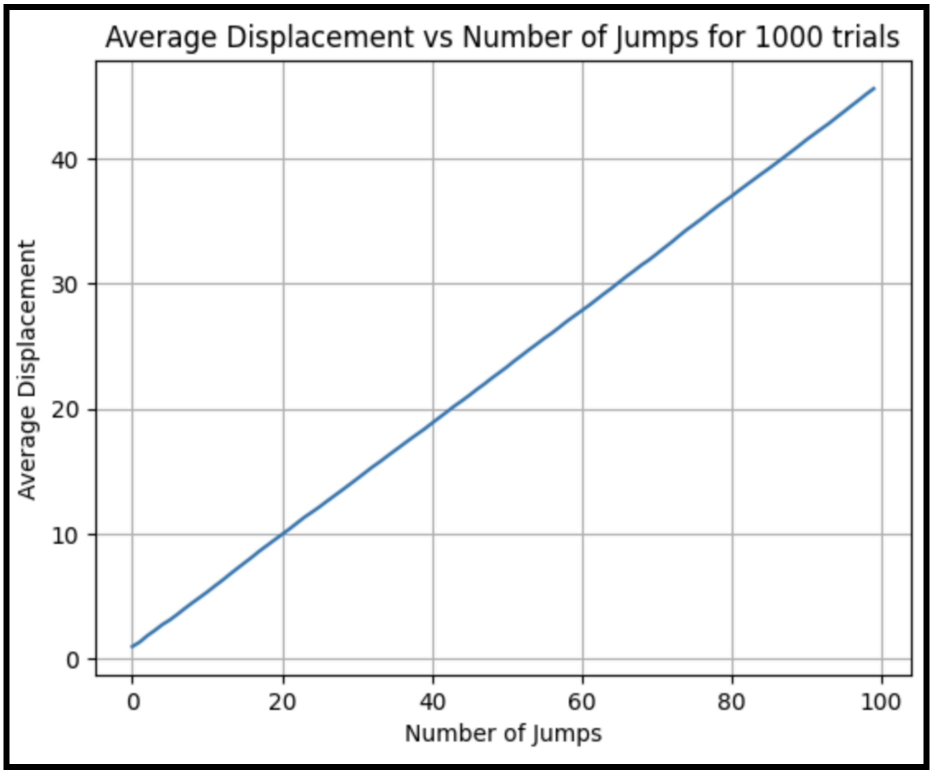

Average Displacement vs Number of Jumps for 100 trials
Average Displacement
Number of Jumps
0
10
20
30
40
0
20
40
60
80
100

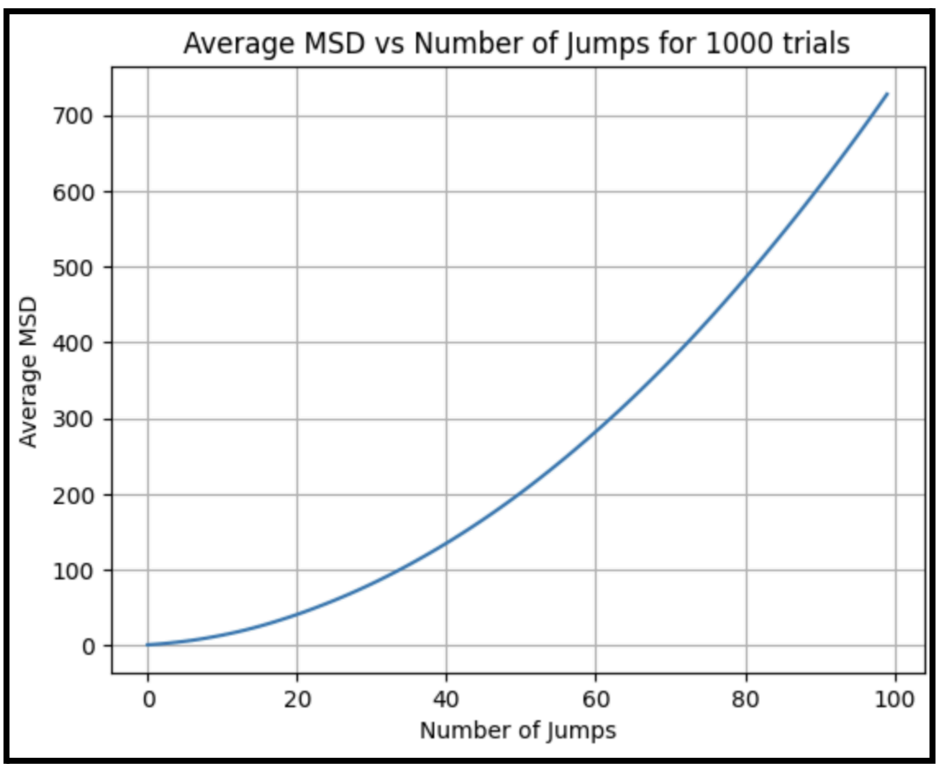

Average MSD vs Number of Jumps for 100 trials
Average MSD
Number of Jumps
0
100
200
300
400
500
600
700
800
0
20
40
60
80
100

v. *1000 Trials for 100 Jumps each*

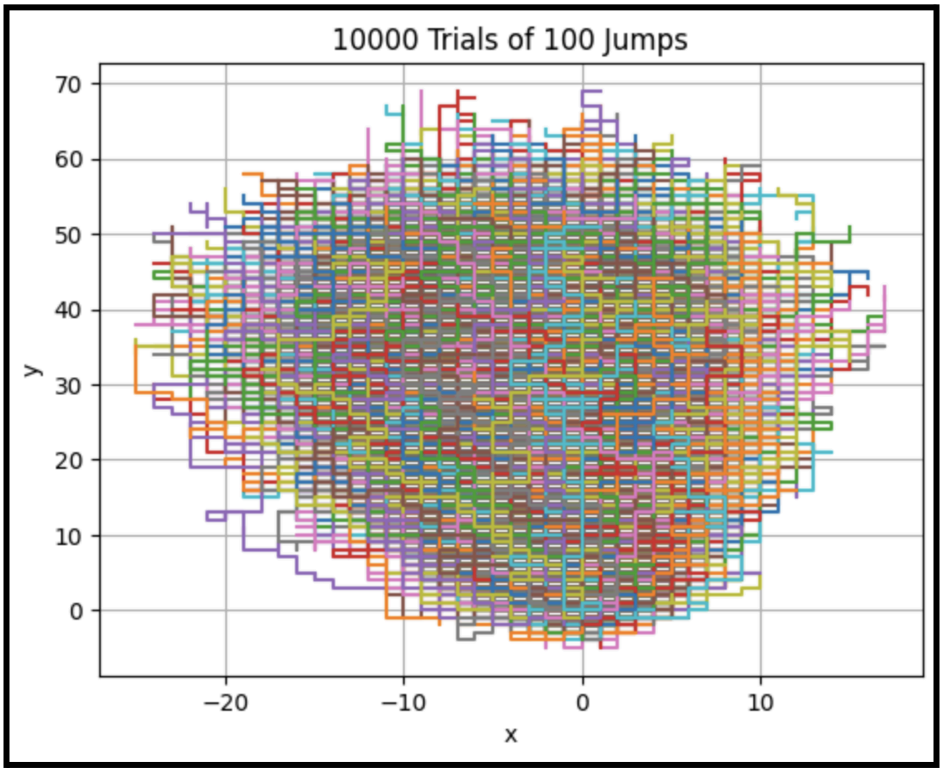


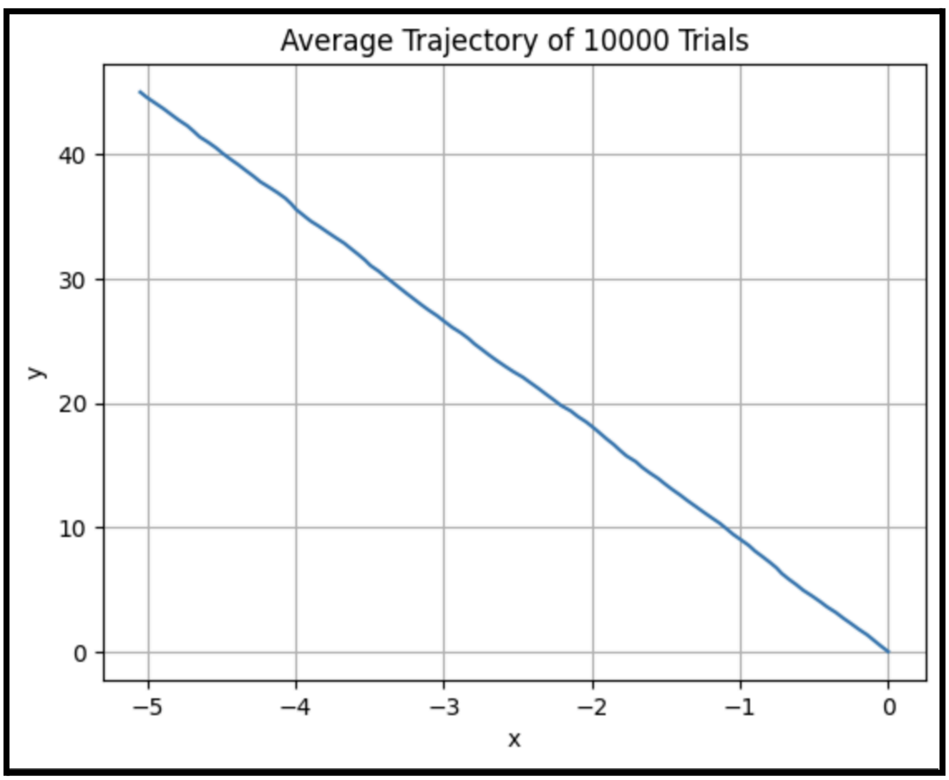


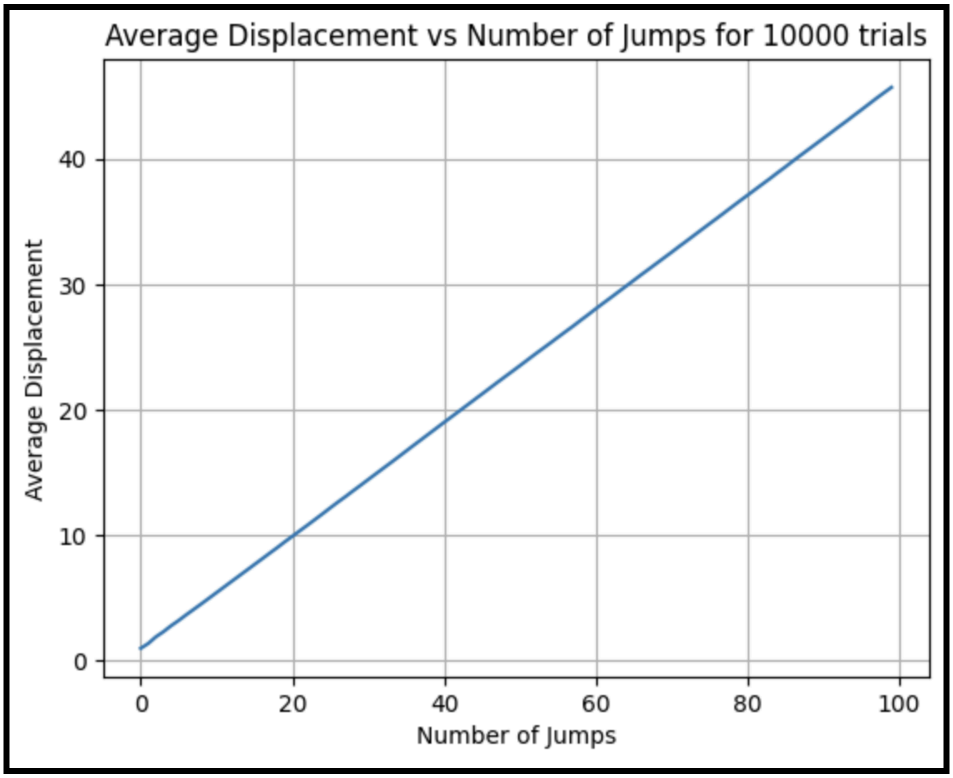

Average Displacement vs Number of Jumps for 1000 trials
Average Displacement
0
10
20
30
40
0
20
40
60
80
100
Number of Jumps

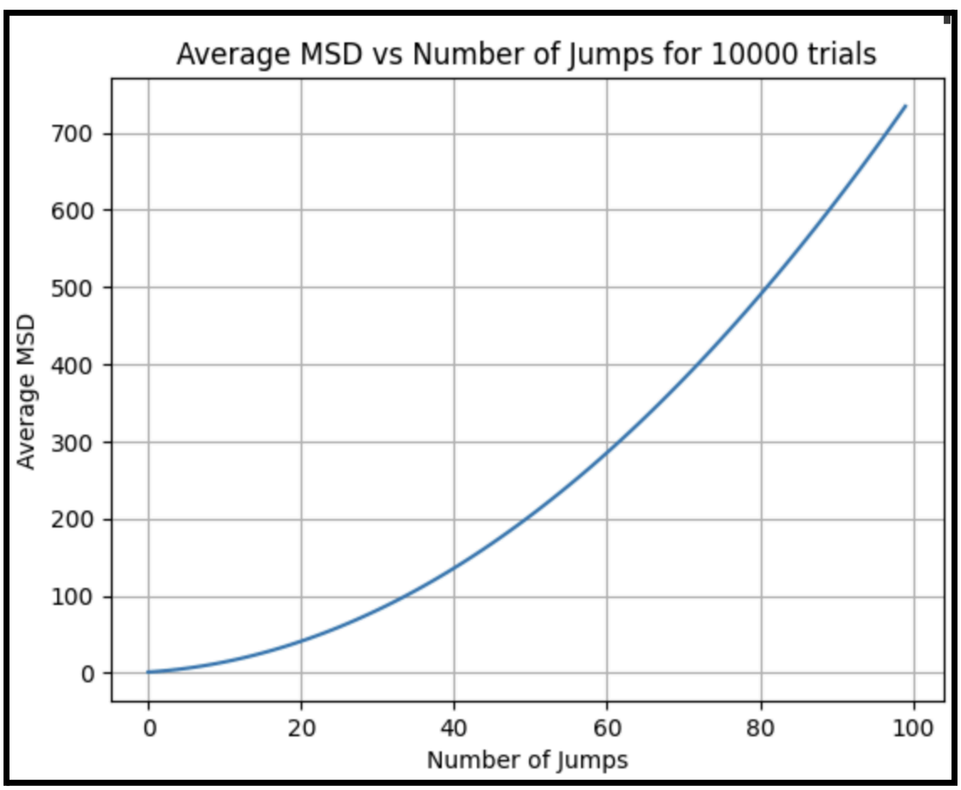

Average MSD vs Number of Jumps for 1000 trials
Average MSD
0
100
200
300
400
500
600
700
0
20
40
60
80
100
Number of Jumps

vi. *10,000 Trials for 100 Jumps each*






Average Displacement vs Number of Jumps for 10000 trials
Average Displacement
Number of Jumps
0
10
20
30
40
0
20
40
60
80
100


Average MSD vs Number of Jumps for 10000 trials
Average MSD
Number of Jumps
0
100
200
300
400
500
600
700
0
20
40
60
80
100

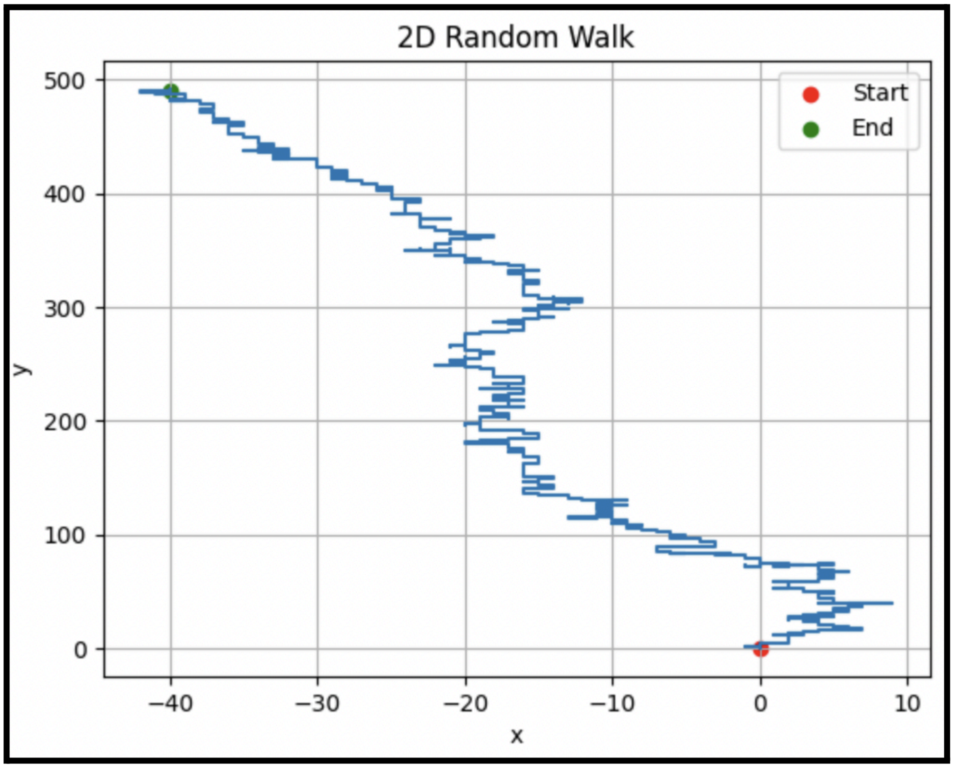
Key Observations on increasing the number of trials:

1. The graph for the average trajectory of the atom gets progressively smoother. As seen above, the average trajectory (x-y plot of the particle) is spiky without any element of smoothness when the number of trials was 5. As we reach 10,000 trials, the average trajectory smoothens and becomes almost a straight line without any spikes
2. The plot of the average displacement vs number of jumps(steps) also gets progressively smoother on increasing the number of trials. At 5 trials, it exhibits a fair amount of unevenness but at 10,000 trials, it becomes a perfectly straight line.
3. The plot for average mean squared displacement vs number of jumps(steps) remains consistent regardless of the number of trials

IV. Changes observed in the Random Walk as biassing probability is varied as a fraction of random probability.

To conclude how the random walk changes in this case, a number of different cases are taken where the probability of the atom moving in a particular direction is different, determined by the biassing probability.

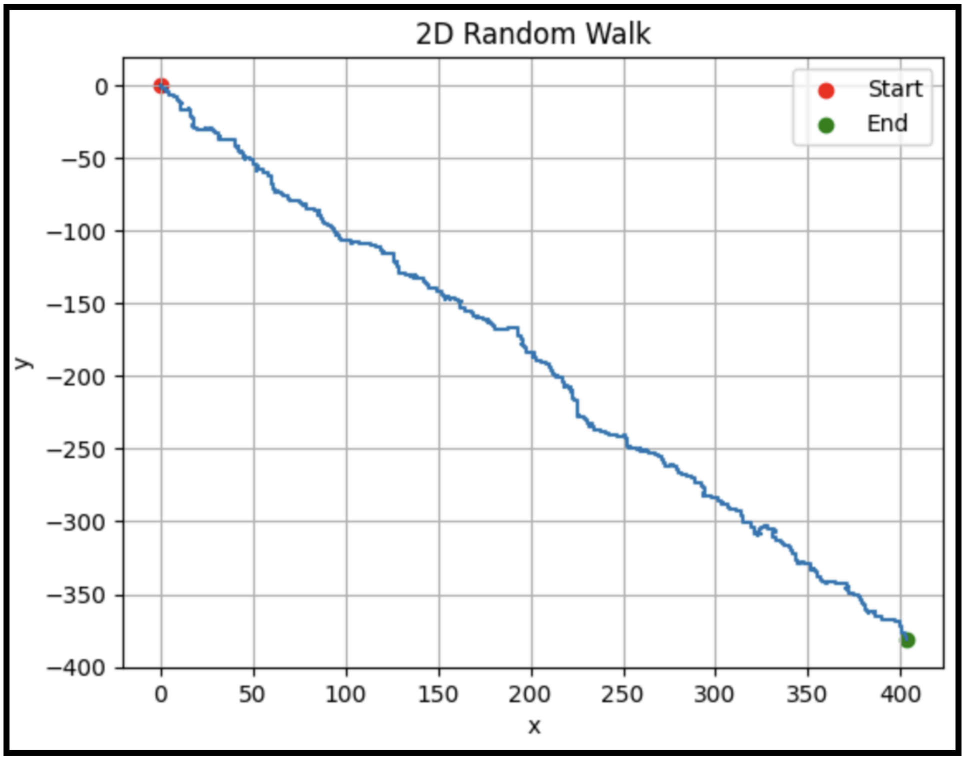
i. **Case 1: P(right, left, down, up) = (0.15, 0.20, 0.10, 0.55)**



In this case, the net movement of the particle is along the +y (up) and -x (left) direction as ‘up’ and ‘left’ movement is dominant as suggested by their respective biassing probabilities (0.55 and 0.20).

The movement along +y is much more than movement along -x as the biassing probability for ‘up’ movement (0.55) is more than biassing probability for ‘left’ movement (0.20) but both of them together dominate over 'right' and 'down' movements as P(left + up) = 0.75 which is more than P(right + down) = 0.25

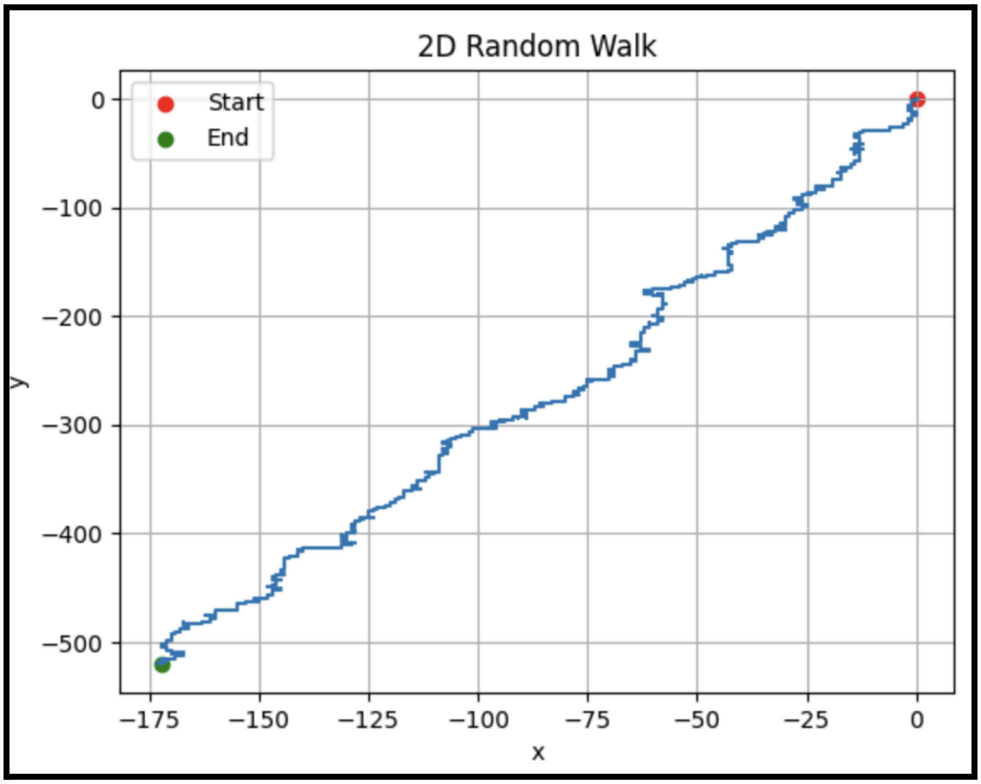
ii. **Case 2: P(right, left, down, up) = (0.45, 0.05, 0.45, 0.05)**



In this case, the net movement of the particle is along the -y(down) and +x(right) direction as 'down' and 'right' movement is dominant as suggested by their respective biassing probabilities (0.45 and 0.45).

The movement along -y is equal to movement along +x as the biassing probability for 'down' movement (0.45) is equal to the biassing probability for 'right' movement (0.45) but both of them together dominate over 'up' and 'left' movements as P(down + right) = 0.90 which is more than P(up + left) = 0.10

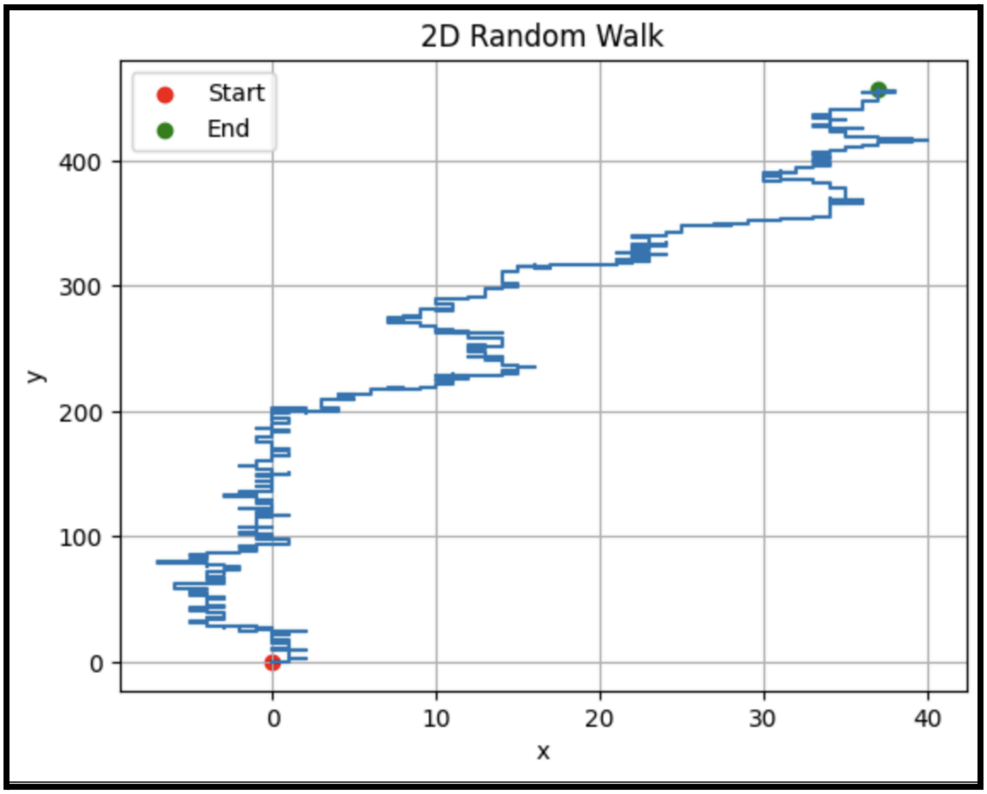
iii. **Case 3: P(right, left, down, up) = (0.10, 0.25, 0.60, 0.05)**



As we can observe, the net movement of the particle is along the -y(down) and -x(left) direction as ‘down’ and ‘left’ movement is dominant as suggested by their respective biassing probabilities (0.60 and 0.25).

The movement along -y is greater than movement along -x as the biassing probability for ‘down’ movement (0.60) is greater than the biassing probability for ‘left’ movement (0.25) but both of them together dominate over ‘up’ and ‘right’ movements as P(down + left) = 0.85 which is more than P(up + right) = 0.15

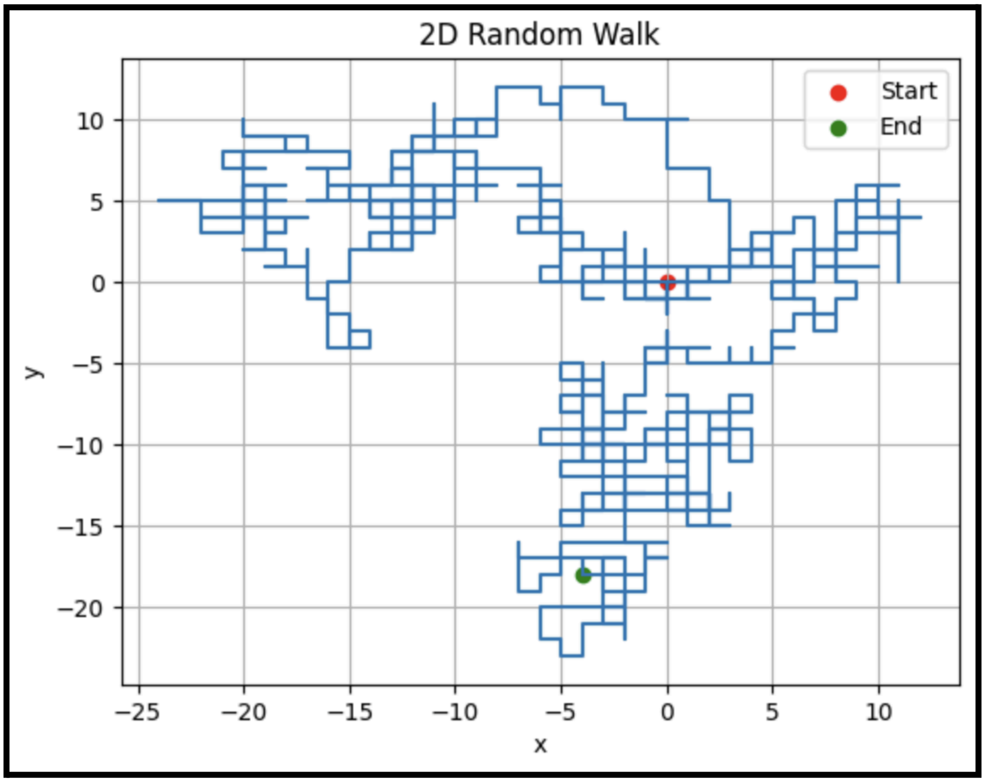
iv. **Case 4: P(right, left, down, up) = (0.20, 0.15, 0.10, 0.55)**



As we can observe, the net movement of the particle is along the +y(up) and +x(right) direction as 'up' and 'right' movement is dominant as suggested by their respective biassing probabilities (0.55 and 0.20).

The movement along +y is greater than movement along +x as the biassing probability for 'up' movement (0.55) is greater than the biassing probability for 'right' movement (0.20) but both of them together dominate over 'down' and 'left' movements as P(up + right) = 0.75 which is more than P(up + right) = 0.25

Case 5: **P(right, left, down, up) = (0.25, 0.25, 0.25, 0.25)**



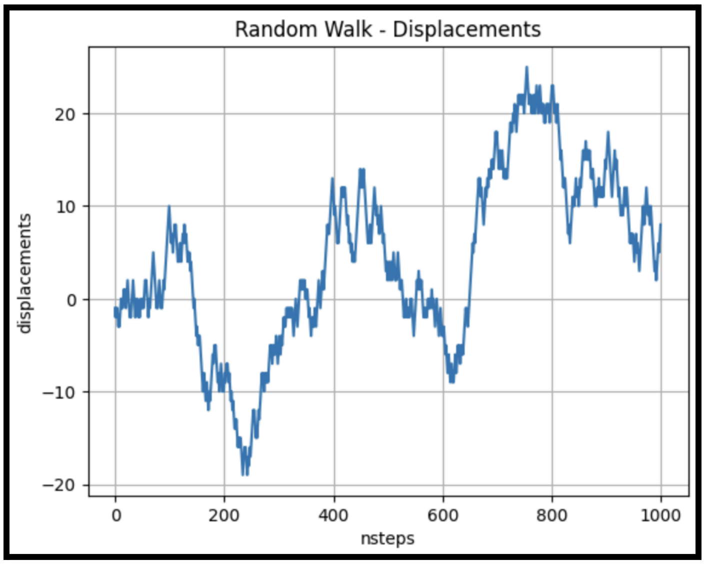
Since the probability of the atom moving along any given direction is the same, the movement of the atom appears completely random where no particular direction of movement is favoured.

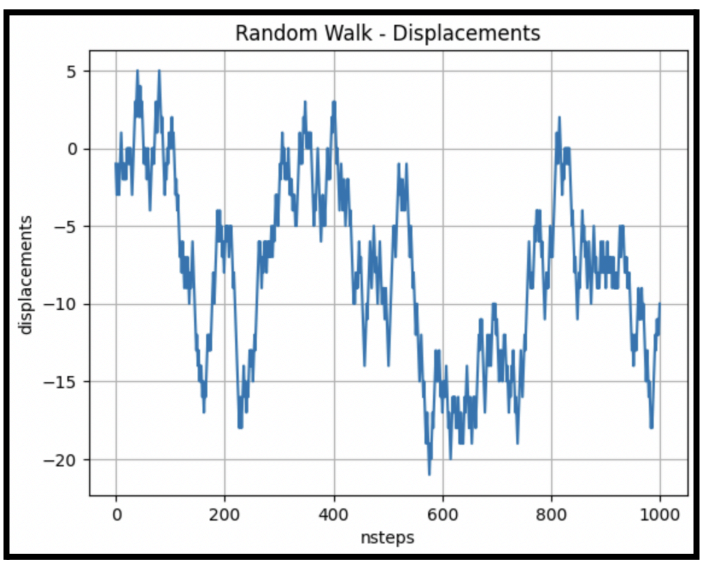
V. A Special Case of Unbiased Random Walk: Number of Jumps is 5 times the Number of Lattice Sites.

To analayze this case, the results of the Random Walk are shown side by side. The upper one before implementing and the lower one after implementing the condition.

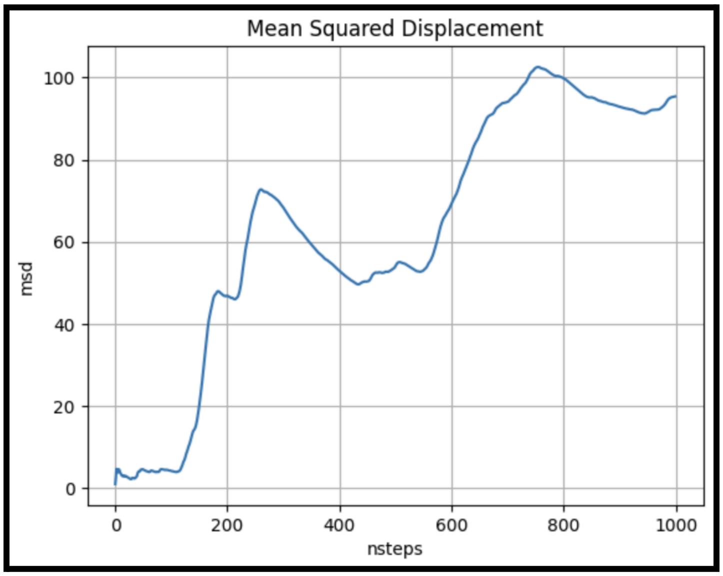
1. ***One Dimensional Case***

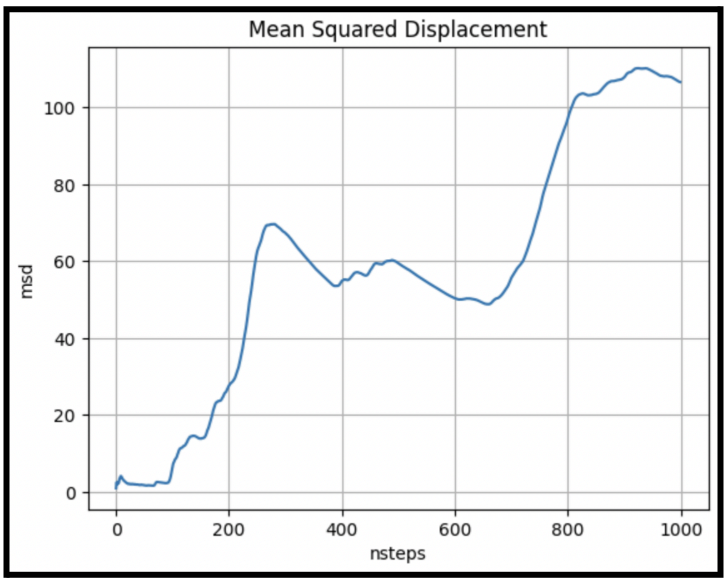
Displacement vs Time





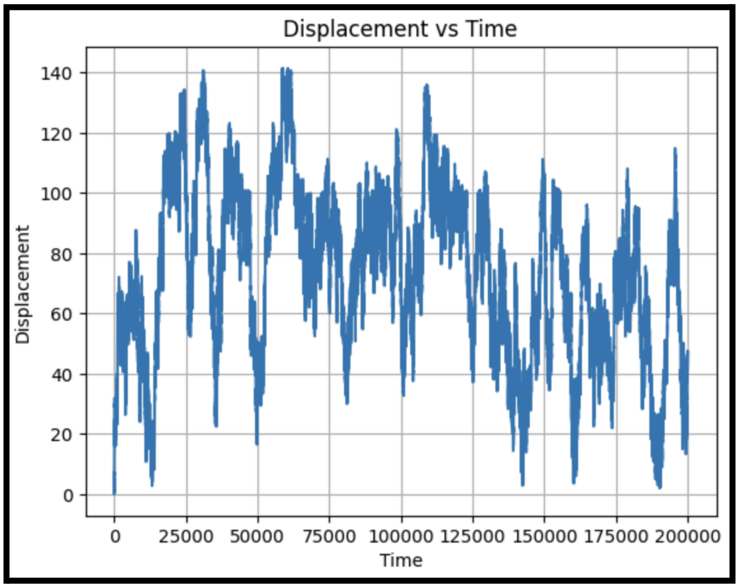
MSD vs Time

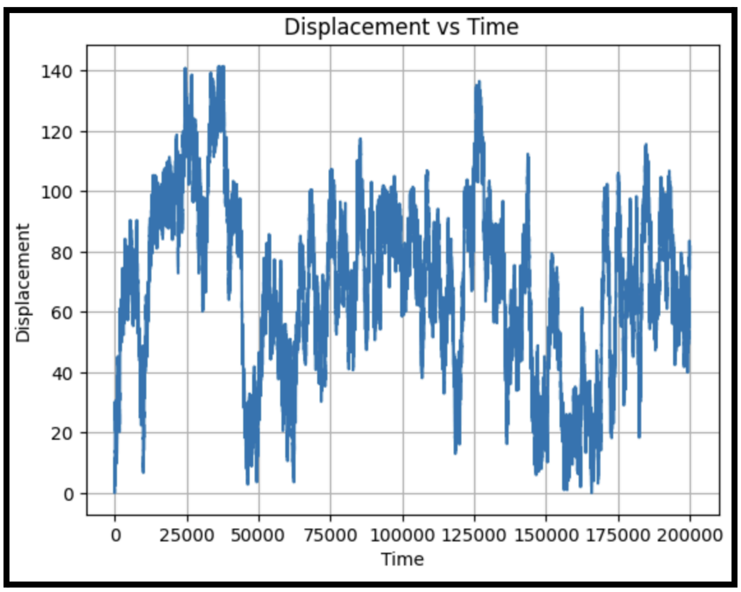

Mean Squared Displacement
msd
nsteps
0
20
40
60
80
100
0
200
400
600
800
1000


Mean Squared Displacement
msd
nsteps
0
20
40
60
80
100
0
200
400
600
800
1000

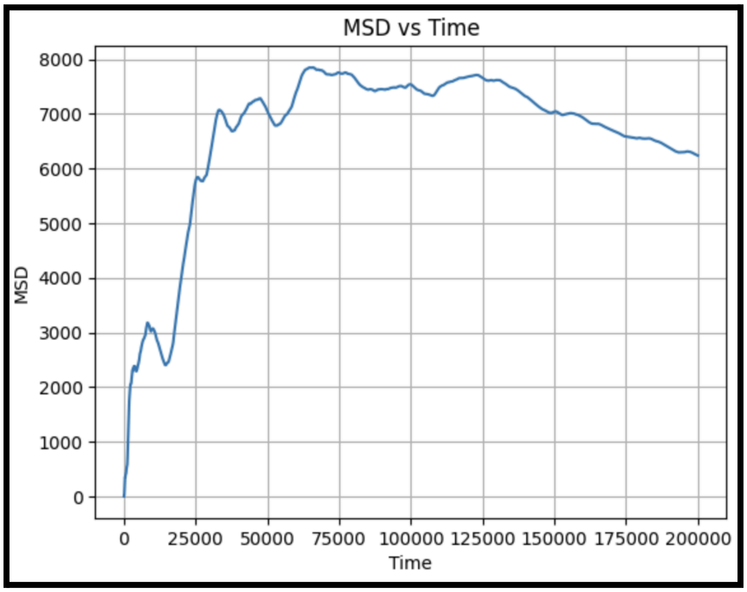
2. Two Dimensional Case

Displacement vs Time

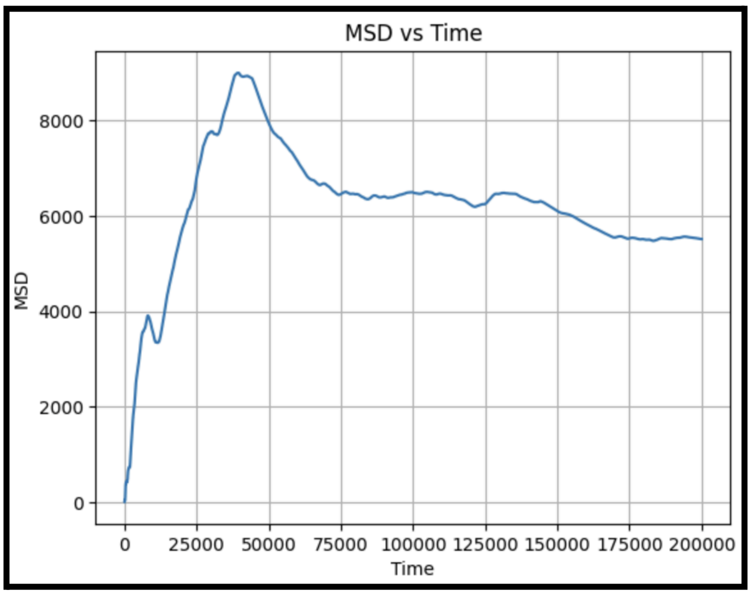




MSD vs Time

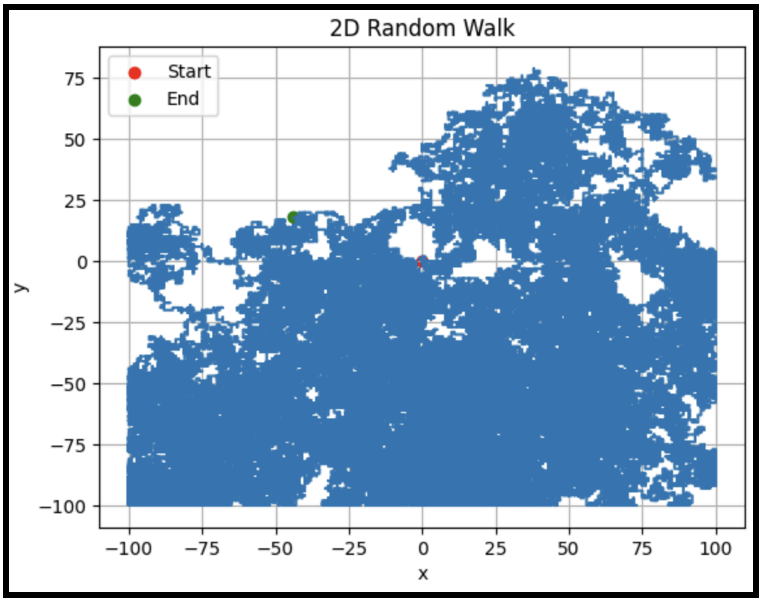


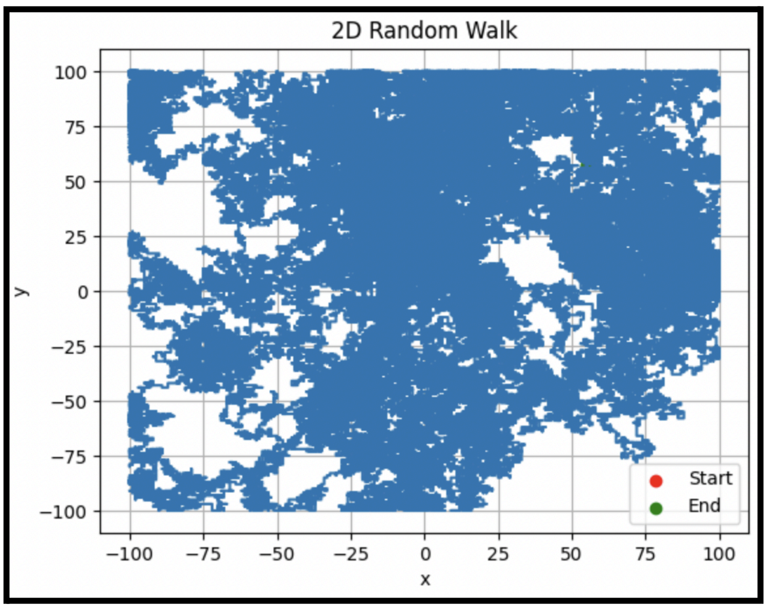
MSD vs Time
8000
7000
6000
5000
4000
3000
2000
1000
0
MSD
0
25000
50000
75000
100000
125000
150000
175000
200000
Time



MSD vs Time
8000
6000
4000
2000
0
MSD
0
25000
50000
75000
100000
125000
150000
175000
200000
Time

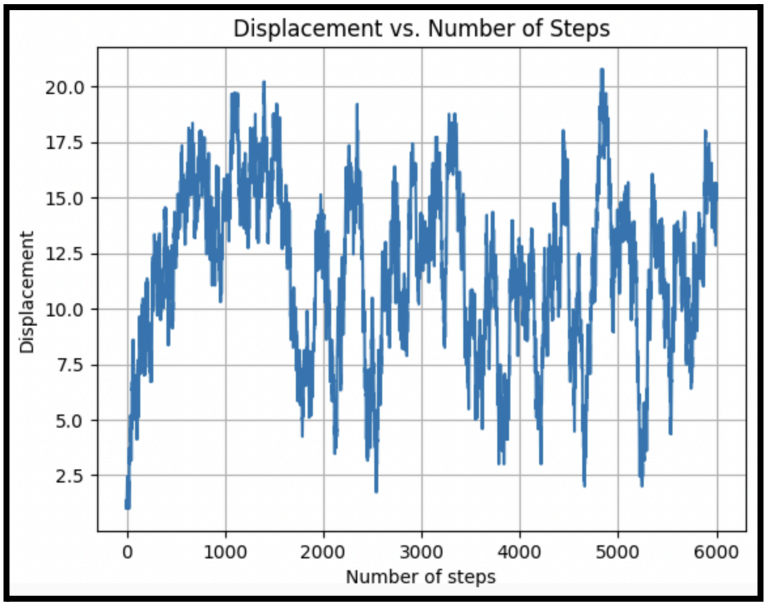
Trajectory of the Atom in XY plane

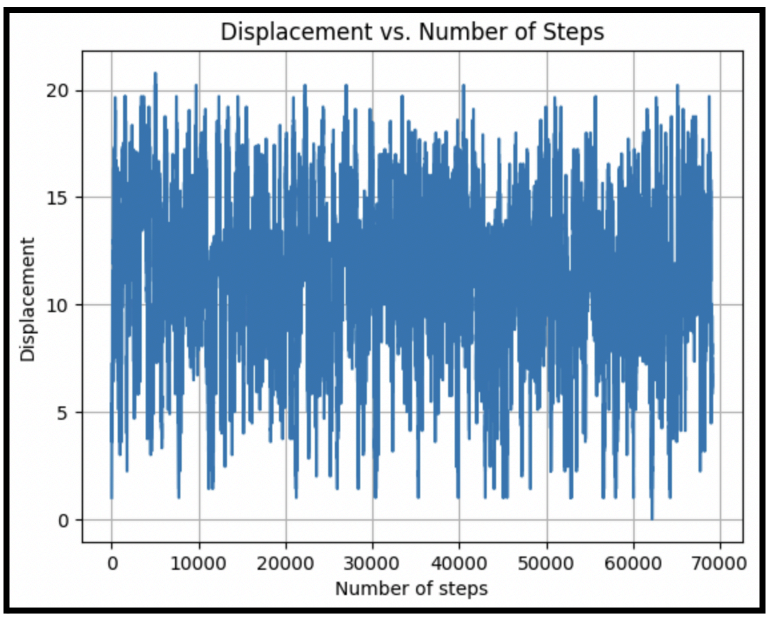




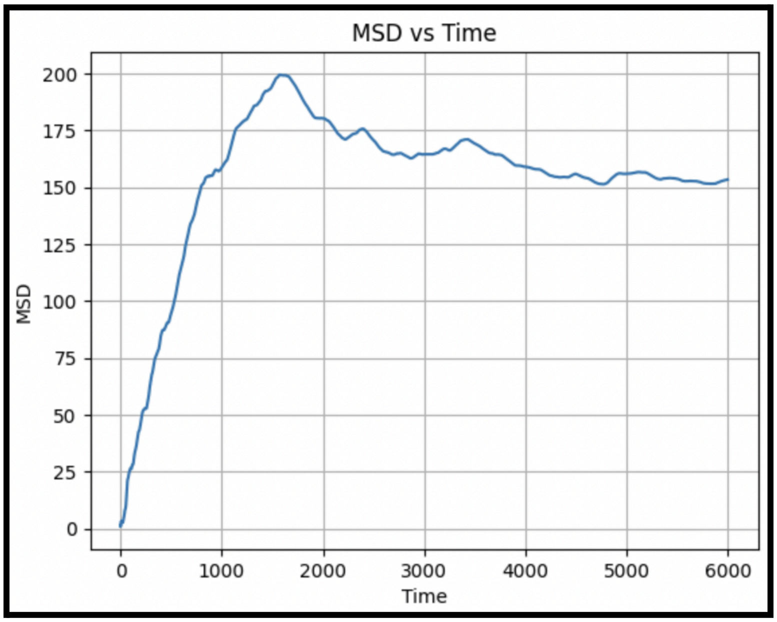
3. Three Dimensional Case

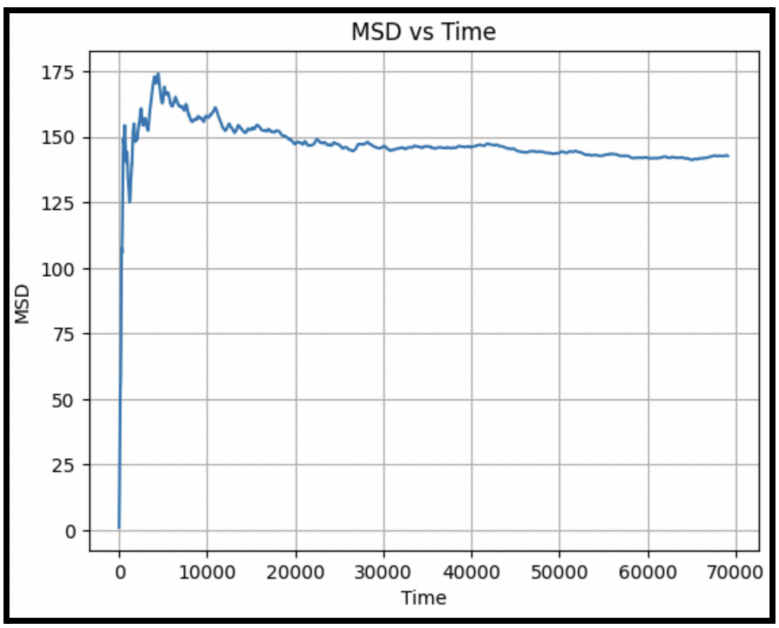
Displacement vs Time



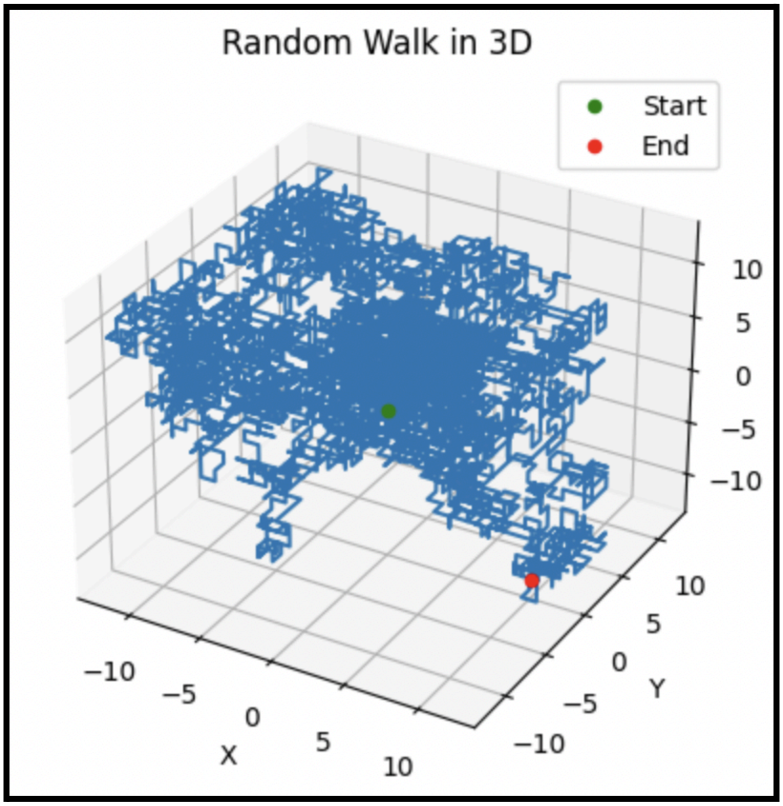


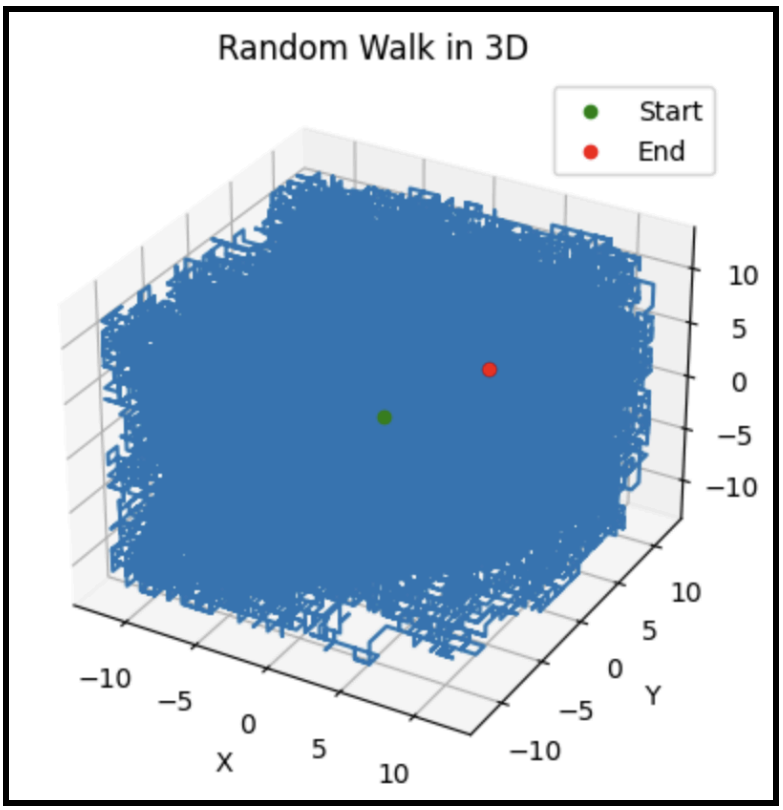
MSD vs Time


MSD vs Time
MSD
Time
0
25
50
75
100
125
150
175
200
0
1000
2000
3000
4000
5000
6000


MSD vs Time
MSD
Time
0
25
50
75
100
125
150
175
0
10000
20000
30000
40000
50000
60000
70000

Trajectory of the Atom





Observations:

1. *Unlike the previous case where an infinite lattice was used to simulate the Random Walk, the Displacement vs Time plot does not exhibit an overall increasing trend (does not progressively increase even though spiky) as the lattice size is finite and the atom eventually collides with the lattice boundaries making further increase in displacement impossible.*
2. *The Displacement vs Time plot is more spiky than the plot for an infinite lattice as the atom repeatedly collides with the lattice boundaries.*
3. *When the number of steps becomes 5 times the number of lattice sites, the Displacement vs Time Plot tends to become more spiky as the number of collisions of the atom with the lattice boundary increases.*
4. *When the number of jumps is 5 times the number of lattice sites, the atom explores the given lattice space more completely than the case when this condition is not implemented. This is seen when comparing the atom trajectories for the two cases.*

VI. Key Challenge/Limitation faced while Coding the Random Walk and Running the Code

The key challenge faced was Computational Complexity and Time

This was observed when the results of a random walk for a certain number of steps had to be averaged over a given number of trials to plot the average displacement vs steps and average mean squared displacement vs steps graphs. As the number of trials increased, the execution time for that particular block of code also increased leading to lower computational efficiency and increased time complexity.

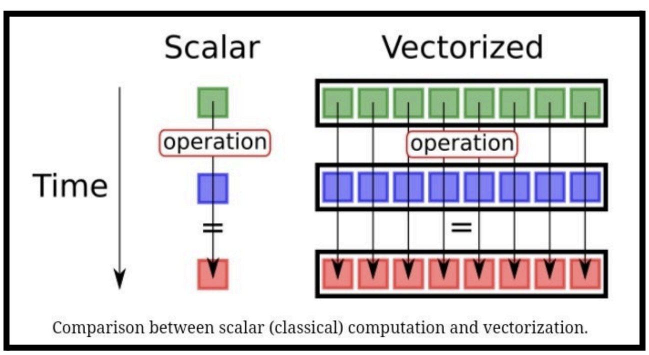
```
trajectories = []
displacements_array = []
msd_array = []
for _ in range(ntrials):
  positions, displacements = walk(init_pos, njumps)
  displacements_array.append(displacements)
  trajectories.append(positions)
  displacements = np.array(displacements)
  msd = np.cumsum(displacements**2)/np.arange(1,njumps+1)
  msd_array.append(msd)
```

The time complexity in this case is O(n).
If the displacements array had to be iterated again to calculate the mean squared displacement, the time complexity would have been an even higher O(n^2) leading to even longer execution periods and lower computational efficiency.

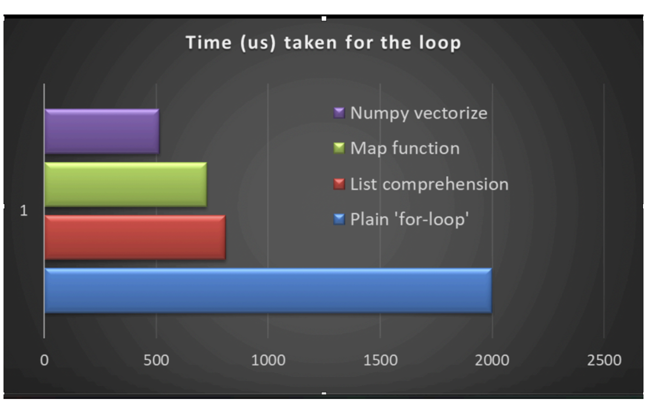
Workaround implemented: To reduce the time complexity from O(n^2) to O(n) the implementation was vectorized using numpy arrays. The displacement python list was converted to a numpy array and mean squared displacement was calculated using numpy methods which are more efficiently implemented behind the scenes using vectorization.

```
displacements = np.array(displacements)
msd = np.cumsum(displacements**2)/np.arange(1,njumps+1)
```

Vectorization in NumPy refers to the process of replacing explicit loops in Python with array operations that are internally implemented in optimized C code. This enables faster execution by taking advantage of low-level operations and parallelism. Instead of iterating element-by-element in Python, vectorized operations allow the entire array to be processed in one go, improving both performance and readability.



Comparison between scalar (classical) computation and vectorization.



CONCLUSIONS

Key conclusions based on the results of running the codes:

I. Unbiased Random Walk Simulations:

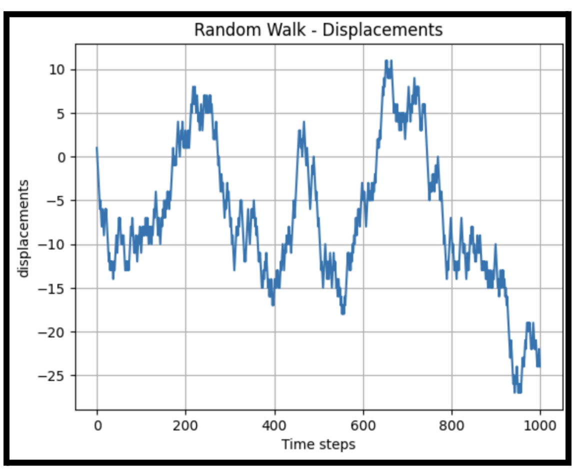
In this case, the atom has no restriction to move in a certain direction. The atom has equal probability to move along any given direction. Progressive increase in distance is more prominent as the degree of freedom of the atom increases.

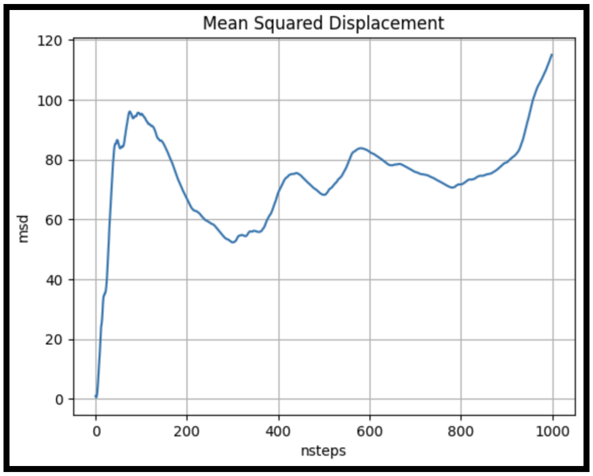
The Displacement vs Time Plots are spiky due to the random natures of the process

The MSD vs Time plots are smooth and their overall shape resembles that of the Displacement vs Time Plots. Their smooth nature is due to the cumulative squared averaging process.

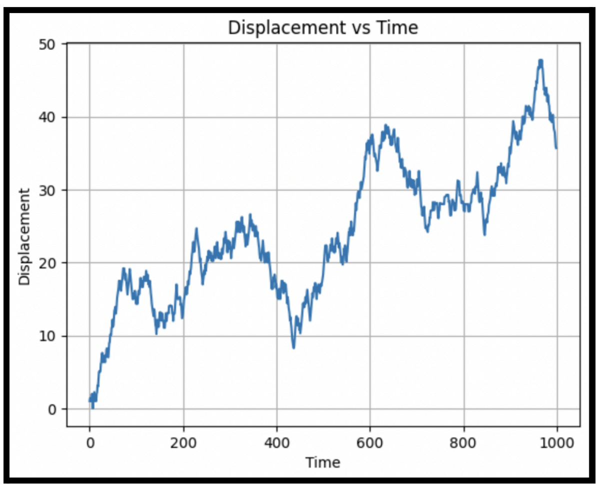
The results are summarised below:

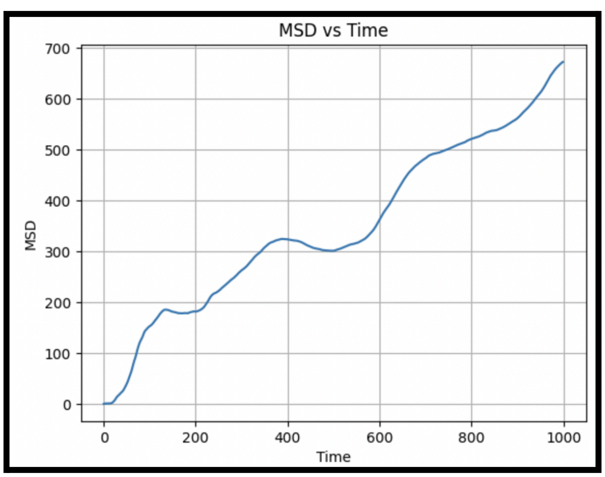
i. One Dimensional Unbiased Random Walk

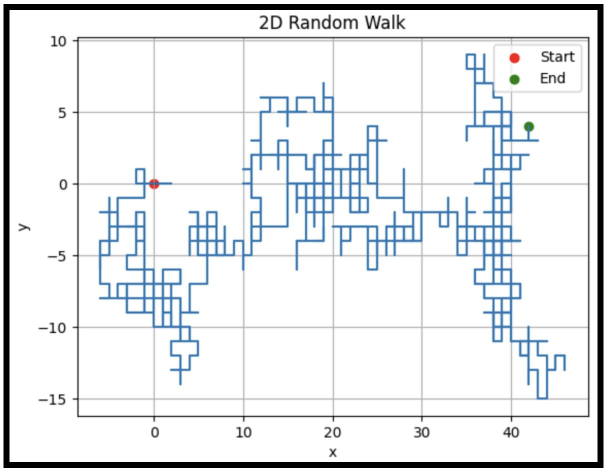


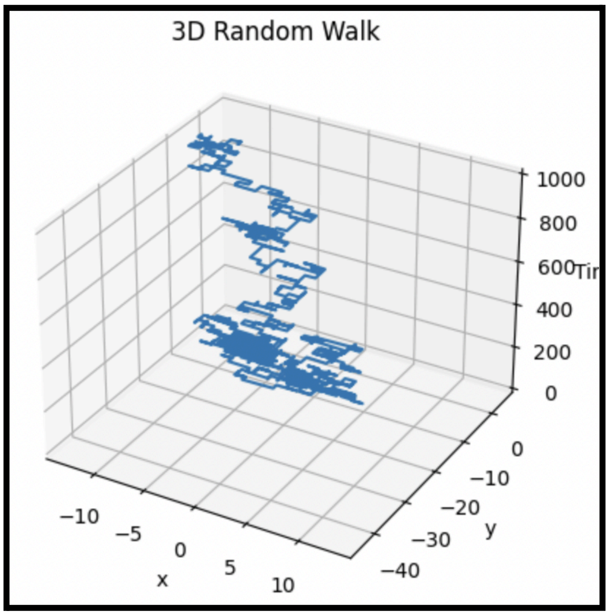


ii. Two Dimensional Random Walk

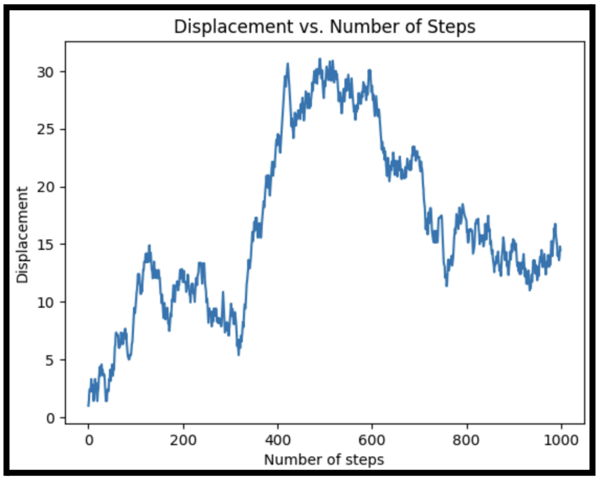


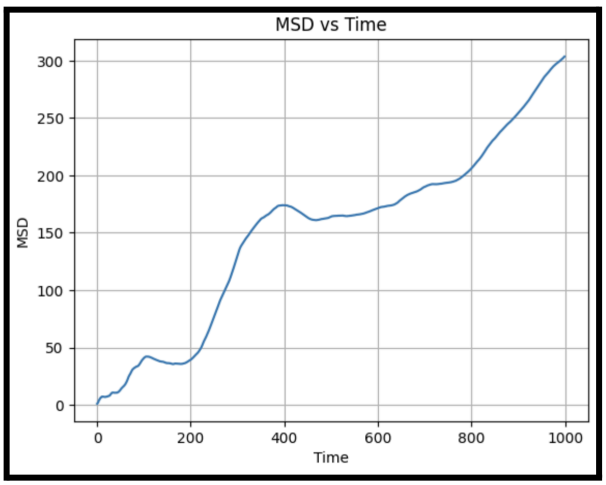


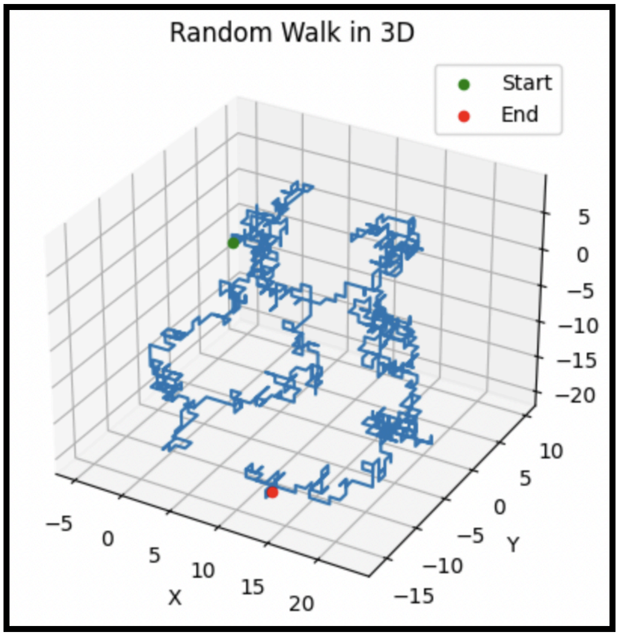

2D Random Walk
Start
End
x
y
10
5
0
−5
−10
−15
0
10
20
30
40


3D Random Walk
x
y
−10
−5
0
5
10
0
−10
−20
−30
−40
1000
800
600
400
200
0

iii. Three Dimensional Random Walk

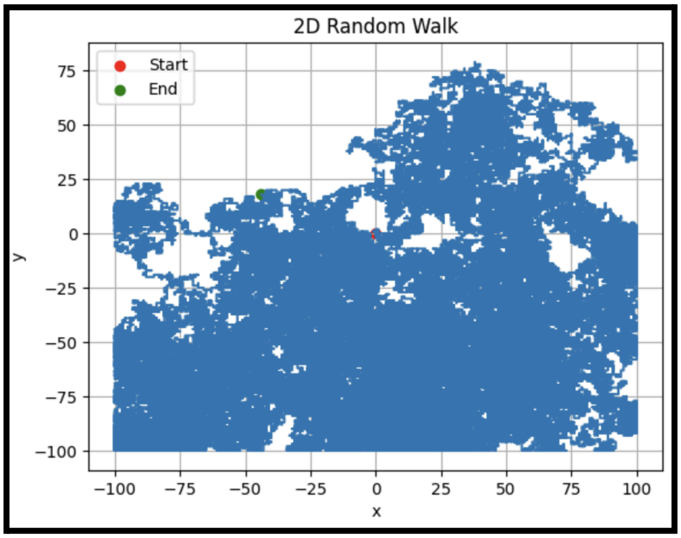




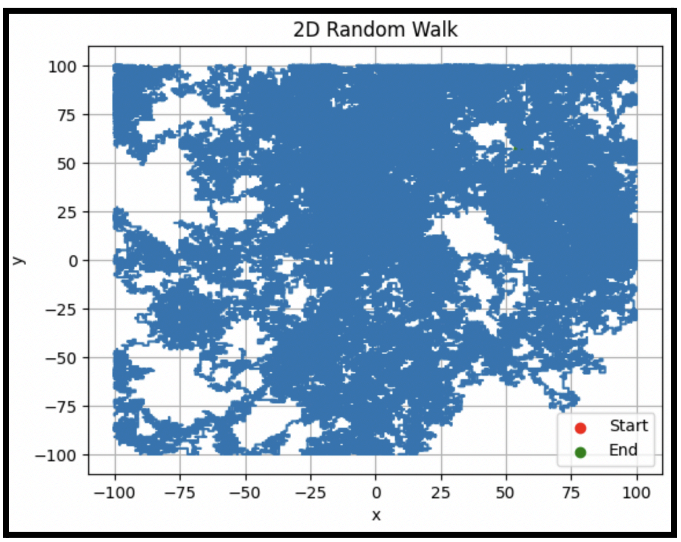


When the lattice is made finite and the number of steps is made equal to five times the number of lattice sites, the spikiness of the Displacement vs Time plot tends to increase as the magnitude of displacements is limited by the lattice boundaries and collision of the atoms with the lattice boundary. So the displacement tends to fluctuate about a single value. When the trajectory of the atom is examined in this case, it is observed that the atom more completely explores the lattice space when compared to the case when the number of jumps is much less than 5 times the number of lattice sites.

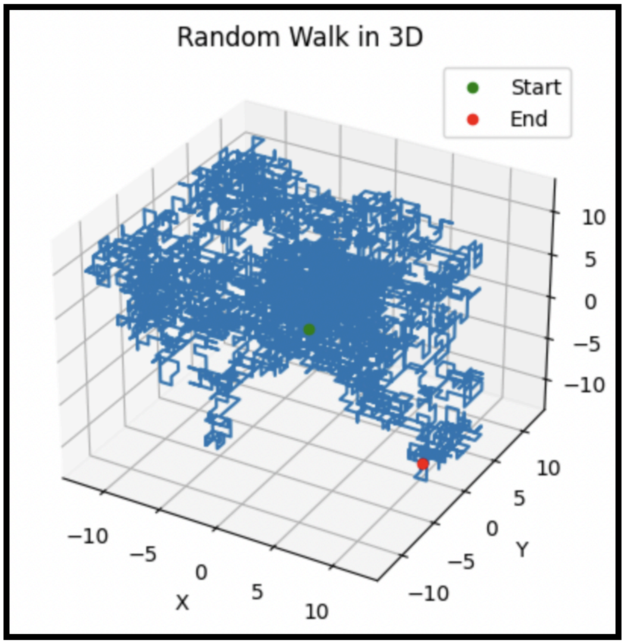
2D Random walk ***<u>without</u>*** *implementing the edge case: No of jumps = 5 times the lattice sites*



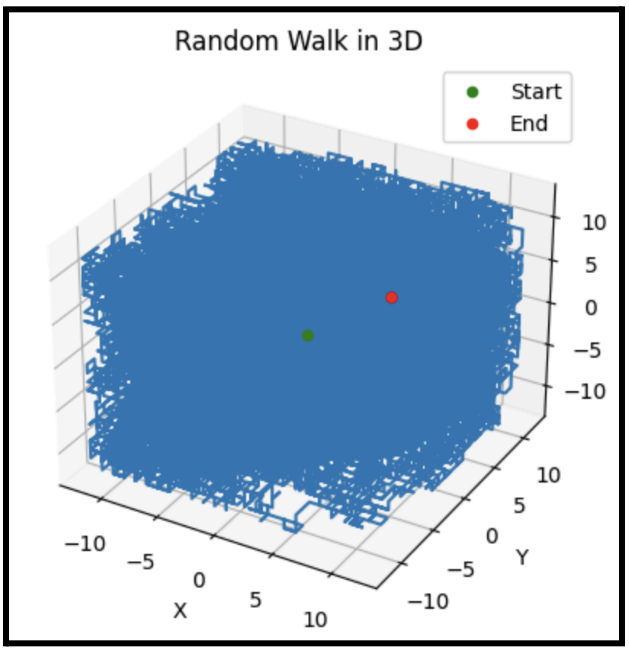
2D Random walk ***<u>after</u>*** *implementing the edge case: No of jumps = 5 times the lattice sites*



3D Random walk ***<u>without</u>*** *implementing the edge case: No of jumps = 5 times the lattice sites*



3D Random walk ***after*** *implementing the edge case: No of jumps = 5 times the lattice sites*



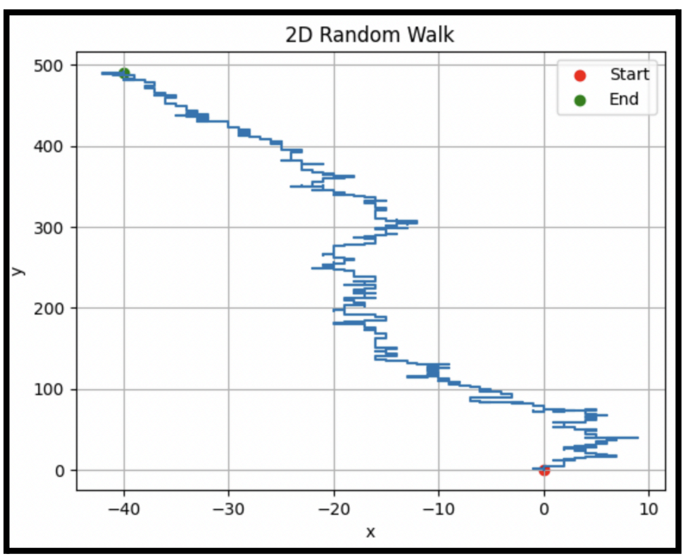
II. Biased Random Walk Simulations

In this case, the probability of the atom moving along different directions is different. The biasing probabilities for the different directions should add up to 1. Depending on the biasing probabilities along different directions, the dominant direction of motion canbe determined.

For example: if P(up) and P(right) together is more dominant (ie P(up) + P(right) > P(down) + P(left)), the net displacement is along the upper right direction.

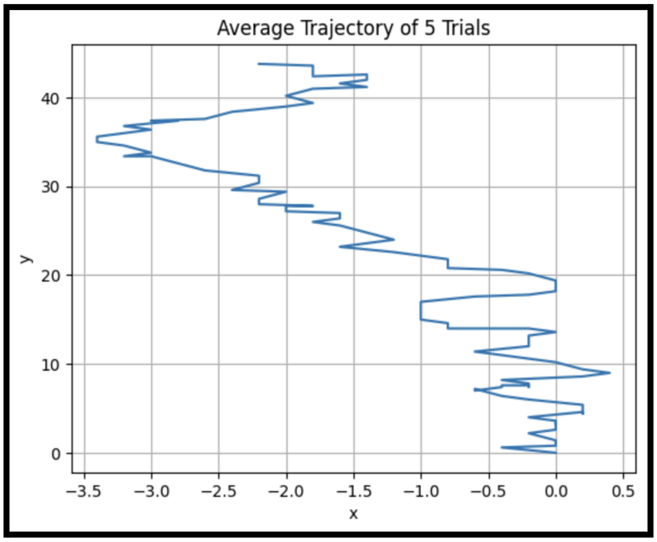
The code was run and analysed for the condition:

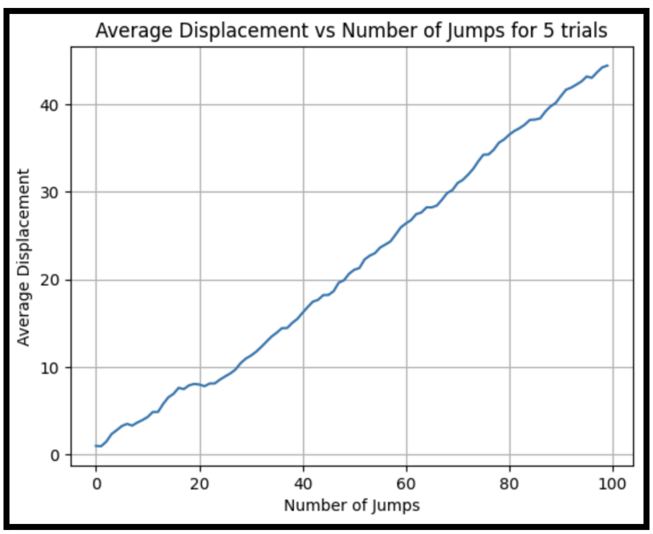
P(right, left, down, up) = (0.15, 0.20, 0.10, 0.55)

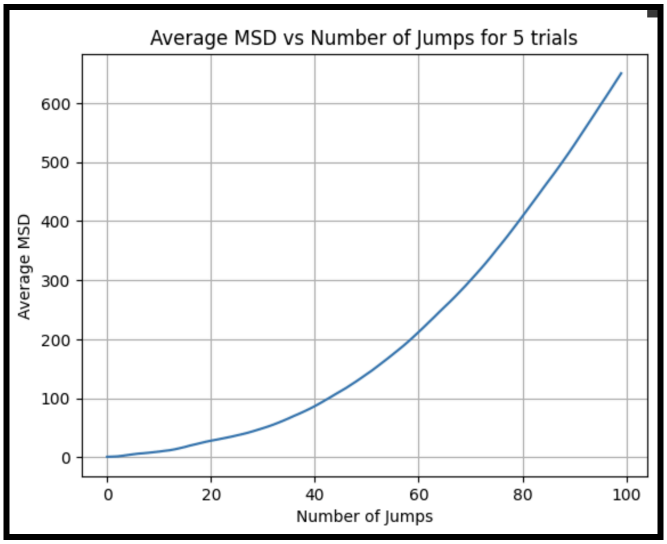


When biased random walk simulation is averaged over a certain number of trials and the average trajectory and average displacement vs time graphs are plotted, the plots are found to be undulating. As the number of trials are increased, the plots get smoother due to the averaging process. As more terms are included to average, smoother is the plot.

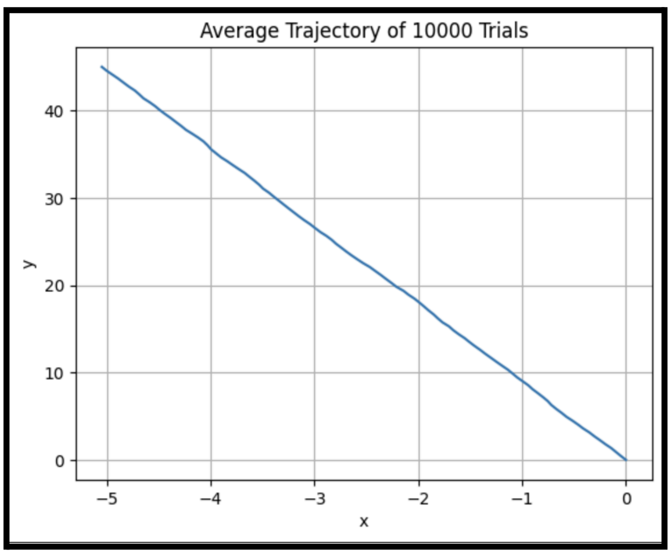
i. 5 Trials for 100 Jumps Each

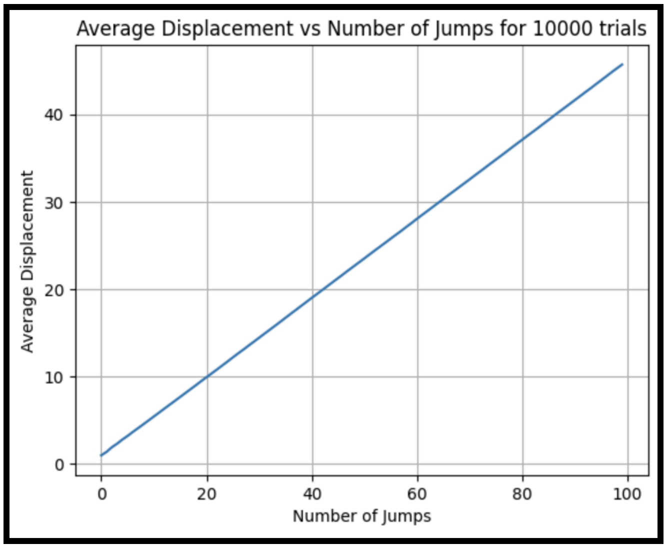


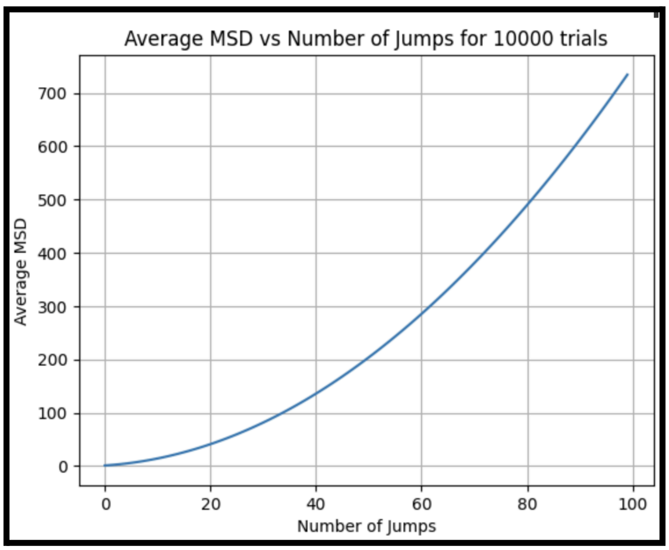




ii. 10,000 Trials for 100 Jumps Each







When the Average Displacement vs Time and Average Trajectory plots for 5 trials and 10,000 trials are compared, the plots for 10,000 trials are significantly smoother.

There is no change in the nature of the MSD vs Time plot irrespective of the number of trials. This is because calculation of MSD itself is a cumulative squared averaging process. It is already an averaged quantity and further averaging it over a certain number of trials won't make any difference. The key point is that MSD is less sensitive to individual fluctuations in a single trial because it represents the average squared displacement at **each** time step. Therefore, even when MSD is computed for many trials, more data points are added which behave in a similar manner.

On the other hand, the calculation of displacements for each trial is not an averaged quantity so the Displacement vs Time plot is spiky as there is significant variation in displacement at each step but when displacement is averaged out over a certain number of trials, the plot is considerably smoothened. More the number of trials, smoother is the plot as more terms are included in the averaging process.

REFERENCES:

1. https://www.geeksforgeeks.org/a-search-algorithm/
2. https://www.britannica.com/technology/civil-engineering/Construction
3. https://www.investopedia.com/terms/r/randomwalktheory.asp
4. https://www.uio.no/studier/emner/matnat/kjemi/KJM5120/v05/undervisningsmateriale/KJM5120-Ch5-Diffusion.pdf
5. https://prancer.physics.louisville.edu/modules/random_walks/index.html#:~:text=For%20a%20random%20walk%20in,go%20either%20left%20or%20right.