Computational Physics Homework Report

Fateme RamezanZade-99100693 Problem Set 5

1. Some Relations for One Dimensional Random Walk

1.1 Theory

We want to prove that for a linear random walk, $\sigma^2 = \frac{4l^2}{\tau} pqt$. We start with an identity for σ:

$$\sigma^2 = < x^2 > - < x >^2$$

On the other hand, we know from previous calculations that $< x(t) >= \frac{lt}{\tau}(p-q)$. So we go on with calculating $< x^2 >$:

$$< x^2(t) \geq < x^2(t-\tau) + a^2 l^2 + 2ax(t-\tau)l \geq < x^2(t-\tau) > + l^2 + \frac{2l^2}{\tau}(p-q)^2(t-\tau)$$

$$\Rightarrow < x^2(t) \geq \sum_{n=0}^{N-1} l^2 + \frac{2l^2}{\tau}(p-q)^2(t-\tau) = \sum_{n=0}^{N-1} l^2 + \frac{2l^2}{\tau}(p-q)^2(n\tau) = Nl^2 + l^2(p-q)^2 N(N-1)$$

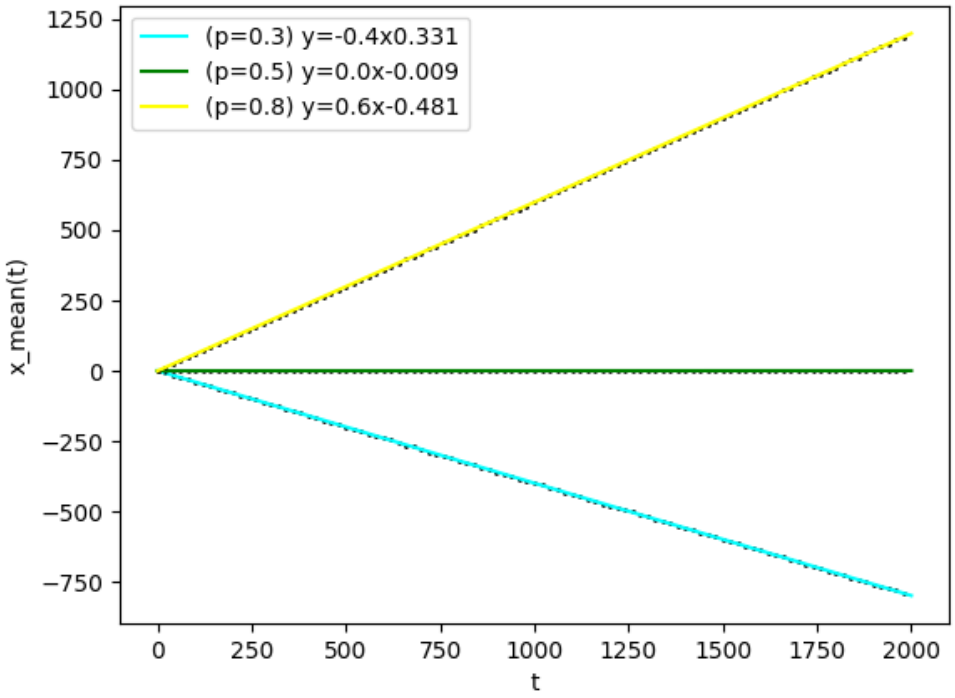
$$= Nl^2(1-(p-q)^2) = \frac{tl^2}{\tau}(p+q-p^2-q^2+2pq) = \frac{tl^2}{\tau}\big(p(1-p+2q)+q(1-q)\big) = \frac{tl^2}{\tau}(3pq+pq)$$

$$\Rightarrow \sigma^2 = \frac{4l^2}{\tau} pqt$$

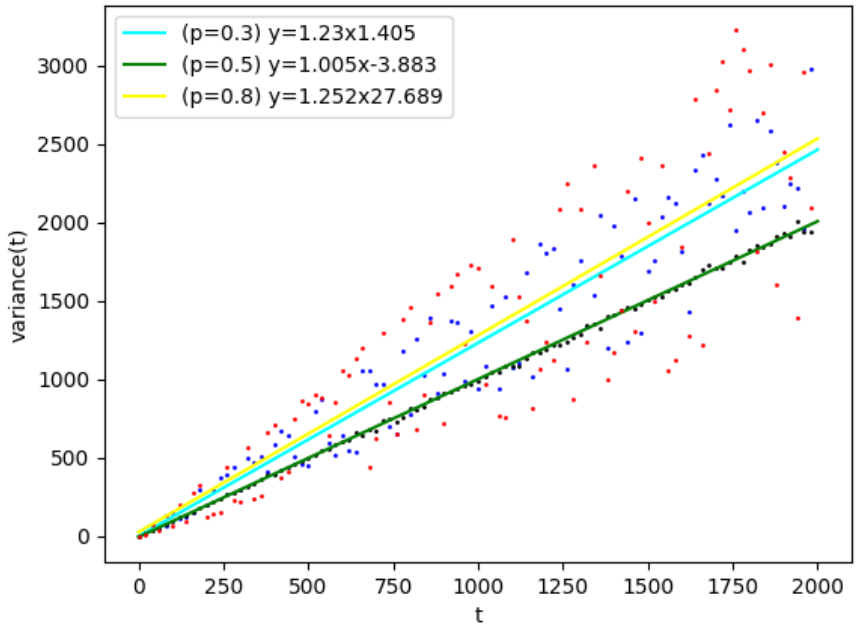
1.2 Main Code

In this algorithm I have divided total time of 2000 into 100 steps. At each step, a loop with 1000 turns starts in each a random walk starts from the origin and goes in until that specific time. The final place of all these turns would be added and the average is calculated which would correspond to x_mean for that time step. The variance for this time step would be calculated too. Lastly, the graph of x_mean over time is plotted and a line is fitted using the least square method to find the slope. The process is repeated 3 times with 3 different values of p.

1.3 Results



The average is calculated over 10000 ensembles. We can see that the results, agree with our expectations that $< x(t) > = \frac{lt}{\tau}(p-q)$. I have assumed that $l = \tau = 1$.



The results agree with theory for p=0.5. The other ones are not very smooth and I think more number of samples are needed before calculating the average.

2. Random Walk with Trap

2.1 Main Code:

-random samples

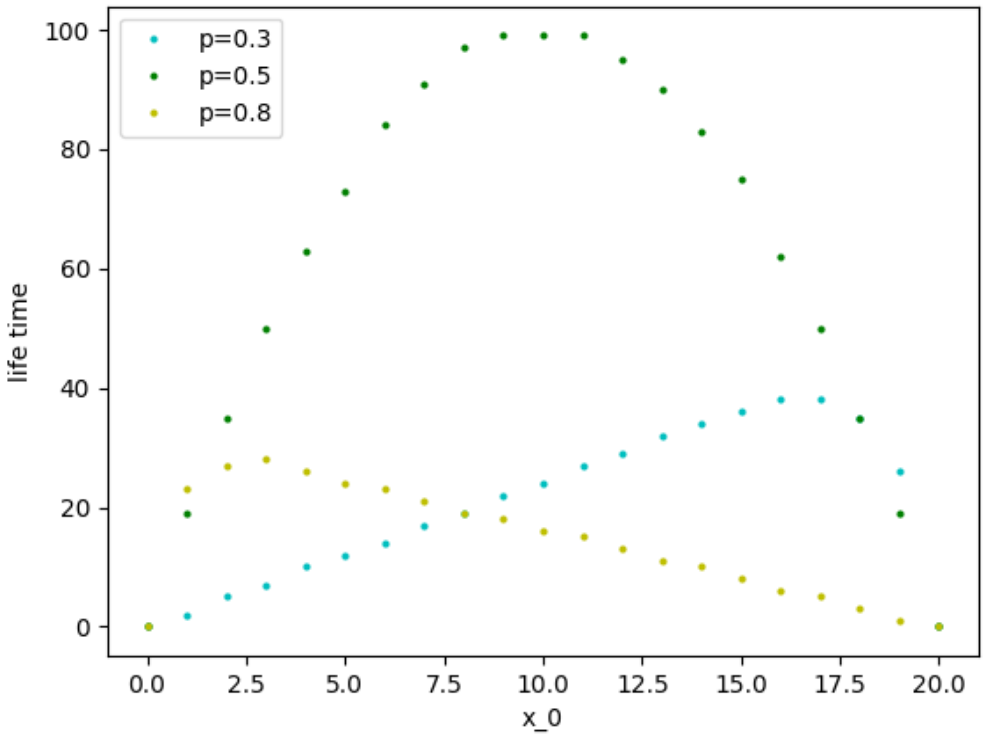
Firstly, a loop starts with 20 turns, each turn dedicated to a specific initial place for the walker. Then a loop with (num) turns starts, in order to calculate the average of a number of samples. In each turn, the walker starts to random walk until reaches any of the boundaries. The number of steps it can take is interpreted as its life time, and the mean life time for every initial position is calculated. Finally the graph of mean life-initial place is drawn for some different values of p.

-enumeration

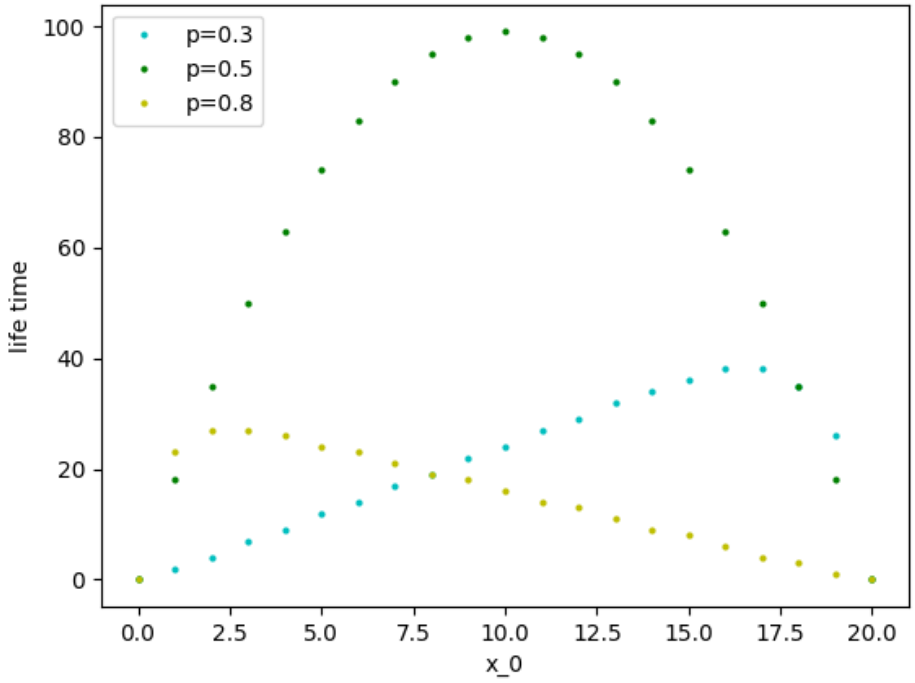
In this algorithm I wanted to produce a table of probability of having each value of x in different times. In the first row of the table, time=0, the walker can only be seen in its initial place so only x_0 is non-zero. The following rows are filled as follows: for each cell, the left and right neighbors in the previous row are checked. the probability of the walker walking towards left is q and walking toward right is p. Therefore, each of the values in the neighbors are squared by their probability and their sum would be the value of our cell. The boundary conditions are attractive, so the walker cannot escape the boundaries. As soon as a boundary cell becomes non-zero, the probability of death of the walker appears. In order to find the life-time of the walker, we need to continue filling the rows until the sum of probabilities in boundary cells approaches 1. At each step, the lifetime would be increased by the amount of time in that step multiply with the probability of dying in that specific step. So we continue to increase the lifetime until the probability of dying is larger than 0.9999. at this point we break the loop and save the lifetime for that specific initial condition. Finally the graph of mean life-initial place is drawn for some different values of p.

2.2 Results

Data achieved by averaging over 10000 random samples:



Data achieved by enumeration method:



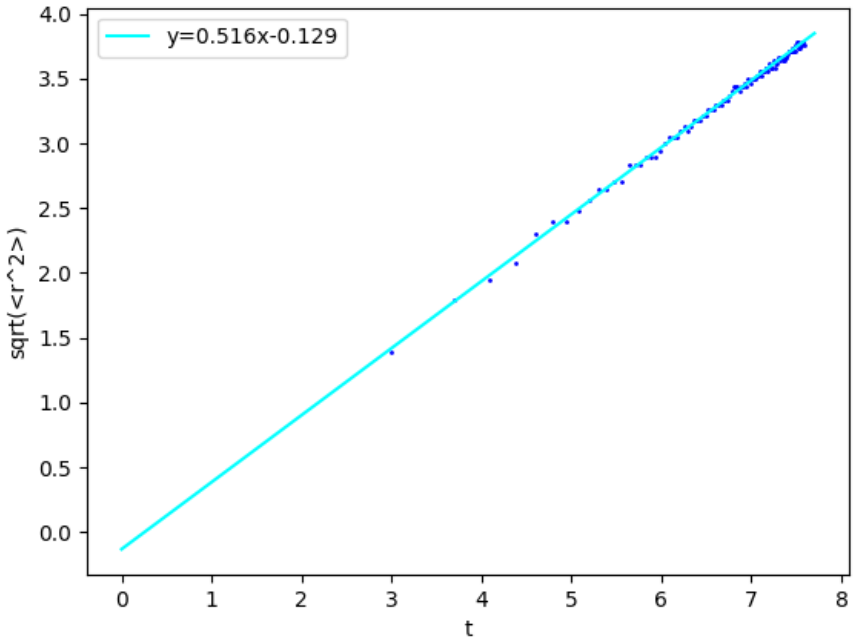
We can see that the results of both methods are very similar. The curves in the enumeration method are smoother.

3. Two Dimensional Random Walk

3.1 Main Code

In this algorithm I have divided total time of 2000 into 100 steps. At each step, a loop with 1000 turns starts in each a random walk starts from the origin and goes in until that specific time. The probability of going toward all directions are equal. The final x and y coordinates of all these turns would be squared and added together and the average is calculated after all rounds. The square root of this value corresponds to gyration radius for that time step. Lastly, the log-log graph of gyration radius over time is plotted and a line is fitted using the least square method to find the slope.

3.2 Results



Note that both axis are logarithmic. As we expected, the slope of the line is around 0.5 which corresponds to $\nu = \frac{1}{2}$.

4. Diffusion limited Aggregation

4.1 Functions Definition

color:

This function takes two arguments, n and number. Then divides the range of number to four parts and depending on which n is in, returns a color value.

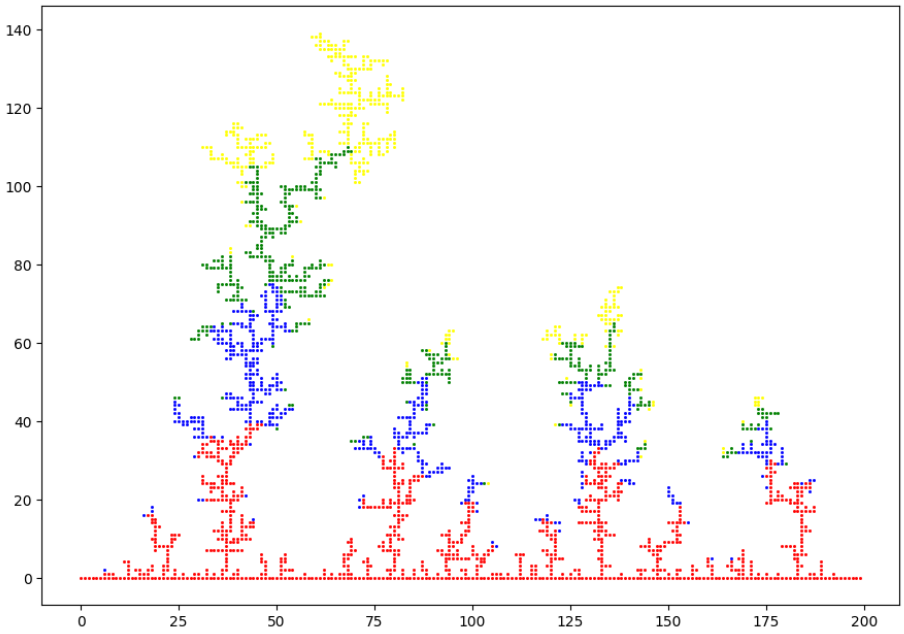
H:

This function takes an array and returns the index of the last element with value 1 in it.

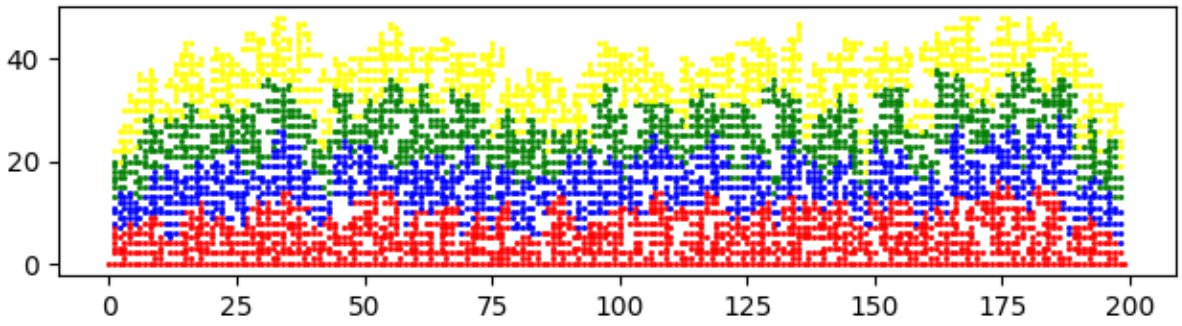
4.2 Main Code

Firstly, an array of 20 elements with value 1 is created, this would be the initial seed. After that a loop starts which let's a random walker walk from a random place above the current cluster in each turn. If the walker visits a place where one of is neighbors is non-zero, it stays in that place. If the walker goes far away, further that 20 units above where it started, it will be ignored and the next walker starts. The code runs for 5000 walkers and then plots the result with different colors to show the dynamics.

4.3 Results



For this result, the initial height of the walkers is chosen above the highest element in the cluster.



For this one, the initial height of the walkers is chosen above the highest element in that specific column using function h.