Probability Project

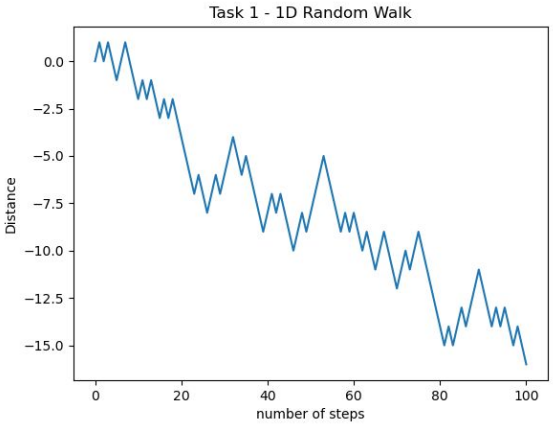
Spring Semester 2020

Group members: Rabeea Atif, Sarrah, Syeda Zainab Fatima

GitHub Respository: https://github.com/rabeeaatif/Probability-and-Statistics-2020

Task 1

Assumed that the object also has an option to not move anywhere and stay in place. We have three different possibilities now. Probability of moving right, left and staying at the same place



Mathematical model

We have a Random Variable P and this can take 3 values. M =[-1, 0, 1] corresponding to left, right and staying in place with corresponding probabilities [0.5, 0 , 0.5]

N = number of times choice is taken by A to move left, right or staying in the same place.So the new position of A after n times is given by;

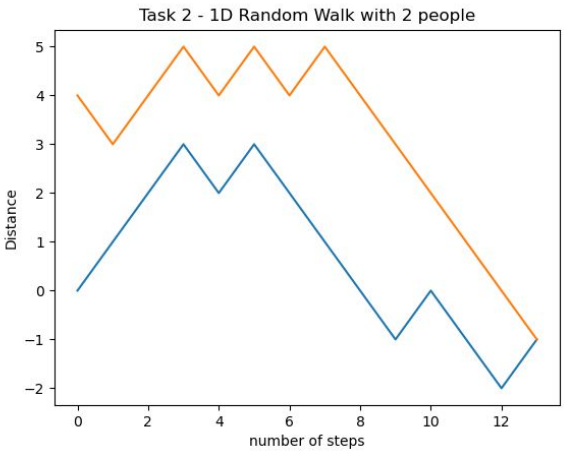
$$\text{NewPosition}\, n = \Sigma_{i=0}^{n} \text{M}\, i$$

Our expectation with the given probabilities would be;

$$E[P] = -1 * P(M = -1) + 0 * P(M = 0) + -1 * P(M = -1)$$

$$E[P] = -1 + 0 + 1 = 0$$

This indicates that our simulation would somewhat hover around 0 with the above given probabilities.

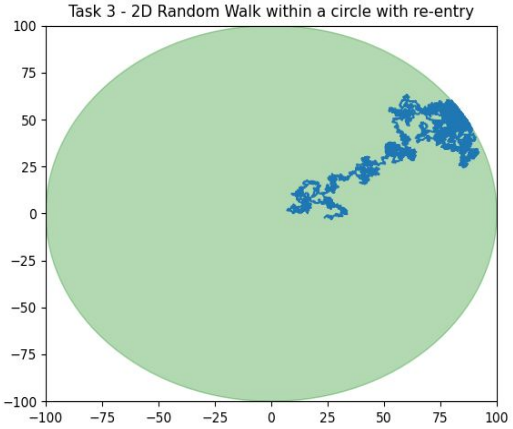


Task 2:

Since we were told to find out the meeting time of the two objects, we ran our code around 4 times and that gave us our average time to be somewhere around 108 iterations but this would vary on the number of simulations we take to calculate the expected value. The more the simulations, the more the accurate the expected value. The code only plots one simulation to show how 2 objects will collide on the graph.

Task 3:

For 2D random walk, we also pick an angle at which the particle moves along with the step.Our range for angles is 0-2π radians.It is to be noted that both angle and step in this task are discrete values. Rounding the angle gives us a discrete value. Then, using the resolution of vectors, we get



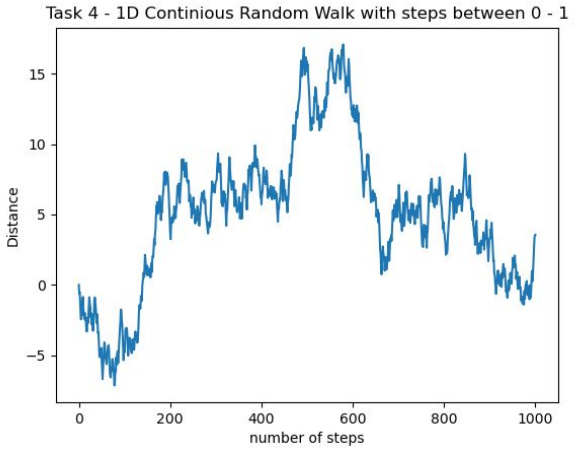
the x and y component which we can plot into our graph/simulation.
Our task is to make sure that the node stays within the 100 unit radius of the circle. When the node does cross this boundary, we restore its position to the last value where it was within the boundary and from there we allow it to move only in a direction that will not cross the 100 unit limit.

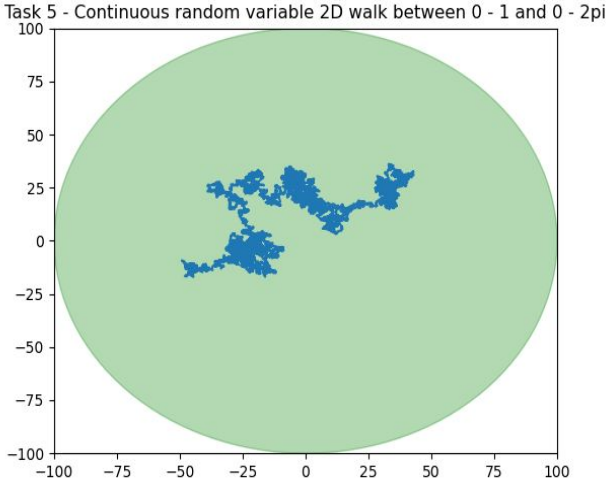
<u>Our model</u> here demonstrates brownian motion within a closed container(highly smooth surface). Our lines are the paths taken by the pollen grains and the reason of change of these paths is due to the gas particles randomly moving. It is not necessary that our gas molecules are moving at the same speed. This directly implies that the speed with which the gas particles collide with the pollens is also different and this is exactly why our pollens take different paths lengths to move. If a higher velocity gas particle hits the pollen it will move a larger distance.

Now, when our pollen hits the boundary ie. a very smooth surface, the energy and the momentum is retained and they reflect back into the closed container with the same velocity. The direction of pollens hitting back into the circle is not definite and can take different paths due to the gas molecules present. We know that according to Brownian motion particles are constantly in a random motion and their pattern of movement remains stochastic. When they hit the boundary they are reflected back and continue their random motion.

<u>Task 4:</u>

In this task , we again deal with 1D random walk. The direction of motion changes according to provided probabilities (left, right and no change) while the step size is a continuous random variable between 0-1. The coordinates of each step taken are then mapped onto our graph, which shows a rather smooth curvature with only minor jaggedness where the direction changes. The step size affects the minuscule changes in the spikes on the graph and each move is determined by the direction that is taken.



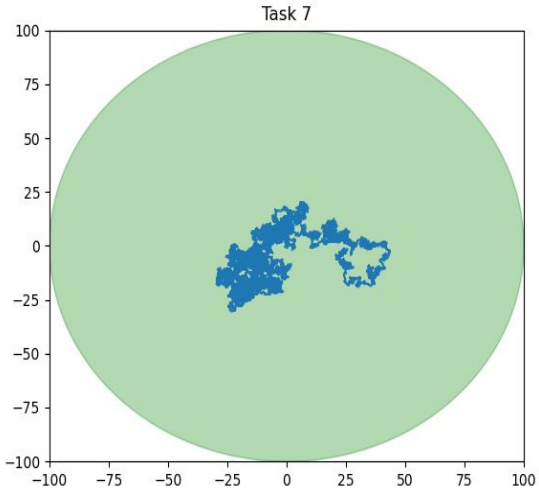


<u>Task 5:</u>

In this task, we are dealing with continuous random variables. So both our step value and orientation are continuous values between (0,1) and (0,2π) respectively. The rest of the procedure is the same as task 3, where our particle moves within the 100 unit radius and bounces back when it hits the boundary.

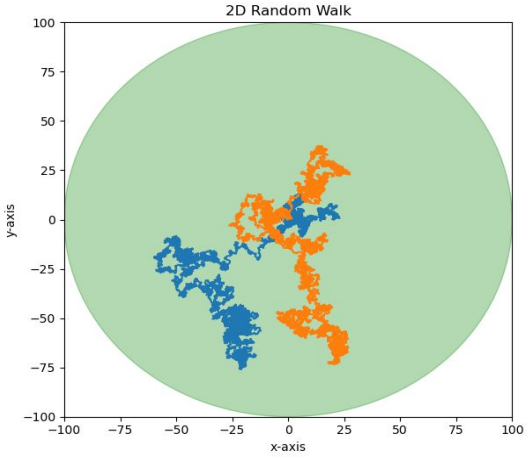
Task 7:

This is again a variation of task 3 where our step size is a discrete random variable but our angle has continuous values. The 2D random walk takes place in the same fashion, only with different types of random variables. The simulation of this task has less clustered steps in one place than the previous tasks (3 and 5) where due to the continuous nature of step size, we encountered steps that were in close proximity to each other over a wide range of steps, resulting in a clustered graph.



Task 8:

We have a pair of nodes, moving randomly in a circle. We allow them 1000,000 steps in which if they collide with each other, the simulation ends. When they do not collide within the given number of steps, our simulation returns us step count as 10,000. One such example is this:



The simulation however ends when 10,000 steps are completed. It is to be noted that this does not eliminate the possibility of collision of the nodes, as they have to collide at one point or another. They are moving in a limited space, bound by a circular region and they bounce back when they collide with the boundaries. It is therefore impossible for the nodes to never bump into each other, while it may take longer time than expected.
According to our model, from 10 simulations, our expected average number of steps until collision is 257545.