Lab-X

Devansh Shukla
I18PH021

1 Aim

Write and execute an octave program to simulate/solve motion of a particle in random walk.

2 Theory

3 Program

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% RandomWalk
%
% Program to solve/simulate the random walk motion of a particle
% Author: Devansh Shukla I18PH021
% 6th April, 2022
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

graphics_toolkit gnuplot

% load the symbolic package
pkg load symbolic

% define symbolic variables
syms N r_rms
% random walk
R = sqrt(N) * r_rms

% computing
step = 1;
total_steps = input("Enter no of steps ");

% initial position (0, 0)
x = y = [0];

% computing
for i=1:total_steps
    a = rand();
    b = rand();
    x_value = step * cos(2 * pi * a);
    y_value = step * sin(2 * pi * b);
    x(i + 1) = x(i) + x_value;
    y(i + 1) = y(i) + y_value;
endfor

% plotting the trajectory
fig = figure();
plot(x, y, "linewidth", 2);
title("Random walk")
xlabel("X");
ylabel("Y");
set(gcf, 'renderer', 'painters');
print("-dpng", "random_walk.png");
```

4 Results

4.1 Terminal output

```
(escape) devansh@ds:~/GitHub/Vault/OctaveLab/Programs/outputs$ octave ../RandomWalk.m
Symbolic pkg v2.9.0: Python communication link active, SymPy v1.5.1.
R = (sym) √N·r_rms
Enter no of steps 1000
```

4.2 Plots

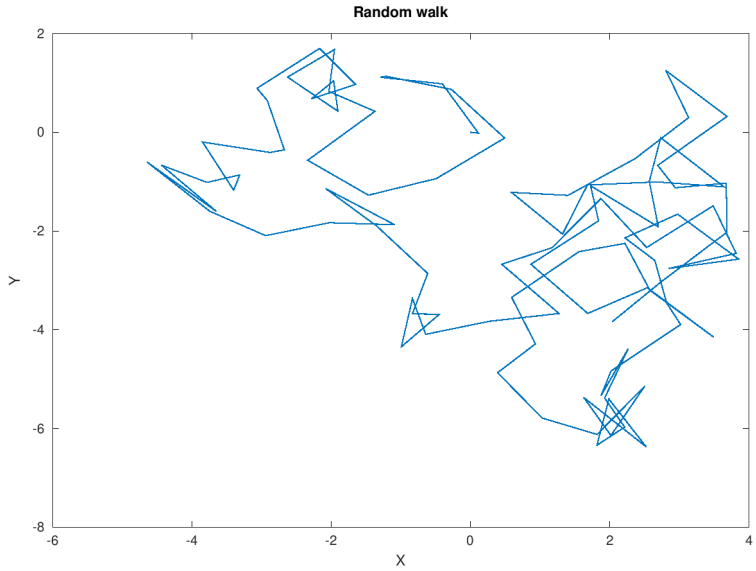

Figure 1: Random walk with $n = 100$

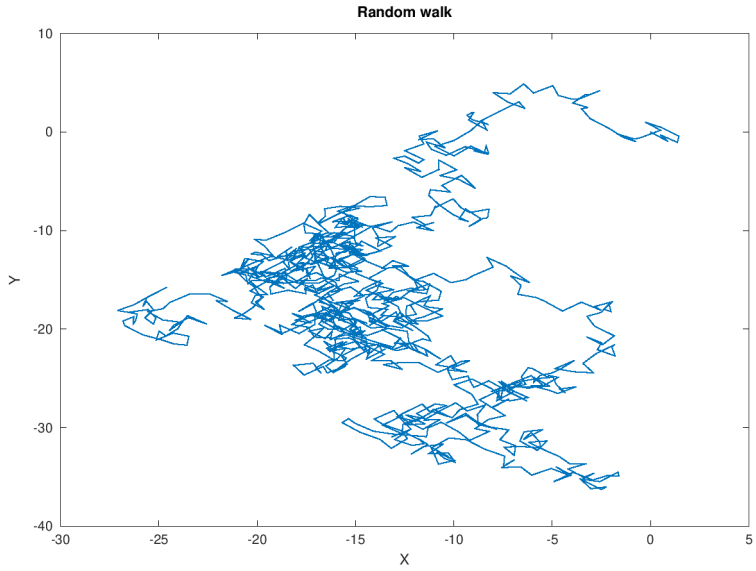

Figure 2: Random walk with $n = 1000$

5 Remarks

The programs can be used to trace and simulate the motion of any particle in random walk by defining the required parameters.