PPS

Programming for Problem Solving

Mini Project

Nishtha Sharma

RA2111019010059

Problem Statement

Write a program in C to simulate the stochastic process of Random Walks in 1D

Analysis:

* Accept the user input
* Generate random numbers for the random walk
* Simulate the process and observe it in a file
* Display the final distance in the distance

FDT-Function Description Table

|  |  |  |  |
| --- | --- | --- | --- |
| Function Name | Return Type | Purpose | Parameter List |
| main | void | To take user’s inputs | - |
| rand | int | To generate random numbers | int r0, int a, int m, int c |
| randomNos | void | To generate random numbers given a seed, and stores them in an array that is passed as an argument. | int r0, int a, int m, int c, int n, int x[n] |
| coinTossSingle | int | To return the result of a coin toss | double r |
| coinToss | void | To generate n coin tosses results, given a seed and other starting conditions, and stores them in an array that is passed as an argument. | int r0, int a, int m, int c, int n, int results[n] |

Algorithm

**Random Walk:**

A random walk is a mathematical object, known as a stochastic or random process that describes a path that consists of a succession of random steps on some mathematical space such as the integers. An elementary example of a random walk is the random walk on the integer number line, which starts at 0 and at each step moves +1 or -1 with equal probability. Other examples include the path traced by a molecule as it travels in a liquid or a gas, the search path of a foraging animal, the price of a fluctuating stock and the financial status of a gambler can all be approximated by random walk models, even though they may not be truly random in reality. As illustrated by those examples, random walks have applications to many scientific fields including ecology, psychology, computer science, physics, chemistry, biology as well as economics. Random walks explain the observed behaviors of many processes in these fields and thus serve as a fundamental model for the recorded stochastic activity. As a more mathematical application, the value of pi can be approximated by the usage of random walk in the agent-based modelling environment.

**One-dimensional random walk:**

An elementary example of a random walk is the random walk on the integer number line, which starts at 0 and at each step moves +1 or 1 with equal probability.

Source Code

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Generating a Random Walk

Code to simulate the stochastic process in 1-D

Nishtha Sharma RA2111019010059

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<stdio.h>

#include<math.h>

int rand (int r0, int a, int m, int c)

{

int r1= (a\*r0+c) %m;

return r1;

}

void randomNos (int r0, int a, int m, int c, int n, int x[n])

{

int r1=rand (r0, a, m, c);

int i;

for (i=0; i<n; i++)

{

x[i] =r1;

r1=rand (r1, a, m, c);

}

}

int coinTossSingle (double r)

{

if (r>0.5)

return 1;

else if(r<0.5)

return 0;

}

void coinToss (int r0, int a, int m, int c, int n, int results[n])

{

int randNos[n];

randomNos (r0, a, m, c, n, randNos);

//Renormalize to 0 to 1

int i;

double randNosNew[n];

for (i=0; i<n; i++)

randNosNew[i] = (double) randNos[i]/ (m-1);

for (i=0; i<n; i++)

results[i] =coinTossSingle (randNosNew[i]);

}

void main ()

{

int a, m, c, r0, n;

printf ("Enter the value of a:\n");

scanf ("%d", &a);

printf ("Enter the value of m:\n");

scanf ("%d", &m);

printf ("Enter the value of c:\n");

scanf ("%d", &c);

printf ("Enter the value of r0 (initial):\n");

scanf ("%d", &r0);

printf ("Enter the no. of steps require: \n");

scanf ("%d", &n);

int tossResults[n];

coinToss (r0, a, m, c, n, tossResults);

int i;

//Step-size

double h=1;

double x0=0, origin=x0;

double x1;

//Array to store the position of the random walker at the ith step

double x[n];

for (i=0; i<n; i++)

{

if (tossResults[i] ==1)

{

//Heads=>Move right

x1=x0+h;

} else

{

//Tails=>Move left

x1=x0-h;

}

//Store the position at the ith step in array x[i]

x[i] =x1;

x0=x1;

}

//Plot the random Walk (Trajectory)

FILE \*fp=NULL;

fp=fopen ("randomWalk1.txt","w");

for (i=0; i<n; i++) {

fprintf (fp,"%d\t%lf\n", i+1, x[i]);

}

double dist=x1-origin;

printf ("\nThe distance travelled is: \n%lf", dist);

}

VDT-Variable Description Table

|  |  |  |  |
| --- | --- | --- | --- |
| Variable Name | Data Type | Purpose | Scope |
| r0 | int | Initial seed | int rand(), void randomNos(),void coinToss(), void main() |
| a | int | Scale factor | int rand(), void randomNos(),void coinToss(), void main() |
| m | int | Gives the max. value of random numbers that can be generated | int rand(), void randomNos(),void coinToss(), void main() |
| c | int | Additional displacement factor | int rand(), void randomNos(),void coinToss(), void main() |
| r1 | int | To store the random number | int rand(), void randomNos() |
| n | int | To store the number of random numbers to be generated | void randomNos(),void coinToss(), void main() |
| x[n] | int | Array that will store the random numbers | void randomNos() |
| randNosNew[n] | double | Array that will store the random numbers | void coinToss() |
| r | double | To store a random number between 0 and 1 | int coinTossSingle (double r) |
| i | int | LCV-Loop Control Variable | void randomNos(),void coinToss(), void main() |
| tossResults[n] | int | To store the results of all coin tosses | void main() |
| h | double | To store the step size | void main() |
| x0 | double | To calculate the next step | void main() |
| x1 | double | To calculate the next step | void main() |
| origin | double | Marks the start of the random walk | void main() |
| \*fp | FILE | To record all steps | void main() |
| dist | int | To store the dist travelled | void main() |

Sample Input Output







