***One-D Random walk***

﻿using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace OneDRandomWalk

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

//1D Random Walk along X Axis

float x = 0, dx = 1;

int w=400,s=50;

float [] x2ave = new float[s];

//to save values of x^2 at each step

Graphics gg = CreateGraphics();

SolidBrush bred = new SolidBrush(Color.Red);

SolidBrush bblue = new SolidBrush(Color.Blue);

Font f = new Font("arial", 14);

Pen pblue=new Pen(Color.Blue);

Random[] walkers = new Random[w];//array of walkers

gg.DrawLine(pblue, 400, 600, 600, 600);

gg.DrawLine(pblue, 400, 600, 400, 100);

for (int i = 0; i < w; i++)

{

walkers[i] = new Random(i);//seed will be different

}

for (int i = 0; i < w; i++)//walker loop

{

x = 0;//every walker will start from x=0

for (int j = 0; j < s; j++)//step loop

{

if (walkers[i].NextDouble() <= 0.5)

{

x = x + dx;//rightward

// gg.FillEllipse(bred, 400 + j \* 5, 500 + x \* 10, 8, 8);

}

else

{

x = x - dx;//leftward

// gg.FillEllipse(bred, 400 + j \* 5, 500 + x \* 10, 8, 8);

}

x2ave[j] = x2ave[j] + x \* x;

// System.Threading.Thread.Sleep(20);

}//end of step loop

}//end of walker loop

System.Threading.Thread.Sleep(1000);

//this.Refresh();

gg.DrawLine(pblue, 400, 600, 600, 600);

gg.DrawLine(pblue, 400, 600, 400, 100);

gg.DrawString("step number/time", f, bred, 500, 620);

gg.DrawString("<x^2>/x\_rms", f, bred, 350, 400);

for (int j = 0; j < s; j++)//step loop

{

//mean squared displacement vs step number

gg.FillEllipse(bblue, 400 + j \* 10,

600 - x2ave[j] / w\*20/\*<x^2>\*/, 4, 4);

gg.FillEllipse(bred, 400 + j \* 10,

600 - (float)Math.Sqrt(x2ave[j] / w)\*20/\*<x^2>\*/, 4, 4);

}

}

private void button3\_Click(object sender, EventArgs e)

{

//event handler

int n\_molecules = 400;//no of cream molecules to be studied

Random[] molecules = new Random[n\_molecules];//array of molecules

for (int i = 0; i < n\_molecules; i++)//Make every molecule a random walker

{

molecules[i] = new Random(i);//i is seed which will be different

}

float xc = textBox1.ClientSize.Width / 2, yc = textBox1.ClientSize.Height/2;

//Divide the container into the large number of grid cells

int n\_cells = 19;

int[,] grid = new int[n\_cells, n\_cells];

float dx = 2 \* xc / n\_cells,dy=2\*yc/n\_cells;

//dx is width of single grid cell, dy is height of cell

//(xc,yc) is center point of the 2D container which is textBox1

//Graphics

Graphics gg = textBox1.CreateGraphics();

Graphics gg1 = this.CreateGraphics();

SolidBrush bred = new SolidBrush(Color.Red);

SolidBrush bblue = new SolidBrush(Color.Blue);

SolidBrush bwhite = new SolidBrush(Color.White);

SolidBrush bGray = new SolidBrush(Color.Gray);

Font f = new Font("arial", 14);

Pen pblue = new Pen(Color.Blue);

for (int i = 0; i < n\_cells; i++)

{

gg.DrawLine(pblue, dx \* i, 0, dx \* i, 2 \* yc);

}

for (int i = 0; i < n\_cells; i++)

{

gg.DrawLine(pblue, 0, i\*dy, 2\*xc, i\*dy);

}

//position arrays

Point [ ] position=new Point[n\_molecules];//100 points

//put cream molecules in the coffee container

double x, y,xmin=-dx/2,xmax=dx/2,ymin=-dy/2,ymax=dy/2,r;

for (int i = 0; i < n\_molecules; i++)//Make every molecule a random walker

{

x = xmin+(xmax-xmin)\*molecules[i].NextDouble();

y = ymin + (ymax - ymin) \* molecules[i].NextDouble();

position[i].X=(int)x;position[i].Y=(int)y;

//min<(x,y)<max

//x,y are position coordinates of molecules

gg.FillEllipse(bwhite, xc + (float)x, yc - (float)y, 4, 4);

// System.Threading.Thread.Sleep(10);

}//cream has been placed inside the coffee cup (2D Container)

//start diffusion

int row, col;

gg1.DrawLine(pblue, 800, 700, 1200, 700);

gg1.DrawLine(pblue, 800, 700, 800, 100);

float entropy=0,prob,time=0,delt=0.1f;

while(true)//until we want to see the diffusion (time loop)

{

for (int i = 0; i < n\_molecules; i++)//Make every molecule a random walker

{

gg.FillEllipse(bGray, xc + (float)position[i].X, yc- (float) position[i].Y, 4, 4);

r = molecules[i].NextDouble();//0<=r<1

if (r <= 0.25)

position[i].X = position[i].X+5;//move right

if(r>0.25&&r<=0.5)

//position[i] means position of ith molecule

//position[i].X=x-coordinate of position of ith molecule

//position[i].Y=y-coordinate of position of ith molecule

position[i].X = position[i].X-5;//move left

if (r > 0.5 && r <= 0.75)

position[i].Y = position[i].Y - 5;//move up

if (r > 0.75)

position[i].Y = position[i].Y + 5;//move down

//changed position

if(position[i].X<=-xc)

position[i].X = position[i].X + 5;//send right

if (position[i].X >= xc)

position[i].X = position[i].X - 5;//send left

if (position[i].Y <= -yc)

position[i].Y = position[i].Y + 5;//send down

if (position[i].Y >= yc)

position[i].Y = position[i].Y - 5;//send UP

//position[i] is position of ith molecule

//position[i].X is x-ccordinate of position of ith molecule

//find and count the cream molecules in grid cells

row = (int)((position[i].Y + yc) / dy);

col = (int)((position[i].X + xc) / dx);

//(row,col) will give that grid cell which contains ith molecule

grid[row, col]++;//count the molecules in the grid cell

gg.FillEllipse(bwhite, xc + (float)position[i].X,

yc - (float)position[i].Y, 4, 4);

}//end of for loop (we have the molecules counted in the grid cells)

//compute entropy

for (int m = 0; m < n\_cells; m++)

{

for (int n = 0; n < n\_cells; n++)

{

prob = (float)(grid[m, n]) / (float)n\_molecules;

if(grid[m,n]!=0)

entropy = entropy + prob \* (float)Math.Log(prob);

}

}

entropy = -entropy;

//plot entropy versus time

gg1.FillEllipse(bblue, 800 + time \* 10, 700 - entropy \* 100, 4, 4);

time = time + delt;

for (int m = 0; m < n\_cells; m++)

{

for (int n = 0; n < n\_cells; n++)

{

grid[m, n] = 0;

}

}

entropy = 0;

}//end of while loop

}

private void button2\_Click(object sender, EventArgs e)

{

//2D Random Walk in xy-plane

float x = 0, y = 0, r2 = 0, dx = 1, dy = 1;

int w = 400, s = 100;

float[] r2ave = new float[s];

//to save values of x^2 at each step

Graphics gg = CreateGraphics();

SolidBrush bred = new SolidBrush(Color.Red);

SolidBrush bblue = new SolidBrush(Color.Blue);

Font f = new Font("arial", 14);

Pen pblue = new Pen(Color.Blue);

Random[] walkers = new Random[w];//array of walkers

gg.DrawLine(pblue, 400, 600, 600, 600);

gg.DrawLine(pblue, 400, 600, 400, 100);

for (int i = 0; i < w; i++)

{

walkers[i] = new Random(i);//seed will be different

}

for (int i = 0; i < w; i++)//walker loop

{

x = 0; y = 0;//every walker will start from x=0

for (int j = 0; j < s; j++)//step loop

{

if (walkers[i].NextDouble() <= 0.25)//0---0.25----0.5----0.75---1.0

{

x = x + dx;//rightward

}

if (walkers[i].NextDouble() > 0.25 && walkers[i].NextDouble() <= 0.5)

{

x = x - dx;//leftward

}

if (walkers[i].NextDouble() > 0.5 && walkers[i].NextDouble() <= 0.75)

{

y = y + dy;//downward in c#

}

if (walkers[i].NextDouble() > 0.75 && walkers[i].NextDouble() < 1.0)//0---0.25----0.5----0.75---1.0

{

y = y - dy;//upward in c#

}

//gg.FillEllipse(bred, 600 + x \* 10, 600-y \* 10, 4, 4);

r2 = x \* x + y \* y;

r2ave[j] = r2ave[j] + r2;

//System.Threading.Thread.Sleep(1);

}//end of step loop

this.Refresh();

//System.Threading.Thread.Sleep(50);

}//end of walker loop

//System.Threading.Thread.Sleep(1000);

gg.DrawLine(pblue, 400, 600, 600, 600);

gg.DrawLine(pblue, 400, 600, 400, 100);

gg.DrawString("step number/time", f, bred, 500, 620);

gg.DrawString("<r^2>/r\_rms", f, bred, 350, 400);

for (int j = 0; j < s; j++)//step loop

{

//mean squared displacement vs step number

gg.FillEllipse(bblue, 400 + j \* 10,

600 - (r2ave[j] / w) \* 20/\*<r^2>\*/, 4, 4);

gg.FillEllipse(bred, 400 + j \* 10,

600 - (float)Math.Sqrt(r2ave[j] / w) \* 20/\*<r^2>\*/, 4, 4);

}

}

}

}