import numpy as np

import pylab

import random

import math

import matplotlib.pyplot as plt

# defining the number of steps

n = 1000 #取多少個時間點

z = 200 #幾個例子

a = 0 #用來計算總碰撞次數的變數

w = 0 #用來碰撞的方便用的代數

e = 0 #用來碰撞的方便用的代數

r = 0 #用來碰撞的方便用的代數

t = 0 #用來碰撞的方便用的代數

q = 1000 #再多大的空間裡碰撞(放大inlarge倍)

RS = 0.5 #例子半徑

v = 5 #最大速度(放大inlarge倍)

qqq = 0

inlarge = 1000

sigma = 2.3

rando = 1 #給定速度方向(1:第一象限)

#creating two array for containing x and y coordinate

#of size equals to the number of size and filled up with 0's

# filling the coordinates with random variables

vx = np.zeros((n+1,z),dtype=np.float)#建立每顆粒子隨時間的X方向的速度矩陣

vy = np.zeros((n+1,z),dtype=np.float)#建立每顆粒子的Y方向速度的速度矩陣

cx = np.zeros((z),dtype=np.float) #每個時間點的x速度

cy = np.zeros((z),dtype=np.float)#每個時間點的y速度

x = np.zeros((n+1,z),dtype=np.float)#建立每顆粒子的X方向的位置矩陣

y = np.zeros((n+1,z),dtype=np.float)#建立每顆粒子的Y方向的位置矩陣

for p in range(0, z):

rando = random.randint(1,4)

if rando == 1 :

vx[0,p] = np.random.normal(v, sigma, size=(1,1))/inlarge#建立每顆粒子X方向初位置

vy[0,p] = np.random.normal(v, sigma, size=(1,1))/inlarge#建立每顆粒子y方向初位置

elif rando == 2 :

vx[0,p] = -1\*np.random.normal(v, sigma, size=(1,1))/inlarge#建立每顆粒子X方向初位置

vy[0,p] = np.random.normal(v, sigma, size=(1,1))/inlarge#建立每顆粒子y方向初位置

elif rando == 3 :

vx[0,p] = -1\*np.random.normal(v, sigma, size=(1,1))/inlarge#建立每顆粒子X方向初位置

vy[0,p] = -1\*np.random.normal(v, sigma, size=(1,1))/inlarge#建立每顆粒子y方向初位置

else:

vx[0,p] = np.random.normal(v, sigma, size=(1,1))/inlarge#建立每顆粒子X方向初位置

vy[0,p] = -1\*np.random.normal(v, sigma, size=(1,1))/inlarge#建立每顆粒子y方向初位置

x[0,p] = random.randint(-q,q)/inlarge#建立每顆粒子X方向初速度

y[0,p] = random.randint(-q,q)/inlarge#建立每顆粒子y方向初速度

for t in range(0,z-1):

cx[t]=vx[0,t]

cy[t]=vy[0,t]

for i in range(0, n):

for j in range(0,z):

if x[i,j]>=q/inlarge or x[i,j]<=-q/inlarge:#如果碰到牆壁例子反彈

cx[j]=-1\*cx[j]

if y[i,j]>=q/inlarge or y[i,j]<=-q/inlarge:

cy[j]=-1\*cy[j]

else:

cy[j]=cy[j]

elif y[i,j]>=q/inlarge or y[i,j]<=-q/inlarge:

cy[j]=-1\*cy[j]

if x[i+1,j]>=q/inlarge or x[i,j]<=-q/inlarge:

cx[j]=-1\*cx[j]

else:

cx[j]=cx[j]

else: #如果發生碰撞

for k in range(j+1,z):

if (x[i,j]-x[i,k])\*\*2+(y[i,j]-y[i,k])\*\*2<=0.25\*(RS\*\*2):

w = cx[j]

e = cy[j]

r = cx[k]

t = cy[k]

cx[j] = r

cy[j] = t

cx[k] = w

cy[k] = e

elif(x[i,j]-x[i,k])\*\*2+(y[i,j]-y[i,k])\*\*2<=RS\*\*2 and (x[i,j]-x[i,k])\*\*2+(y[i,j]-y[i,k])\*\*2>0.25\*(RS\*\*2):

qqq = (x[i,j]-x[i,k])\*\*2+(y[i,j]-y[i,k])\*\*2

qqq = qqq\*\*0.5

w = cx[j]+(-cx[j]+cx[k])\*abs((x[i,j]-x[i,k]))/qqq

e = cy[j]+(-cy[j]+cy[k])\*abs((y[i,j]-y[i,k]))/qqq

r = cx[k]+(cx[j]-cx[k])\*abs((x[i,j]-x[i,k]))/qqq

t = cy[k]+(cy[j]-cy[k])\*abs((y[i,j]-y[i,k]))/qqq

cx[j] = w

cy[j] = e

cx[k] = r

cy[k] = t

a = a+1

for m in range (0,z):#改變位置

vx[i+1,m]=cx[m]

vy[i+1,m]=cy[m]

x[i+1,m]=x[i,m]+vx[i+1,m]

y[i+1,m]=y[i,m]+vy[i+1,m]

print(a)

picturex = np.zeros(n-1)

picturey =np.zeros(n-1)

# plotting stuff:

for i in range(0, n-1):

picturex[i] = x[i,0]

picturey[i] = y[i,0]

pylab.title("Random Walk ($n = " + str(n) + "$ steps)")

pylab.plot(picturex,picturey,'purple')

pylab.savefig("rand\_walk"+str(n)+".png",bbox\_inches="tight",dpi=600)

pylab.show()

#Initial Position