# -\*- coding: utf-8 -\*-

import random

import matplotlib.pyplot as plt

import math

position = [0, 0]

oldx = []

oldy = []

def move():

'''

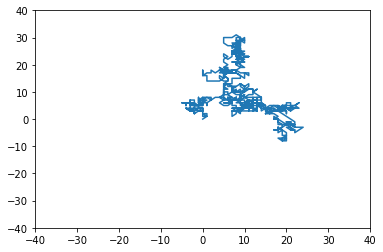
0 - Move positive

1 - Move negative

2 - Don't move

'''

directionX = random.randint(0,3)

 directionY = random.randint(0,3)

if directionX == 0:

position[0] += 1

elif directionX == 1:

position[0] += -1

elif directionX == 2:

position[0] += 0

if directionY == 0:

position[1] += 1

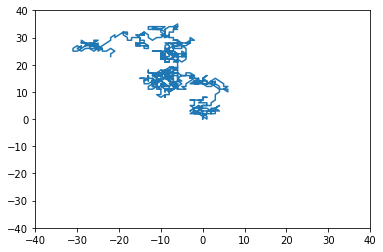
elif directionY == 1:

position[1] += -1

elif directionY == 2:

position[1] += 0

oldx.append(position[0])

 oldy.append(position[1])

def randomWalk(steps):

for x in range(0, steps):

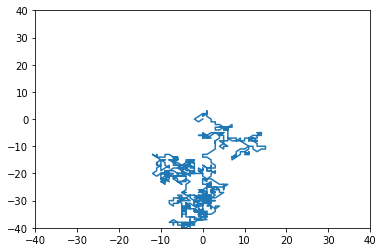
move()

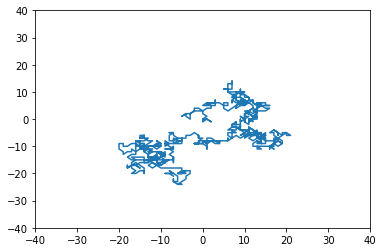
x += 1

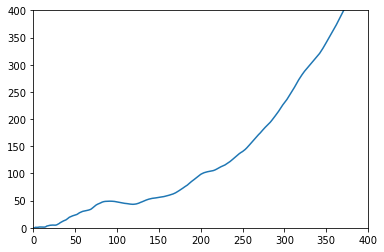
plt.plot(oldx, oldy, '-')

plt.xlim(-40, 40)

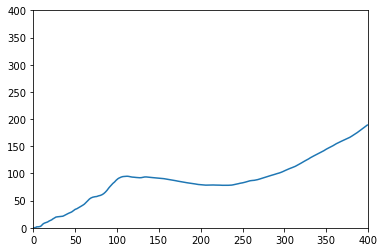
plt.ylim(-40, 40)

randomWalk(1000)

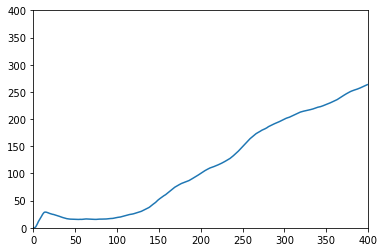




Mean Squared value for 10 trials using 400 steps



Mean squared for 50 trials using 400 steps



Mean squared for 100 trials using 400 steps

Code:

# -\*- coding: utf-8 -\*-

import random

import matplotlib.pyplot as plt

import math

position = [0, 0]

oldx = []

oldy = []

def move():

'''

0 - Move positive

1 - Move negative

2 - Don't move

'''

directionX = random.randint(0,3)

directionY = random.randint(0,3)

if directionX == 0:

position[0] += 1

elif directionX == 1:

position[0] += -1

elif directionX == 2:

position[0] += 0

if directionY == 0:

position[1] += 1

elif directionY == 1:

position[1] += -1

elif directionY == 2:

position[1] += 0

oldx.append(position[0])

oldy.append(position[1])

def randomWalk(steps):

for x in range(0, steps):

move()

x += 1

def distance(positionOne, positionTwo):

xDif = abs(positionOne[0] - positionTwo[0])

yDif = abs(positionOne[1] - positionTwo[1])

totDif = math.sqrt(xDif\*\*2 + yDif\*\*2)

return totDif

def meanSquaredDisplacement(finalt):

t = 1

squareVals = []

while (t < finalt):

positionOne = [oldx[t], oldy[t]]

positionTwo = [oldx[0], oldy[0]]

squareVals.append(distance(positionOne, positionTwo)\*\*2)

t += 1

msd = sum(squareVals)/finalt

return msd

meanSquareValues = []

iteration = 0

for x in range(0, 300):

steps = 400

randomWalk(steps)

j = 1

if iteration == 0:

while (j <= steps):

currentVal = meanSquaredDisplacement(j)

meanSquareValues.append(currentVal)

j += 1

else:

while (j < steps):

currentVal = meanSquaredDisplacement(j)

meanSquareValues[j] = (meanSquareValues[j] + currentVal)/2

j += 1

iteration +=1

plt.plot([i for i in range(400)],meanSquareValues, '-')

plt.xlim(0, 400)

plt.ylim(0, 400)