**Estudantes:**

**- Eduardo Eiji Goto**

**- Gustavo Hammerschmidt**

**- João Vitor Andrioli**

**TDE 01 - Simulação do Passeio Aleatório**

1. Problema

Uma partícula se desloca passo a passo de modo aleatório em uma trajetória reta. A partícula pode realizar um deslocamento (um passo) de comprimento 1, para uma direção (direita ou esquerda) com igual probabilidade. A probabilidade de ela se deslocar para a esquerda é a mesma de ela se deslocar para direta, e igual a 0,5 (a partícula não pode permanecer parada de uma etapa para outra). A partícula inicia na origem (posição zero), e que um passo à direita é considerado um deslocamento positivo e um passo à esquerda é considerado um deslocamento negativo. Queremos encontrar é qual a probabilidade de que partícula esteja na posição k após n passos? (n ≥ 1)

1. Solução teórica

Para se calcular a probabilidade que um passeio aleatório de *n* passos termine na posição *k* usamos a seguinte equação:

****

Detalhes no Anexo 1.

1. Assistir o vídeo:

<https://www.youtube.com/watch?v=vz1wWCFpzl0&feature=youtu.be&hd=1>

1. Implementação da simulação vetorial

O arquivo “TDE01.ipynb” contém o código Python no Jupyter Notebook tem a implementação do cálculo teórico e da simulação interativa do passeio aleatório.

Contém também as orientações para a implementação vetorial. Implementar a simulação vetorial e comparar os tempos de execução do algoritmo de simulação interativo e do algoritmo de simulação vetorial.

Copie o código da sua implementação aqui:

import numpy as np

from math import factorial

def combinacao(n, x):

    return factorial(n)/(factorial(x)\*(factorial(n - x)))

def mesma\_paridade(n, k):

    return (n % 2 == 0 and k % 2 == 0) or (n % 2 != 0 and k % 2 != 0)

def PasseioV(npassos, pos, nsim):

    # simulacao vetorial

    # Passo 1

    # sortear uma matriz (array) com nsim linhas e npassos colunas

    # cada linha da matriz é um passeio com npassos

    # cada passo tem valor 1 (passo à direita) ou -1 (passo à esquerda) com a mesma probabilidade

    # dica: utilizar np.random.choice

    matriz = np.array([ np.random.randint(0, 2, npassos) for \_ in range(nsim)])

    # Passo 2

    # somar os valores das linhas da matriz calculada anteriormente

    # o resultado é um vetor com as posicoes finais de cada passeio

    row\_sum = [ s-(npassos-s) for s in [ sum(\_) for \_ in matriz] ]

    # Passo 3

    # calcular a probabilidade simulada

    # pSim deve ser igual a quantidade de passeios que terminaram na posicao pos divida pelo numero de simulacoes

    # o comando abaixo faz pSim = 0 para o programa não dar erro se for executado

    # você deve substituir o comando pelo calculo apropriado

    # Dica: se o o nome do vetor calculado no Passo 2 for "posicoes", o comando "posicoes == pos" produz um novo vetor

    # Os valores do novo vetor são iguais a True se o passeio terminou na posicao pos e False caso contrario

    # Usar np.sum para contar a quantidade de True

    pSim = sum([1 for \_ in row\_sum if \_==pos]) / nsim

    # calculo teorico

    if mesma\_paridade(npassos, pos):

        pTeorica = combinacao(npassos, (npassos + pos) / 2) \* (2 \*\* (-npassos))

    else:

        pTeorica = 0

    return pSim, pTeorica

passos = 100 #int(input("Defina o numero de passos: "))

pos = 2      #int(input("Defina a posicao final da trajetoria: "))

nsim = 1000  #int(input("Def1ina o numero de simulacoes: "))

tstart = time.perf\_counter()

pSim, pTeorica = PasseioV(passos, pos, nsim)

tend = time.perf\_counter()

print('Probabilidade simulada:  {:.4f}'.format(pSim))

print('Probabilidade teorica: {:.4f}'.format(pTeorica))

print("Tempo de execução: {:.4f}".format(tend-tstart))

Obs.: Desconsidere as quebras de linha, é a formatação do word.

1. Orientações

Trabalho em equipe (tamanho da equipe de 2 a 5 estudantes).

Copiar o código da sua solução no item 4.

Preencher as tabelas do item 6.

Enviar esse arquivo “TDE 01 Probabilidade do Passeio Aleatório.doc” como resposta.

Não enviar o arquivo “TDE01.ipynb”.

1. Comparação de desempenho dos algoritmos

Preencher as tabelas a seguir, para o passeio de 13 passos. Número de simulações = 100000 (cem mil). Registre o tempo de simulação em segundos.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Cálculo Teórico | | | | | | | | | | | | | | |
| Posição | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| Probabilidade | 0.0000 | 0.2094 | 0.0000 | 0.1571 | 0.0000 | 0.0872 | 0.0000 | 0.0349 | 0.0000 | 0.0095 | 0.0000 | 0.0015 | 0.0000 | 0.0001 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Simulação Interativa | | | | | | | | | | | | | | |
| Posição | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| Probabilidade | 0.0000 | 0.2098 | 0.0000 | 0.1588 | 0.0000 | 0.0860 | 0.0000 | 0.0353 | 0.0000 | 0.0096 | 0.0000 | 0.0014 | 0.0000 | 0.0001 |
| Tempo de simulação | 1.8048 | 1.8049 | 1.8161 | 1.8173 | 1.7968 | 1.7988 | 1.8011 | 1.8045 | 1.8180 | 1.8126 | 1.7917 | 1.8117 | 1.8329 | 1.8082 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Simulação Vetorial | | | | | | | | | | | | | | |
| Posição | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| Probabilidade | 0.0000 | 0.2079 | 0.0000 | 0.1550 | 0.0000 | 0.0863 | 0.0000 | 0.0344 | 0.0000 | 0.0098 | 0.0000 | 0.0015 | 0.0000 | 0.0001 |
| Tempo de simulação | 1.7089 | 1.8273 | 1.6243 | 1.8588 | 1.6934 | 1.6470 | 1.7301 | 1.6395 | 1.6444 | 1.6951 | 1.6568 | 1.7236 | 1.6710 | 1.6956 |

Código para execução dos 13 passos e resultado:

\_passos = 13

\_nsim = 100000

\_pos = 0

import time

#t1 = time.perf\_counter()

probabilities, times, temp, prob = [], [], [0, 0], [0, 0, 0] # prob[pteorica, psim, psimVetorial]

for \_pos in range(\_passos+1):

    tstart = time.perf\_counter()

    \_pteorica, \_psim = Passeio(\_passos, \_pos, \_nsim)

    tend = time.perf\_counter()

    prob[0], prob[1], temp[0] = \_pteorica, \_psim, (tend - tstart)

    tstart = time.perf\_counter()

    \_psim, \_ = PasseioV(\_passos, \_pos, \_nsim)

    tend = time.perf\_counter()

    prob[2], temp[1] = \_psim, (tend - tstart)

    times.append(temp)

    probabilities.append(prob)

    print("-"\*40,"\npos:",\_pos)

    print('-'\*40)

    print("P[teórica]     = {:.5f}".format(prob[0]))

    print("P[simulada]    = {:.5f}".format(prob[1]))

    print("P[simVetorial] = {:.5f}".format(prob[2]))

    print('-'\*40)

    print("Tempo(PSimulada)    = {:.5f}".format(temp[0]))

    print("Tempo(PsimVetorial) = {:.5f}".format(temp[1]))

    print('-'\*40, '\n')

----------------------------------------

pos: 0

----------------------------------------

P[teórica] = 0.00000

P[simulada] = 0.00000

P[simVetorial] = 0.00000

----------------------------------------

Tempo(PSimulada) = 1.80488

Tempo(PsimVetorial) = 1.70892

----------------------------------------

----------------------------------------

pos: 1

----------------------------------------

P[teórica] = 0.20947

P[simulada] = 0.20987

P[simVetorial] = 0.20798

----------------------------------------

Tempo(PSimulada) = 1.80493

Tempo(PsimVetorial) = 1.82737

----------------------------------------

----------------------------------------

pos: 2

----------------------------------------

P[teórica] = 0.00000

P[simulada] = 0.00000

P[simVetorial] = 0.00000

----------------------------------------

Tempo(PSimulada) = 1.81616

Tempo(PsimVetorial) = 1.62430

----------------------------------------

----------------------------------------

pos: 3

----------------------------------------

P[teórica] = 0.15710

P[simulada] = 0.15880

P[simVetorial] = 0.15507

----------------------------------------

Tempo(PSimulada) = 1.81730

Tempo(PsimVetorial) = 1.85884

----------------------------------------

----------------------------------------

pos: 4

----------------------------------------

P[teórica] = 0.00000

P[simulada] = 0.00000

P[simVetorial] = 0.00000

----------------------------------------

Tempo(PSimulada) = 1.79681

Tempo(PsimVetorial) = 1.69347

----------------------------------------

----------------------------------------

pos: 5

----------------------------------------

P[teórica] = 0.08728

P[simulada] = 0.08604

P[simVetorial] = 0.08639

----------------------------------------

Tempo(PSimulada) = 1.79886

Tempo(PsimVetorial) = 1.64700

----------------------------------------

----------------------------------------

pos: 6

----------------------------------------

P[teórica] = 0.00000

P[simulada] = 0.00000

P[simVetorial] = 0.00000

----------------------------------------

Tempo(PSimulada) = 1.80118

Tempo(PsimVetorial) = 1.73015

----------------------------------------

----------------------------------------

pos: 7

----------------------------------------

P[teórica] = 0.03491

P[simulada] = 0.03533

P[simVetorial] = 0.03440

----------------------------------------

Tempo(PSimulada) = 1.80459

Tempo(PsimVetorial) = 1.63955

----------------------------------------

----------------------------------------

pos: 8

----------------------------------------

P[teórica] = 0.00000

P[simulada] = 0.00000

P[simVetorial] = 0.00000

----------------------------------------

Tempo(PSimulada) = 1.81808

Tempo(PsimVetorial) = 1.64442

----------------------------------------

----------------------------------------

pos: 9

----------------------------------------

P[teórica] = 0.00952

P[simulada] = 0.00967

P[simVetorial] = 0.00986

----------------------------------------

Tempo(PSimulada) = 1.81269

Tempo(PsimVetorial) = 1.69512

----------------------------------------

----------------------------------------

pos: 10

----------------------------------------

P[teórica] = 0.00000

P[simulada] = 0.00000

P[simVetorial] = 0.00000

----------------------------------------

Tempo(PSimulada) = 1.79178

Tempo(PsimVetorial) = 1.65688

----------------------------------------

----------------------------------------

pos: 11

----------------------------------------

P[teórica] = 0.00159

P[simulada] = 0.00147

P[simVetorial] = 0.00153

----------------------------------------

Tempo(PSimulada) = 1.81172

Tempo(PsimVetorial) = 1.72367

----------------------------------------

----------------------------------------

pos: 12

----------------------------------------

P[teórica] = 0.00000

P[simulada] = 0.00000

P[simVetorial] = 0.00000

----------------------------------------

Tempo(PSimulada) = 1.83299

Tempo(PsimVetorial) = 1.67108

----------------------------------------

----------------------------------------

pos: 13

----------------------------------------

P[teórica] = 0.00012

P[simulada] = 0.00014

P[simVetorial] = 0.00013

----------------------------------------

Tempo(PSimulada) = 1.80824

Tempo(PsimVetorial) = 1.69562

----------------------------------------