

TRABALHO DE MATEMÁTICA

Gustavo Pinho e Silva
Matheus Henrique de Barros Ileck
Roberta Carioca Braz

2022

1 Teorema de Laplace

O teorema de Laplace consiste em escolher uma das filas (linha ou coluna) da matriz e somar os produtos dos elementos dessa fila pelos seus respectivos cofatores.

1.1 Matriz 4x4

$$A_{4 \times 4} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

$$\text{Det } A = a_{11} \cdot A_{11} + a_{21} \cdot A_{21} + a_{31} \cdot A_{31} + a_{41} \cdot A_{41}$$

$$A_{ij} = (-1)^{i+j} \cdot D_{ij}$$

1.2 Exemplo de Matriz 4x4

$$A = \begin{bmatrix} 2 & 3 & -1 & 2 \\ 0 & 4 & -3 & 5 \\ 1 & 2 & 1 & 3 \\ 0 & 4 & 1 & 0 \end{bmatrix}$$

$$2 \cdot C_{11} + 0 \cdot C_{21} + 1 \cdot C_{31} + 0 \cdot C_{41}$$

$$C_{11} = (-1)^{1+1} \cdot \left[\begin{array}{ccc|cc} 4 & -3 & 5 & 4 & -3 \\ 2 & 1 & 3 & 2 & 1 \\ 4 & 1 & 0 & 4 & 1 \end{array} \right]$$

$$(0 - 36 + 10) - (20 + 12 + 0)$$

$$-26 - 32$$

$$\mathbf{-58}$$

$$C_{31} = (-1)^{3+1} \cdot \left[\begin{array}{ccc|cc} 3 & -1 & 2 & 3 & -1 \\ 4 & 3 & 5 & 4 & 3 \\ 4 & 1 & 0 & 4 & 1 \end{array} \right]$$

$$(0 - 20 + 8) - (-24 + 15 + 0)$$

$$-12 - (-9)$$

$$\mathbf{-3}$$

$$\text{Det } A = 2 \cdot 1 \cdot (-58) + 1 \cdot 1 \cdot (-3) = -116 - 3 = \mathbf{-119}$$

2 Determinante com Numpy e sem Numpy

2.1 Código Fonte - Python

```
1 #Inicio Determinante com Numpy
2 from numpy import matrix, linalg
3 import matplotlib.pyplot as plt
4 import time
5
6 tempo_numpy = time.perf_counter_ns()
7
8
9 matriz = matrix([[2,3,-1,2],
10                 [0,4,-3,5],
11                 [1,2,1,3],
12                 [0,4,1,0]])
13
14
15 print("A Matriz A : "\n',matriz)
16
17 resultado = linalg.det (matriz)
18
19 print("\n O derteminante da matriz A : ", round(resultado),"\n")
20
21 tempo_final = time.perf_counter_ns()
22 tempo_total_NP= (tempo_final - tempo_numpy)
23 print(f'\n O tempo de processamento do c digo com numpy foi de: {tempo_total_NP
24       } nanosegundos')
25
26 print("\n ===== \n")
27 #Fim determinante com Numpy
28
29 """
30 //////////////////////////////////////
31 """
32
33 #Inicio determinante sem Numpy
34
35 tempo_inicial = time.perf_counter_ns()
36 def remover(matriz_original,tirar_i,tirar_j):
37     matriz_nova = [[int(0) for i in range(len(matriz_original)-1)] for j in
38                     range(len(matriz_original)-1)]
39     ni = 0
40     for i in range(len(matriz_original)):
41         nj = 0
42         for j in range(len(matriz_original)):
43             if j != tirar_j:
44                 matriz_nova[ni][nj] = matriz_original[i][j]
45                 nj += 1
46             if i != tirar_i:
47                 ni += 1
48     return matriz_nova
49
```

```

def determinante(matriz_original):
51     if len(matriz_original) != len(matriz_original[0]):
        print("A matriz n o quadrada!")
53     return matriz_original
    resposta = 0
55     if len(matriz_original) == 2:
        resposta = (matriz_original[0][0] * matriz_original[1][1]) - (
            matriz_original[1][0] * matriz_original[0][1])
57     return resposta
    if len(matriz_original) > 2:
59         for j in range(len(matriz_original)):
            resposta += matriz_original[0][j] * (-1)**(0+j) * determinante(
                remover(matriz_original, 0, j))
61     return resposta

63 matriz_1 = [[2,3,-1,2],
              [0,4,-3,5],
65              [1,2,1,3],
              [0,4,1,0]]

67 for i in range(0, 4):
69     for j in range(0, 4):
        print(f'[{matriz_1[i][j]:^5}] ', end='')
71     print()

73 print("\n Determinante : ", determinante(matriz_1), "\n")
#Restante do c digo

75 #Print do tempo que demorou para rodar a parte espec fica do c digo
77 tempo_final = time.perf_counter_ns()
    tempo_total = tempo_final - tempo_inicial

79 print("\nTempo para execu o sem numpy foi de: ", tempo_total , "nanosegundos"
    , "\n")

81 #Fim determinante sem Numpy
83 Matrices=["Matriz sem numpy" , "Matriz com Numpy"]
85 vl_tempo_total=[(tempo_total)*10, tempo_total_NP]

87 plt.stem(Matrices , vl_tempo_total)
    plt.xlabel('Tamanho da Matriz')
89 plt.ylabel('Tempo de execu o em nanosegundo')
    plt.title('Rela o tempo de execu o X tamanho da Matriz')
91 plt.show()

```

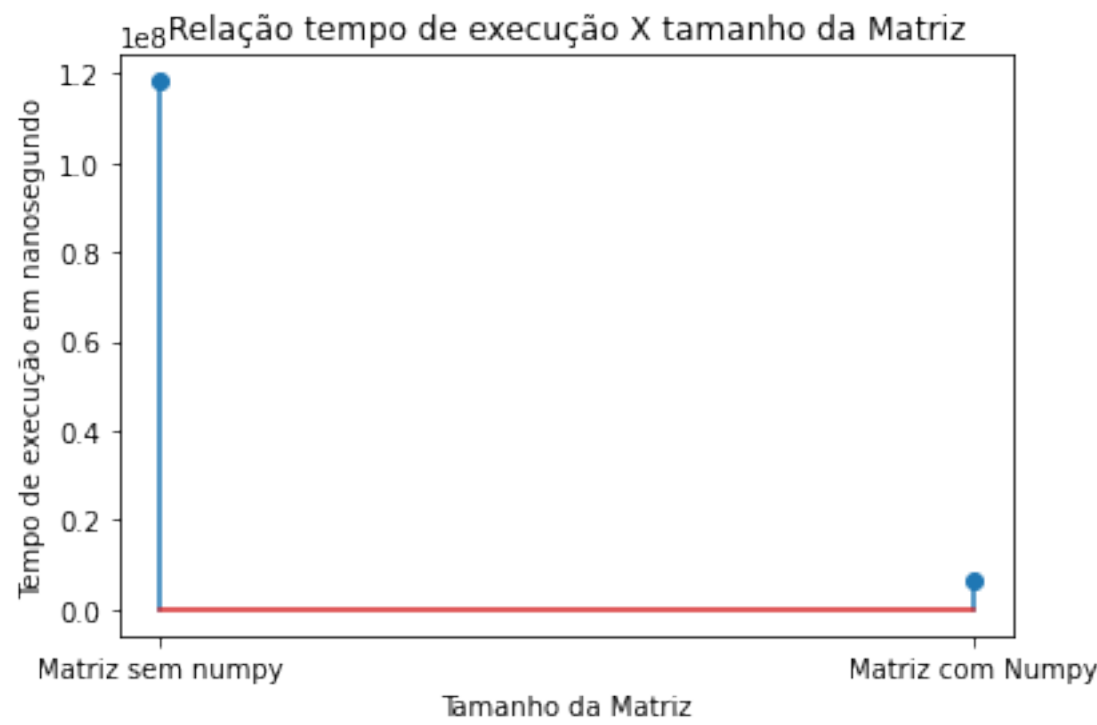
2.2 Executável

```
1 A Matriz A :  
  [[ 2  3 -1  2]  
3  [ 0  4 -3  5]  
  [ 1  2  1  3]  
5  [ 0  4  1  0]]  
  
7 O determinante da matriz A : -119  
  
9  
10 O tempo de processamento do código com numpy foi de: 6282175 nanosegundos  
11  
=====
```

[2]	[3]	[-1]	[2]
[0]	[4]	[-3]	[5]
[1]	[2]	[1]	[3]
[0]	[4]	[1]	[0]

```
17  
19 Determinante : -119  
21  
22 Tempo para execução sem numpy foi de: 11828287 nanosegundos
```

2.3 Gráfico



3 Determinante das matrizes: 2x2, 3x3, 4x4, 5x5, 6x6, 7x7, 8x8, 9x9

3.1 Código Fonte - Python

```
import numpy as np
import time
import matplotlib.pyplot as plt

# Entrada e Processamentos dos dados

ordem = [x for x in range(2, 10)] # lista com ordem da matriz
tempo = [] # lista para armazenar o tempo

# la o for para calcular o determinante das matriz
for x in range(2,10):
    tempo_inicial = time.perf_counter_ns()
    matriz_A = np.random.randint(10, size=(x, x)) # monta uma matriz com valores
    aleatorios
    determinante = np.linalg.det(matriz_A) # calcula o determinante da matriz
    tempo_final = time.perf_counter_ns()
    tempo_total = tempo_final - tempo_inicial
    # calcula o tempo total da execucao do algoritmo
    tempo.append(tempo_total)
    print(f'''Matriz =
{matriz_A}\n
Determinante da Matriz de Ordem {x} = {determinante:.0f}\n
Tempo de execu o , Matriz de Ordem {x} = {tempo_total}\n''')
    print("===== \n")
print(f'''O tamanho das Matrizes{ordem} \n
O tempo de processamento foi de: {tempo}''')

fig, ax = plt.subplots()

grfmatriz = ['2x2', '3x3', '4x4', '5x5', '6x6', '7x7', '8x8', '9x9'] # X
counts = tempo # Y
bar_labels = 'Rosa'
bar_colors = ['tab:pink']

ax.bar(grfmatriz, counts, label=bar_labels, color=bar_colors)

ax.set_ylabel('Tempo de execu o em nanosegundo')
ax.set_title('')
ax.legend(title='')

plt.show()
```

3.2 Executável

```

      Matriz =
2      [[0 5]
      [5 5]]

4      Determinante da Matriz de Ordem 2 = -25

6      Tempo de execução , Matriz de Ordem 2 = 5484624

8      Matriz =
10     [[7 1 0]
12     [3 4 6]
14     [7 3 1]]

16     Determinante da Matriz de Ordem 3 = -59

18     Tempo de execução , Matriz de Ordem 3 = 230085

20     Matriz =
22     [[2 3 1 8]
24     [6 7 8 3]
26     [4 2 7 9]
28     [3 8 7 8]]

30     Determinante da Matriz de Ordem 4 = 1180

32     Tempo de execução , Matriz de Ordem 4 = 129650

34     Matriz =
36     [[4 6 5 2 0]
38     [3 6 8 1 6]
40     [9 2 9 3 5]
42     [5 7 6 3 3]
44     [5 8 2 4 5]]

46     Determinante da Matriz de Ordem 5 = 857

48     Tempo de execução , Matriz de Ordem 5 = 110339

50     Matriz =
52     [[4 4 8 4 5 4]
54     [1 8 6 4 5 8]
     [4 0 2 4 9 9]
     [0 8 4 3 8 8]
     [8 1 5 1 7 9]
     [3 2 0 1 8 6]]

     Determinante da Matriz de Ordem 6 = 10980

     Tempo de execução , Matriz de Ordem 6 = 133838

     Matriz =
     [[6 1 6 6 6 7 4]
     [3 5 4 5 4 0 7]
     [9 2 7 1 9 3 7]
```



```

56 [3 4 4 4 9 9 2]
[5 1 7 5 6 2 7]
58 [2 4 7 8 4 7 7]
[4 0 3 0 2 4 7]]

60 Determinante da Matriz de Ordem 7 = -116125

62 Tempo de execu o , Matriz de Ordem 7 = 666494

64 Matriz =
[[1 3 1 7 3 6 1 0]
66 [2 6 0 3 9 7 2 8]
[9 3 8 0 7 2 6 4]
68 [9 0 0 6 8 1 8 6]
[4 9 8 7 4 2 8 8]
70 [6 6 6 0 6 8 6 0]
[0 6 7 8 4 8 0 9]
72 [3 5 4 3 6 2 7 9]]

74 Determinante da Matriz de Ordem 8 = -3581260

76 Tempo de execu o , Matriz de Ordem 8 = 714851

78 Matriz =
[[6 7 2 3 8 3 2 7 5]
80 [0 9 6 5 4 9 2 5 0]
[8 6 9 6 4 3 0 6 1]
82 [4 2 5 4 3 6 7 3 9]
[5 0 3 2 0 2 0 0 4]
84 [2 4 7 0 3 3 9 1 9]
[9 4 6 9 4 8 9 2 4]
86 [0 9 6 9 1 6 1 1 9]
[1 6 9 8 7 1 5 9 7]]

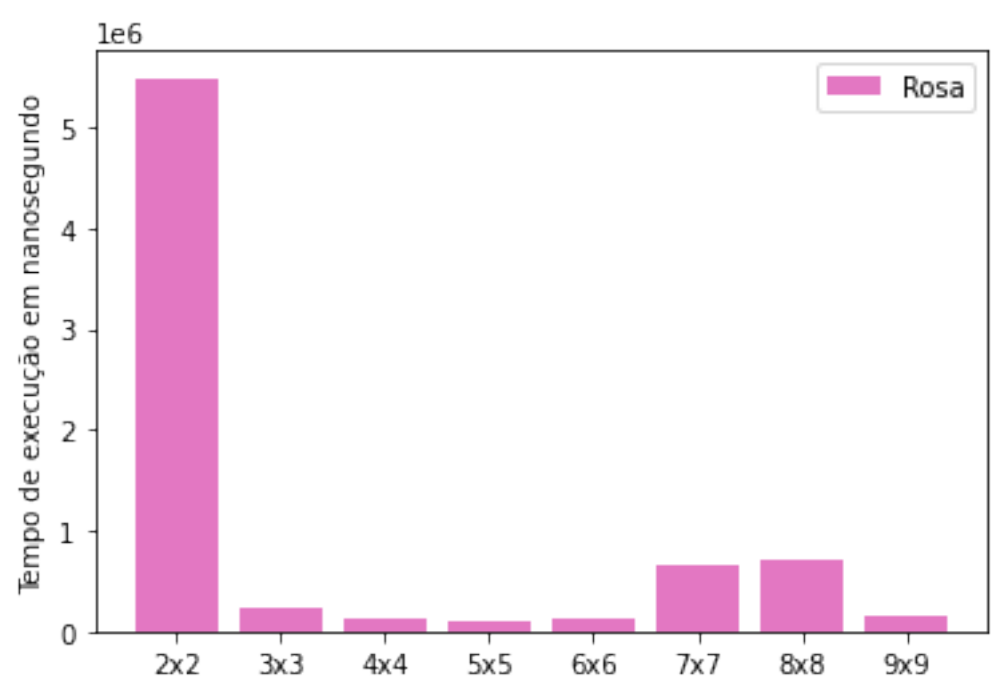
88 Determinante da Matriz de Ordem 9 = -27315244

90 Tempo de execu o , Matriz de Ordem 9 = 141106

92
=====
94 O tamanho das Matrizes[2, 3, 4, 5, 6, 7, 8, 9]
96 O tempo de processamento foi de: [5484624, 230085, 129650, 110339, 133838,
666494, 714851, 141106]

```

3.3 Gráfico



4 GitHub

Gustavo Pinho e Silva

Matheus Henrique de Barros Ileck

Roberta Carioca Braz