Exercicios 01 - Cálculo Numérico

Wanderson Faustino Patricio

```
# Antes das soluções vamos importar as bibliotecas necessárias
import numpy as np
import numpy.linalg as la
from math import *
```

```
def badSolver(A : np.array, b : np.array) -> np.array:
    """Solves the system Ax = b
    A and b must be of the np.array
    """
    At = A.transpose()
    AtA = At.dot(A)
    Atb = At.dot(b)

    return la.solve(AtA, Atb)
```

```
def linAlgSolver(A : np.array, b : np.array) -> np.array:
    """Solves the system Ax = b with the library np.linalg
    A and b must be of the np.array
    """
    return la.lstsq(A, b)[0]
```

Questão 01

Resolva o sistema abaixo usando o metodo dos mínimos quadrados.

$$\begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix}$$

Solução Analítica

Multiplicando pela transposta da primeira matriz em ambos os lados

$$\begin{bmatrix} 1 & 2 & 3 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 14 & 6 \\ 6 & 3 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 5 \\ 3 \end{bmatrix}$$

A solução será

$$\begin{bmatrix} x_o \\ x_1 \end{bmatrix} = \begin{bmatrix} -\frac{1}{2} \\ 2 \end{bmatrix}$$

Solução Numérica

```
A = np.array([[1, 1],[2,1],[3,1]])
b = np.array([1,2,0])

badSolution = badSolver(A, b)
linalgSolution = linAlgSolver(A, b)
```

```
print("Bad Solution:")
badSolution
```

```
Bad Solution:
array([-0.5,  2. ])
```

```
print("linAlg Solution:")
linalgSolution
```

```
linAlg Solution:
```

```
array([-0.5,  2. ])
```

Questão 02

a) Aplicando os valores

- $f(0) = a + b = 26$
- $f(6) = a + c = 21$
- $f(12) = a - b = 34$
- $f(18) = a - c = 30$

teremos o seguinte sistema

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 26 \\ 21 \\ 34 \\ 30 \end{bmatrix}$$

b) Seguindo a mesma ideia do item anterior

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} 26 \\ 21 \\ 34 \\ 30 \end{bmatrix}$$

$$\begin{bmatrix} 4 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 111 \\ -8 \\ -9 \end{bmatrix}$$

O que nos dá a solução

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \frac{111}{4} \\ -4 \\ -\frac{9}{2} \end{bmatrix} = \begin{bmatrix} 27.25 \\ -4 \\ -4.5 \end{bmatrix}$$

```
A = np.array([[1, 1, 0],[1, 0, 1],[1, -1, 0],[1,0,-1]])
b = np.array([26, 21, 34, 30])

badSolution = badSolver(A, b)
linalgSolution = linAlgSolver(A, b)
```

```
print("Bad Solution:")
badSolution
```

```
Bad Solution:
array([27.75, -4.  , -4.5 ])
```

```
print("linAlg Solution:")
linalgSolution
```

```
linAlg Solution:
array([27.75, -4.  , -4.5 ])
```

c) $T(14) = 27.75 - 4 \cdot \cos\left(\frac{7\pi}{6}\right) - 7.5 \cdot \sin\left(\frac{7\pi}{6}\right) = 33.464^\circ C$

```
linalgSolution[0] + linalgSolution[1] * cos((7*pi)/6) + linalgSolution[2] * sin((7*pi)/6)
```

```
33.46410161513776
```

Questão 03

a) Aplicando os valores

- $f(0) = a + c = 0$
- $f(0.5) = a \cdot \sqrt{e} + c = 0.57$
- $f(1) = a \cdot e + c = 1.46$

- $f(2) = a \cdot e^2 + c = 5.05$

teremos o seguinte sistema

$$\begin{bmatrix} 1 & 1 \\ \sqrt{e} & 1 \\ e & 1 \\ e^2 & 1 \end{bmatrix} \begin{bmatrix} a \\ c \end{bmatrix} = \begin{bmatrix} 0 \\ 0.57 \\ 1.46 \\ 5.05 \end{bmatrix}$$

Fazendo mínimos quadrados

$$\begin{bmatrix} 1 & \sqrt{e} & e & e^2 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ \sqrt{e} & 1 \\ e & 1 \\ e^2 & 1 \end{bmatrix} \begin{bmatrix} a \\ c \end{bmatrix} = \begin{bmatrix} 1 & \sqrt{e} & e & e^2 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0.57 \\ 1.46 \\ 5.05 \end{bmatrix}$$

$$\begin{bmatrix} 65.704 & 12.756 \\ 12.756 & 4 \end{bmatrix} \begin{bmatrix} a \\ c \end{bmatrix} = \begin{bmatrix} 42.223 \\ 7.08 \end{bmatrix}$$

A solução será

$$\begin{bmatrix} a \\ c \end{bmatrix} = \begin{bmatrix} 0.785 \\ -0.733 \end{bmatrix}$$

```
A = np.array([[1, 1],[sqrt(e), 1],[e, 1],[e**2,1]])
b = np.array([0,0.57,1.46,5.05])

badSolution = badSolver(A, b)
linalgSolution = linAlgSolver(A, b)


print("Bad Solution:")
badSolution
```

```
Bad Solution:
array([ 0.78497536, -0.73329805])
```

```
print("linAlg Solution:")
linalgSolution
```

```
linAlg Solution:
array([ 0.78497536, -0.73329805])
```

b) Aplicando os valores de a e c para achar b:

- $f(0) = 0.785 \cdot e^{b \cdot 0} - 0.733 = 0$
- $f(0.5) = 0.785 \cdot e^{b \cdot 0.5} - 0.733 = 0.57$
- $f(1) = 0.785 \cdot e^{b \cdot 1} - 0.733 = 1.46$
- $f(2) = 0.785 \cdot e^{b \cdot 2} - 0.733 = 5.05$

Organizando

- $e^{b \cdot 0} = 0.934$
- $e^{b \cdot 0.5} = 1.660$
- $e^{b \cdot 1} = 2.794$
- $e^{b \cdot 2} = 7.367$

Encontramos portanto o sistema

$$\begin{bmatrix} 0 \\ 0.5 \\ 1 \\ 2 \end{bmatrix} [b] = \begin{bmatrix} \ln(0.934) \\ \ln(1.660) \\ \ln(2.794) \\ \ln(7.367) \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0.5 & 1 & 2 \end{bmatrix} \begin{bmatrix} 0 \\ 0.5 \\ 1 \\ 2 \end{bmatrix} [b] = \begin{bmatrix} 0 & 0.5 & 1 & 2 \end{bmatrix} \begin{bmatrix} \ln(0.934) \\ \ln(1.660) \\ \ln(2.794) \\ \ln(7.367) \end{bmatrix}$$

$$[5.25][b] = [5, 275]$$

Portanto:

$$b = 1.00476$$

```
A = np.array([[0], [0.5], [1], [2]])
b = np.log(np.array([0.934, 1.66, 2.794, 7.367]))

badSolution = badSolver(A, b)
linalgSolution = linAlgSolver(A, b)


print("Bad Solution:")
badSolution
```

```
array([1.00474366])
```

```
print("linAlg Solution:")
linalgSolution
```

```
linAlg Solution:
array([1.00474366])
```

c) Faremos agora o cálculo do coeficiente de correlação dos dois modelos.

```
def mean(lista):
    sum = 0
    for item in lista:
        sum += item

    return sum/lista.size

def R(y_real, y_pred):
    n = y_real.size
    y_mean = mean(y_real)
    SQtot = 0
    for index in range(n):
        SQtot += (y_real[index]- y_mean)**2

    SQres = 0
    for index in range(n):
        SQres += (y_real[index]- y_pred[index])**2

    coeff = 1 - SQres/SQtot
    return coeff


y_real = np.array([0,0.57,1.46,5.05])
x = np.array([0, 0.5, 1, 2])
y_pred1 = 0.785 * np.exp(x) - 0.733
y_pred2 = 0.785 * np.exp(1.00474366*x) - 0.733

R1 = R(y_real, y_pred1)
R2 = R(y_real, y_pred2)

print(f'Coeficiente da abordagem 1 : {R1}')
print(f'Coeficiente da abordagem 2 : {R2}')
```

```
Coeficiente da abordagem 1 : 0.9995733383453307
Coeficiente da abordagem 2 : 0.999324385247391
```

A primeira abordagem gerou um coeficiente de correlação maior, ou seja, foi mais preciso

Questão 04

a) Trabalhando um pouco a equação, podemos chegar ao seguinte resultado

$$\sqrt{y} = \frac{a}{b} \cdot \frac{1}{\sqrt{x}} + \frac{1}{b}$$

Ou seja, uma equação do tipo

$$u = m \cdot v + n$$

com $u = \sqrt{y}, v = \dfrac{1}{\sqrt{x}}, m = \dfrac{a}{b}$ e $n = \dfrac{1}{b}$.

b) Reescrevendo a tabela

$$u = [3.225, 2.408, 1.817, 1.549, 1.414]$$
$$v = [1.414, 1, 0.707, 0.577, 0.5]$$

Teremos o seguinte sistema

$$\begin{bmatrix} 1.414 & 1 \\ 1 & 1 \\ 0.707 & 1 \\ 0.577 & 1 \\ 0.5 & 1 \end{bmatrix} \begin{bmatrix} m \\ n \end{bmatrix} = \begin{bmatrix} 3.225 \\ 2.408 \\ 1.817 \\ 1.549 \\ 1.414 \end{bmatrix}$$

Aplicando o método dos mínimos quadrados

$$\begin{bmatrix} 1.414 & 1 & 0.707 & 0.577 & 0.5 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1.414 & 1 \\ 1 & 1 \\ 0.707 & 1 \\ 0.577 & 1 \\ 0.5 & 1 \end{bmatrix} \begin{bmatrix} m \\ n \end{bmatrix} = \begin{bmatrix} 1.414 & 1 & 0.707 & 0.577 & 0.5 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 3.225 \\ 2.408 \\ 1.817 \\ 1.549 \\ 1.414 \end{bmatrix}$$

$$\begin{bmatrix} 4.082 & 4.198 \\ 4.198 & 5 \end{bmatrix} \begin{bmatrix} m \\ n \end{bmatrix} = \begin{bmatrix} 9.854 \\ 10.413 \end{bmatrix}$$

O que nos conduz à solução

$$\begin{bmatrix} m \\ n \end{bmatrix} = \begin{bmatrix} 1.992 \\ 0.410 \end{bmatrix}$$

```
A = np.array([[1.414, 1], [1, 1], [0.707, 1], [0.577, 1], [0.5, 1]])
b = np.array([3.225, 2.408, 1.817, 1.549, 1.414])

badSolution = badSolver(A, b)
linalgSolution = linAlgSolver(A, b)
```

```
print("Bad Solution:")
badSolution
```

```
Bad Solution:
array([1.99232476, 0.40984413])
```

```
print("linAlg Solution:")
linalgSolution
```

```
linAlg Solution:
array([1.99232476, 0.40984413])
```

Temos

$$n = \frac{1}{b} = 0.410 \Rightarrow b = 2.439$$
$$m = \frac{a}{b} \Rightarrow 1.992 = \frac{a}{2.439} \Rightarrow a = 4.858$$