TECHMIMO

Autor: Rafael Pereira da Silva

Seguem alguns recados para ajudá-los e para contribuir com o curso:

- Fiquem à vontade para me contatar pelo Linkdin, costumo responder por lá também: https://www.linkedin.com/in/rafael-pereira-da-silva-23890799/ (https://www.linkedin.com/in/rafael-pereira-da-silva-23890799/)
- Fiquem a vontade para compartilharem o certificado do curso no Linkedin. Eu costumo curtir e comentar para dar mais credibilidade
- Vocês podem usar esses notebooks para resolver os exercícios e desafios
- Não se esqueçam de avaliar o curso e dar feedback, eu costumo criar conteúdos baseado nas demandas de vocês
- Se tiverem gostando do curso, recomendem aos amigos, pois isso também ajuda a impulsionar e a crescer a comunidade
- Bons estudos e grande abraços!

Seção 7 - Spyder e Sympy

7.1 Introdução ao Spyder a ao SymPy

O que é Spyder

Sypyder é um ambiente similar a *softwares* como Matlab ou Mathematica. Ele contém funcionalidades interessantes para quem desenvolve trabalhos científicos. Para mais informações veja: https://www.spyder-ide.org/ (https://www.spyder-ide.org/)

Algumas de suas funcionalidades são:

- Editor
- Ipython Console
- Explorador de variáveis
- Para outras funcionalidades acesse: https://docs.spyder-ide.org/overview.html (https://docs.spyder-ide.org/overview.html)

O que é SymPy

SymPy é uma biblioteca para trabalhar com matemática simbólica.

Algumas funcionalidades são:

- Trabalhar com matrizes
- Realizar cálculo de derivadas, integrais, etc.
- Utilizar *solvers*
- Para outras funcionalidades acesse: https://docs.sympy.org/latest/index.html (https://docs.sympy.org/latest/index.html)

7.2 Símbolos

- Função **symbols(** *string_de_símbolo* **)**
- Função **init_printing()**
- Método **subs(** *símbolos* , *valores* **)**
- Função **simplify**(*equação*)
- Operações booleanas funcionam com símbolos
- Para outras maneiras de simplificar, acesse: [https://docs.sympy.org/latest/tutorial/simplification.html (https://docs.sympy.org/latest/tutorial/simplification.html)](https://docs.sympy.org/latest/tutorial/simplification.html)

In [9]:

```
import sympy as sp

x = sp.symbols('x')

x*x
```

Out[9]:

```
x**2
```

In [13]:

```
sp.init_printing()

Lista_simbolos = ['y','x']
y,x = sp.symbols(Lista_simbolos)

f_x_y = x**2 + y**2 + x*y
f_x_y
```

Out[13]:

$$x^2 + xy + y^2$$

In [14]:

```
resposta = f_x_y.subs(x,1).subs(y,2)
resposta
```

Out[14]:

$$7$$

In [16]:

```
f_xy = (x+y)**2 + y**2 + 2*x*y + y**2
sp.simplify(f_xy)
```

Out[16]:

$$x^2 + 4xy + 3y^2$$

In [22]:

```
x <= y
```

Out[22]:

$x \leq y$

In [ ]:

```

```

7.3 - Matrizes

- Função **Matrix([])**
- Propriedade **.shape**
- Matrizes úteis **eye(** *tamanho*), **zeros(** *i,j*), **ones(** *i,j*),
- **diag(** *lista*), no caso do diag pode ser usado * antes da lista ou o argumento unpack=True
- Operações comuns: *+,-, **
- Para inverter a matriz, basta elevar a -1
- Para transpor use o método **.T**
- Para achar o determinante use o método **.det()**
- Para mais informações acesse: [https://docs.sympy.org/latest/tutorial/matrices.html (https://docs.sympy.org/latest/tutorial/matrices.html)](https://docs.sympy.org/latest/tutorial/matrices.html)

In [83]:

```
import sympy as sp
sp.init_printing()

A = sp.Matrix([[1,2,3],[4,5,6]])
A.shape
```

Out[83]:

$(2, \quad 3)$

In [89]:

```
B = sp.Matrix([[1,2,3]])
C = sp.Matrix([[1],[2],[3]])
D = sp.Matrix([1,2,3])

B == D
```

Out[89]:

```
False
```

In [102]:

```
I = sp.zeros(3)


Lista_E = [7,8,9]
E = sp.diag(*Lista_E)

E.det()
```

Out[102]:

$$504$$

In [106]:

```
## Sistema linear
x1,x2,x3 = sp.symbols(['x1','x2','x3'])

X = sp.Matrix([x1,x2,x3])
E = sp.Matrix([[1,2,3],[4,5,6],[7,8,9]])
R = sp.Matrix([11,22,33])

## EX = R

E*X - R
```

Out[106]:

$$\begin{bmatrix} x_1 + 2x_2 + 3x_3 - 11 \\ 4x_1 + 5x_2 + 6x_3 - 22 \\ 7x_1 + 8x_2 + 9x_3 - 33 \end{bmatrix}$$

In [ ]:

In [ ]:

In [ ]:

In [ ]:

7.4 - Funções do cálculo

- Derivada **diff(** *função*,*variável* **)**
- Integral **integrate(** *função*,(*variável*,*início*,*fim*)**)**
- Outras funções disponíveis são: limites, expansão de séries, diferenças finitas.

- Para mais informações acesse: https://docs.sympy.org/latest/tutorial/calculus.html (https://docs.sympy.org/latest/tutorial/calculus.html)

In [39]:

```
import sympy as sp
sp.init_printing()
import math as mt
import numpy as np


a = sp.pi*2 + 198*sp.pi
a
```

Out[39]:

$$200\pi$$

In [43]:

```
x,y = sp.symbols(['x','y'])

f_x = sp.sin(x)

dfdx = sp.diff(f_x,x)
d2fdx2 = sp.diff(f_x,x,x)

d2fdx2
```

Out[43]:

$$-\sin(x)$$

In [44]:

```
f_x_y = x**2 + y**2 - 100

dfdy = sp.diff(f_x_y,y)
dfdy
```

Out[44]:

$$2y$$

In [47]:

```
C1 = sp.symbols('C1')

f_x2 = sp.sin(x)**2

int_f_x2 = sp.integrate(f_x2,x)

int_f_x2 += C1
int_f_x2
```

Out[47]:

$$C_1 + \frac{x}{2} - \frac{1}{2}\sin(x)\cos(x)$$

In [50]:

```
f_x3 = sp.exp(-x) #exp é função exponencial

sp.oo #infinito

sp.integrate(f_x3,(x,0,sp.oo))
```

Out[50]:

$$1$$

In [ ]:

In [ ]:

In [ ]:

In [ ]:

7.5 Solvers

- **solveset**(*eq*, *args* **)** ou **solve**
- **dsolve**(*eq*, *args* **)**, **Function('fx')** e **Derivative(** *fx*,*args* **)**
- Acesse: [https://docs.sympy.org/latest/tutorial/solvers.html (https://docs.sympy.org/latest/tutorial/solvers.html)](https://docs.sympy.org/latest/tutorial/solvers.html)
- Acesse: [https://docs.sympy.org/latest/modules/solvers/solveset.html#sympy.solvers.solveset.nonlinsolve (https://docs.sympy.org/latest/modules/solvers/solveset.html#sympy.solvers.solveset.nonlinsolve)](https://docs.sympy.org/latest/modules/solvers/solveset.html#sympy.solvers.solveset.nonlinsolve)

In [ ]:

```
import sympy as sp
sp.init_printing()

x = sp.symbols('x')
```

In [65]:

```
f_x = x**3 + 5.*x**2 + x

sp.solveset(f_x,x)

x1 = -4.79128784747792
f_x.subs(x,x1)
```

Out[65]:

$$-1.4210854715202 \cdot 10^{-14}$$

In [82]:

```
A = sp.Matrix([[3,2,4],[1,1,2],[4,3,-2]])

x1,x2,x3 = sp.symbols(['x1','x2','x3'])

X = sp.Matrix([[x1,x2,x3]]).T

R =sp.Matrix([[1,2,3]]).T

## AX = R
## AX - R = 0

Sistema = A*X -R

res = sp.solve(Sistema,(x1,x2,x3))
res
```

Out[82]:

$$\{x_1 : -3, \quad x_2 : 5, \quad x_3 : 0\}$$

In [84]:

```
A_nl = sp.Matrix([[x1 + x2 -3],[x1**2 +x2**2 - 9]])

res_nl = sp.solve(A_nl,(x1,x2))
res_nl
```

Out[84]:

$$[(0, \quad 3), \quad (3, \quad 0)]$$

In [ ]:

```
# Equações diferenciais
```

In [91]:

```
y_x = sp.Function('y_x')

d2y_x = sp.Derivative(y_x(x),x,x)

eq = d2y_x - 5

print(sp.dsolve(eq))
```

```
Eq(y_x(x), C1 + C2*x + 5*x**2/2)
```

In [ ]:

In [ ]:

In [ ]:

7.6 Exercício 1

Dadas as matrizes A,B, C e D, calcule, para cada uma, o determinante, a matriz transposta e a matriz inversa.

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

$$B = \begin{bmatrix} cos(\pi/4) & -sen(\pi/4) \\ \\ sen(\pi/4) & cos(\pi/4) \end{bmatrix}$$

$$C = \begin{bmatrix} 2 & -2 \\ \\ -2 & 5 \end{bmatrix}$$

$$D = \begin{bmatrix} 2 & 2 \\ \\ 4 & 4 \end{bmatrix}$$

Notas:

- A tem determinante diferente de zero
- B é uma matriz orgonal, sua inversa é igual a sua transposta, e seu det deve ser + ou -1
- C é uma matriz simétrica, ela é igual a sua transposta
- D possui determinante igual a zero, portanto não é inversível

In [ ]:

In [ ]:

In [ ]:

7.7 Exercício 2

Dadas as funções, calcule sua derivada de primeira ordem e sua primitiva (integral sem limites de integração).

$$A(x) = e^x$$

$$B(x) = x^3$$

$$C(x) = 1/x$$

Nota: O sympy não acrescenta constantes para integrais indefinidas, elas devem ser acrescentadas manualmente.

In [ ]:

In [ ]:

In [ ]:

7.8 Exercício 3

Calcule a integral dupla:

$$A(x, y) = \iint_A dx.\,dy$$

Considere: $0 \leq x \leq 3$ e $0 \leq y \leq 4$

In [ ]:

In [ ]:

In [ ]:

```

```

In [ ]:

```

```

7.9 Exercício 4

Ache as raízes das equações:

$$A(x) = x^3 + 5x^2$$

$$B(x) = x^2 + 9$$

$$C(x) = sen(x)$$

$$D(x) = x^2 + cos(x)$$

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

7.10 Exercício 5

Dadas as matrizes, resolva o sistema de equação $[A].\{X\} = \{B\}$

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 2 & 1 & 3 \end{bmatrix}$$

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$B = \begin{bmatrix} 6 \\ \\ 9 \\ \\ 11 \end{bmatrix}$$

In [ ]:

In [ ]:

In [ ]:

7.11 Exercício 6

Resolva as EDOs:

$$A)\ \ \frac{dy}{dx} = \frac{x^2}{y}$$

$$B)\ \ m\ddot{x}(t) + c\dot{x}(t) + kx(t) = 0;\ \ com\ x(0) = x_0\ e\ \dot{x}(0) = v_0$$

In [ ]:

In [ ]:

7.12 Desafio 1

Calcule a área da superfície de uma semi-esfera baseada em sua equação: $x^2 + y^2 + z^2 = 100$

Equação paramétrica da esfera:

$$\sigma(\theta, \phi) = \begin{cases} x = r.\,sen(\phi).\,cos(\theta) \\ \\ y = r.\,sen(\phi).\,sen(\theta) \\ \\ z = r.\,cos(\phi) \end{cases}$$

A área de σ pode ser calculada como:

$$A_\sigma = \iint_K \left\| \frac{\partial \sigma}{\partial \theta} \wedge \frac{\partial \sigma}{\partial \phi} \right\| d\theta d\phi$$

Considere a semi-esfera: $0 \leq \phi \leq \pi/2$ e $0 \leq \theta \leq 2\pi$

Referências:

- GUIDORIZZI, Hamilton Luiz. Um curso de cálculo volume 3. 5a edição, LTC, 2002.
- https://www.ime.unicamp.br/~valle/Teaching/MA211/Aula23.pdf (https://www.ime.unicamp.br/~valle/Teaching/MA211/Aula23.pdf)
- https://socratic.org/questions/58e321437c014904021733e9 (https://socratic.org/questions/58e321437c014904021733e9)

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [1]:

```
# Resolvido no Spyder
```

7.13 - Desafio 2 - Longarina de uma aeronave

Projete uma longarina considerndo as tensões devido ao momento fletor. Considere uma asa de aeronave que opera com as seguintes características:

- Dimensões geométricas: $b = 2m$;
- Peso máximo de decolagem é $W = 100N$
- Fator de carga $n_{max} = 2,3$
- Longarina de alumínio com $E = 70.10^9 Pa$ e $sigma_{admissível} = 500.10^6 Pa$

A força de sustentação (distribuição elíptica) na asa pode ser calculada por meio das fórmulas:

$$\begin{cases} L = n_{max} \, . \, W \\ \\ L(y)_E = \frac{4.L}{b.\pi} \sqrt{1 - \left(\frac{2.y}{b}\right)^2} \end{cases}$$

Equações da linha elástica (nas coordenadas da figura):

$$\begin{cases} E.\, I \frac{d^4 v}{dy^4} = -L(y) \\ \\ E.\, I \frac{d^3 v}{dy^3} = V(y) \\ \\ E.\, I \frac{d^2 v}{dy^2} = M(y) \end{cases}$$

Tensão devido ao momento fletor:

$$\begin{cases} \sigma_f = \frac{M(y).c}{I} \\ \\ I = \frac{d_1.h^3}{12} \end{cases}$$

- Para facilitar, podemos considerar -1 < y < 1 e utilizar o trecho 0 < y < 1.
- Considere M(1) = 0 e V(1) = 0

Referências:

- RODRIGUES, Luiz Eduardo Miranda José. Engenharia Aeronáutica. Cengage Learning, 2014.
- HIBBELER, R. C. Resistência dos Materiais. 5a ed. São Paulo: Pearson Prentice Hall, 2004.

In [ ]:

```

```

In [1]:

```
#Resolvido no Spyder
```

7.14 Desafio 3 - Impacto de um veículo passando em um tronco

Calcule o fator de carga do impacto de um veículo passando em um tronco. Dados:

- $m = 200kg$
- $k = 1.10^4\,N/m$
- $c = 5.10^2\,N.s/m$
- $v = 30km/h$

Equação de movimento:

$$m\ddot{x} + c(\dot{x} - \dot{y}) + k(x - y) = 0$$

Força transmitida ao carro:

$$F(t) = k(x - y) + c(\dot{x} - \dot{y})$$

Modelo do tronco:

$$y(t) = Y.sen(\omega_b.t)$$

Referência:

- INMAN, Daniel J. Engineering Vibration. New Jersey: Pearson Prentice Hell: 2008. pg 131, 136

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```