

• Packages usados nas aulas:

```
* import math as mt
* import matplotlib.pyplot as plt
* import numpy as np
  o from numpy.linalg import matrix_power as mpower
```

• Cálculo numérico / Funções matemáticas / Constantes relevantes:

Comandos de cálculo numérico
+
-
*
/
**

Constantes relevantes
mt.pi
mt.e

Funções matemáticas
mt.floor(x)
mt.ceil(x)
int(x)
k % m
k//q
mt.factorial(n)

Funções matemáticas elementares
mt.pow(x,p)
mt.exp(x)
mt.log(x)
mt.log10(x)
mt.log2(x)
abs(x)
mt.sin(x)
mt.cos(x)
mt.tan(x)
mt.asin(x)
mt.acos(x)
mt.atan(x)
round(x)
round(x,k)
max(x1,x2,...)

• Entrada e saída de dados:

Comandos úteis na entrada/saída de dados
print
str
+
int
float
input
print(f"Texto {x} texto {y}.")

• Strings:

Strings
string[n]
string[i:j]
len(string)

- Constantes lógicas/Operadores Lógicos/Operadores Relacionais:

Constantes lógicas
True
False

Operadores Lógicos
~ , not
&, and
, or

Operadores Relacionais
==
!=
<
>
<=
>=

- Matrizes:

Comandos básicos para a construção de um vetor linha
x=np.arange(x0,x1)
x=np.arange(x0,x1,step)
x=np.linspace(x0,x1,n)

Definição de vetor / matriz
A=np.array([x <sub>0</sub> ,x <sub>1</sub> ,...,x <sub>n</sub> ])
A=np.array([[x <sub>00</sub> ,...,x <sub>0n</sub> ],...,[x <sub>m0</sub> ,...,x <sub>mn</sub> ]])
A.size
A.shape

Características de uma matriz
A[a,b]
A[a,:]
A[:,b]
A[:,:]
A[i:j,:]
np.delete(A,i,axis)
np.append(A,x)
np.array_equal(A,B) ou (A==B).all()
np.concatenate((array1,array2),axis=0)
np.concatenate((array1,array2),axis=1)

Matrizes específicas
np.eye(m)
np.zeros((m,n))
np.ones((m,n))
np.empty((m,n))

Alteração de elementos da matriz
A[A!=0]=1

Operadores sobre matrizes com a mesma dimensão (elemento a elemento)
C=A+B
C=A-B
C=A*B
C=A**k

Funções matriciais
A.T
A@B
np.linalg.det(A)
np.trace(A)
np.linalg.inv(A)
np.linalg.matrix_power(A,k)

- Definição de uma função:

```
def nome(parâmetros):
    corpo
```

- Visualização gráfica:

Visualização gráfica
<code>plt.plot(X,Y)</code>
<code>plt.plot(X, Y, 'S')</code>

Inserção de texto em gráficos
<code>plt.legend(...)</code>
<code>plt.title(string_título)</code>
<code>plt.xlabel(string_legenda_xx)</code>
<code>plt.ylabel(string_legenda_yy)</code>

Caractere	Cor	Caractere	Marcador	Caractere	Tipo de linha
b	azul	.	ponto	-	linha contínua
g	verde	o	círculo	- -	linha tracejada
r	vermelho	x	X	-.	traço e ponto

- Estruturas de controlo:

<b>if ...:</b>
...
<b>elif...:</b>
...
<b>else:</b>
...

<b>for ...:</b>
...

<b>while ...:</b>
...

- Teoria de conjuntos:

Teoria de conjuntos
$A \& B$
ou
<code>A.intersection(B)</code>
$A B$
ou
<code>A.union(B)</code>
$A-B$
ou
<code>A.difference(B)</code>
$A^B$
<code>len(A)</code>
<code>A==B</code>
<code>A&lt;B</code>
<code>x in A</code>
<code>x not in A</code>
<code>A.add(x)</code>
<code>A.remove(x)</code>
<code>A.discard(x)</code>
<code>A.clear()</code>

- Comandos a usar em listas:

Comandos a usar em listas
<code>L=list()</code> ou <code>L=[]</code>
<code>L[i]</code>
<code>L[i]=x</code>
<code>L[i:j]</code>
<code>del L[i]</code>
<code>L.clear()</code>
<code>len(L)</code>
<code>L.count(x)</code>
<code>L.extend(L2)</code>
<code>L.append(x)</code>
<code>L.insert(i,x)</code>
<code>L.remove(x)</code>
<code>L.index(x)</code>
<code>L.sort()</code>
<code>L.sort(reverse=True)</code>
<code>L2=sorted(L)</code>
<code>list(string)</code>
<code>L=list([string1,string2])</code>
<code>' '.join(L)</code>
<code>L1+L2</code>
<code>L.pop(i)</code>
<code>L3=L.copy()</code>

- Comandos para ler informação de um ficheiro Excel:

Comandos para ler informação de um ficheiro Excel
import pandas as pd
FICHEIRO = pd.ExcelFile(r'CCC\ficheiro.xlsx')
F1 = pd.read_excel(FICHEIRO, 'Folha1', header=None)
A=F1.to_numpy()

- Outros comandos:

M.astype(...)
np.inf
P=M.copy()