

NumPy – Computação Numérica

Python para Sala de Aula de Matemática

Luis Alberto D'Afonseca

CEFET-MG

Conteúdo

Computação Numérica

Criando Vetores e Matrizes

Operações

Álgebra Linear

NumPy

- ▶ Computação numérica
- ▶ Vetores e matrizes: `array`
- ▶ Elementos do mesmo tipo
- ▶ Eficiência computacional
- ▶ Funcionalidades mais avançadas



<https://numpy.org>

Importando a Biblioteca

```
import numpy as np
```

Conteúdo

Computação Numérica

Criando Vetores e Matrizes

Operações

Álgebra Linear

Criando Vetores

Listando elemento a elemento

```
x = np.array( [ 1, 2, 5, 6, 7 ] )  
print(x)
```

Utilizando uma função

```
u = np.linspace( -5, 5, 100 )  
print(u)
```

```
v = np.zeros( (5, 7) )  
print(v)
```

```
w = np.arange( 0, 1, 0.1 )  
print(w)
```

Criando Matrizes

Listando elemento a elemento

```
P=np.array([[1, 2], [3, 4]])  
print(P)
```

```
Q=np.array([[10, 20, 30], [11, 21, 31], [12, 22, 33]])  
print(Q)
```

Criando Matrizes

Utilizando uma função

```
Z = np.zeros( (2, 3) )  
print(Z)
```

```
U = np.ones( (3, 2) )  
print(U)
```

```
I = np.eye( 3 )  
print(I)
```

```
D = np.diag( [1, 2, 3] )  
print(D)
```


Conteúdo

Computação Numérica

Criando Vetores e Matrizes

Operações

Álgebra Linear

Operações Elemento a Elemento com Vetores

```
y = x**2 + 2*x - 3  
print(y)
```

```
f = np.sin(x)  
print(f)
```

```
g = np.exp(x)  
print(g)
```

Definindo Matrizes

```
A = np.array( [ [1, 2] , [3, 4] ] )
```

```
B = np.array( [ [5, 6] , [7, 8] ] )
```

```
print(A)
```

```
print(B)
```

Operações Elemento a Elemento com Matrizes

Soma

```
print(A + B)
```

Subtração

```
print(A - B)
```

Multiplicação elemento a elemento

```
print(A * B)
```

Divisão elemento a elemento

```
print(A / B)
```

Operações Matriciais

Produto matricial

```
print(A @ B)
```

Transposta

```
print(A.T)
```

Conteúdo

Computação Numérica

Criando Vetores e Matrizes

Operações

Álgebra Linear

Determinantes

```
import numpy as np  
  
import numpy.linalg as la  
  
det_A = la.det( A )  
  
print(det_A)
```

Matiz Inversa

```
inv_A = la.inv( A )
```

```
print(inv_A)
```

```
print(A * inv_A)
```

```
print(A @ inv_A)
```


Sistemas Lineares

Resolvendo o sistema linear $Ax = b$

```
b = np.array( [5 , 6] )  
print(b)
```

```
x = la.solve( A, b )  
print(x)
```

Conferindo

```
print(A @ x - b)
```

Sistemas Lineares – 2

Criando uma matriz 7×7

```
A = np.random.rand(7, 7)  
print(A)
```

Criando um vetor com 7 elementos

```
b = np.random.rand(7, 1)  
print(b)
```

Resolvendo o sistema

```
x = la.solve( A, b )  
print(x)
```

Conferindo

```
print(A @ x - b)
```