

Curso básico de Python para estudantes de Física

Germán A. Racca

Universidade do Estado do Rio Grande do Norte
Faculdade de Ciências Exatas e Naturais
Departamento de Física
Mossoró - RN

08 de Junho de 2016

1 NumPy

- Broadcasting
- copy
- savetxt
- loadtxt

2 Matplotlib

- Partes da figura
- Marcadores
- Nomes dos eixos
- Intervalos dos eixos
- Estilos de linha
- Regiões com sombra
- Bordas dos gráficos
- Divisões dos eixos
- Etiquetas nos eixos
- Legendas
- Anotações

3 Material do curso

1 NumPy

- Broadcasting
- copy
- savetxt
- loadtxt

2 Matplotlib

- Partes da figura
- Marcadores
- Nomes dos eixos
- Intervalos dos eixos
- Estilos de linha
- Regiões com sombra
- Bordas dos gráficos
- Divisões dos eixos
- Etiquetas nos eixos
- Legendas
- Anotações

3 Material do curso

1 NumPy

- Broadcasting
- copy
- savetxt
- loadtxt

2 Matplotlib

- Partes da figura
- Marcadores
- Nomes dos eixos
- Intervalos dos eixos
- Estilos de linha
- Regiões com sombra
- Bordas dos gráficos
- Divisões dos eixos
- Etiquetas nos eixos
- Legendas
- Anotações

3 Material do curso

Broadcasting

Definição

Broadcasting permite realizar operações aritméticas entre *arrays* de formas diferentes.

Primeiro exemplo de *broadcasting*:

```
>>> A = np.array([ [11, 12, 13], [21, 22, 23], [31, 32, 33] ])
```

```
>>> B = np.array([1, 2, 3])
```

```
>>> print A * B
```

```
[[11 24 39]
```

```
 [21 44 69]
```

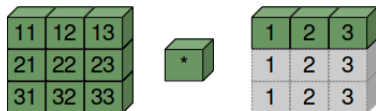
```
 [31 64 99]]
```

```
>>> print A + B
```

```
[[12 14 16]
```

```
 [22 24 26]
```

```
 [32 34 36]]
```



B foi usado como se ele fosse:

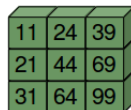
```
>>> B = np.array([[1, 2, 3],] * 3)
```

```
>>> print B
```

```
[[1 2 3]
```

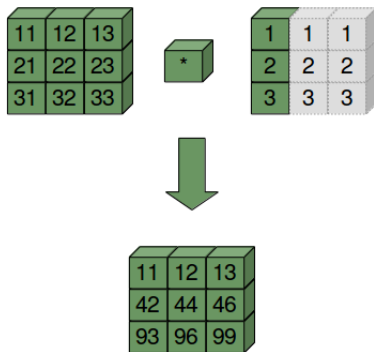
```
 [1 2 3]
```

```
 [1 2 3]]
```



Segundo exemplo de *broadcasting*:

```
>>> A = np.array([ [11, 12, 13], [21, 22, 23], [31, 32, 33] ])
>>> B = np.array([1, 2, 3])
>>> print B[:, np.newaxis]
[[1]
 [2]
 [3]]
>>> A * B[:, np.newaxis]
array([[11, 12, 13],
       [42, 44, 46],
       [93, 96, 99]])
```



B foi usado como se ele fosse:

```
>>> B = np.array([[1, 2, 3],] * 3).transpose()
>>> print B
[[1 1 1]
 [2 2 2]
 [3 3 3]]
```

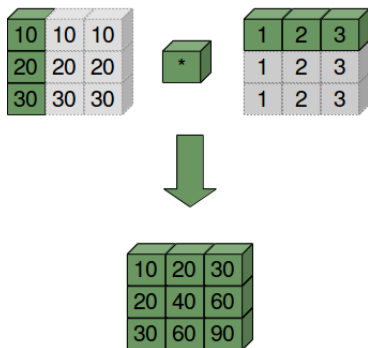
Broadcasting

Terceiro exemplo de *broadcasting*:

```
>>> A = np.array([10, 20, 30])
>>> B = np.array([1, 2, 3])
>>> print A[:, np.newaxis]
[[10]
 [20]
 [30]]
>>> A[:, np.newaxis] * B
array([[10, 20, 30],
       [20, 40, 60],
       [30, 60, 90]])
```

A e B foram usados como se eles fossem:

```
>>> A = np.array([[10, 20, 30],] * 3).transpose()
>>> print A
[[10 10 10]
 [20 20 20]
 [30 30 30]]
>>> B = np.array([[1, 2, 3],] * 3)
>>> print B
[[1 2 3]
 [1 2 3]
 [1 2 3]]
```



1 NumPy

- Broadcasting
- **copy**
- savetxt
- loadtxt

2 Matplotlib

- Partes da figura
- Marcadores
- Nomes dos eixos
- Intervalos dos eixos
- Estilos de linha
- Regiões com sombra
- Bordas dos gráficos
- Divisões dos eixos
- Etiquetas nos eixos
- Legendas
- Anotações

3 Material do curso

Definição

Retorna a cópia de um dado *array*
`copy(a, order='K')`

Exemplo de copy:

```
import numpy as np
>>> x = np.array([[42, 22, 12], [44, 53, 66]])
>>> y = x
>>> z = x.copy()
>>> x[0, 0] = 1001
>>> print x
[[1001  22  12]
 [ 44  53  66]]
>>> print y
[[1001  22  12]
 [ 44  53  66]]
>>> print z
[[42 22 12]
 [44 53 66]]
>>> y is x
True
>>> z is x
False
```

1 NumPy

- Broadcasting
- copy
- **savetxt**
- loadtxt

2 Matplotlib

- Partes da figura
- Marcadores
- Nomes dos eixos
- Intervalos dos eixos
- Estilos de linha
- Regiões com sombra
- Bordas dos gráficos
- Divisões dos eixos
- Etiquetas nos eixos
- Legendas
- Anotações

3 Material do curso

savetxt

Definição

Salva um *array* em um arquivo de texto

```
savetxt(fname, X, fmt='%.18e', delimiter=' ', newline='\n',  
header='', footer='', comments='# ')
```

Exemplo de savetxt:

```
>>> x = np.array([[1, 2, 3],  
...               [4, 5, 6],  
...               [7, 8, 9]])  
>>> np.savetxt("test1.txt", x)
```

```
$ cat test1.txt  
1.0000000000000000e+00 2.0000000000000000e+00 3.0000000000000000e+00  
4.0000000000000000e+00 5.0000000000000000e+00 6.0000000000000000e+00  
7.0000000000000000e+00 8.0000000000000000e+00 9.0000000000000000e+00
```

```
>>> np.savetxt("test2.txt", x, fmt="%.3f", delimiter=",")  
>>> np.savetxt("test3.txt", x, fmt="%04d", delimiter=" | ")
```

```
$ cat test2.txt  
1.000,2.000,3.000  
4.000,5.000,6.000  
7.000,8.000,9.000
```

```
$ cat test3.txt  
0001 | 0002 | 0003  
0004 | 0005 | 0006  
0007 | 0008 | 0009
```

1 NumPy

- Broadcasting
- copy
- savetxt
- **loadtxt**

2 Matplotlib

- Partes da figura
- Marcadores
- Nomes dos eixos
- Intervalos dos eixos
- Estilos de linha
- Regiões com sombra
- Bordas dos gráficos
- Divisões dos eixos
- Etiquetas nos eixos
- Legendas
- Anotações

3 Material do curso

loadtxt

Definição

Lê dados de um arquivo de texto

```
loadtxt(fname, dtype=<type 'float'>, comments='#', delimiter=None,
converters=None, skiprows=0, usecols=None, unpack=False, ndmin=0)
```

Exemplo de loadtxt:

```
>>> y = np.loadtxt("test1.txt")
>>> print y
[[ 1.  2.  3.]
 [ 4.  5.  6.]
 [ 7.  8.  9.]]
>>> y = np.loadtxt("test2.txt", delimiter=",")
>>> print y
[[ 1.  2.  3.]
 [ 4.  5.  6.]
 [ 7.  8.  9.]]
>>> y1, y2 = np.loadtxt("test1.txt", usecols=(0, 2), unpack=True)
>>> print y1
[ 1.  4.  7.]
>>> print y2
[ 3.  6.  9.]
```

1 NumPy

- Broadcasting
- copy
- savetxt
- loadtxt

2 Matplotlib

- Partes da figura
- Marcadores
- Nomes dos eixos
- Intervalos dos eixos
- Estilos de linha
- Regiões com sombra
- Bordas dos gráficos
- Divisões dos eixos
- Etiquetas nos eixos
- Legendas
- Anotações

3 Material do curso

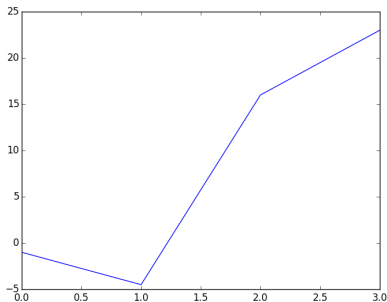
Matplotlib

- Módulo open source para Python
- Biblioteca para realizar gráficos 2D de *arrays*
- Escrita principalmente em Python e faz uso extensivo do NumPy
- Figuras com qualidade de publicação em diferentes formatos (PNG, PS, etc)

Exemplo simples de Matplotlib:

```
>>> import matplotlib.pyplot as plt
>>> plt.plot([-1, -4.5, 16, 23])
>>> plt.show()
```

```
Y = [-1, -4.5, 16, 23]
X = [0, 1, 2, 3]
```



1 NumPy

- Broadcasting
- copy
- savetxt
- loadtxt

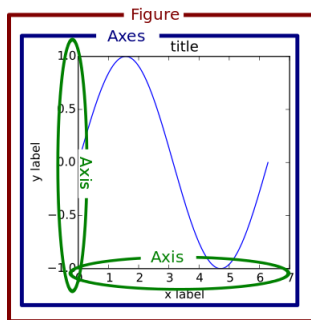
2 Matplotlib

- Partes da figura
- Marcadores
- Nomes dos eixos
- Intervalos dos eixos
- Estilos de linha
- Regiões com sombra
- Bordas dos gráficos
- Divisões dos eixos
- Etiquetas nos eixos
- Legendas
- Anotações

3 Material do curso

Partes da figura

- **Figure** ⇒ A figura completa (gráficos, títulos, legendas, etc)
- **Axes** ⇒ O gráfico (região da imagem com o espaço de dados)
- **Axis** ⇒ Os limites do gráfico (divisões dos eixos, etiquetas, etc)



<http://matplotlib.org/gallery.html>

1 NumPy

- Broadcasting
- copy
- savetxt
- loadtxt

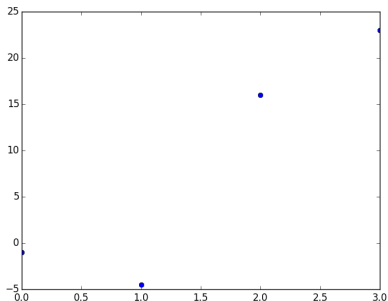
2 Matplotlib

- Partes da figura
- **Marcadores**
- Nomes dos eixos
- Intervalos dos eixos
- Estilos de linha
- Regiões com sombra
- Bordas dos gráficos
- Divisões dos eixos
- Etiquetas nos eixos
- Legendas
- Anotações

3 Material do curso

Marcadores

```
>>> plt.plot([-1, -4.5, 16, 23], "ob")
>>> plt.show()
```



character	color
'b'	blue
'g'	green
'r'	red
'c'	cyan
'm'	magenta
'y'	yellow
'k'	black
'w'	white

character	description
'-'	solid line style
'--'	dashed line style
'-.'	dash-dot line style
':'	dotted line style
'.'	point marker
'p'	pixel marker
'o'	circle marker
'v'	triangle_down marker
'^'	triangle_up marker
'<'	triangle_left marker
'>'	triangle_right marker
'1'	tri_down marker
'2'	tri_up marker
'3'	tri_left marker
'4'	tri_right marker
's'	square marker
'p'	pentagon marker
'*'	star marker
'h'	hexagon1 marker
'H'	hexagon2 marker
'+'	plus marker
'x'	x marker
'D'	diamond marker
'd'	thin_diamond marker
' '	vline marker
'_'	hline marker

1 NumPy

- Broadcasting
- copy
- savetxt
- loadtxt

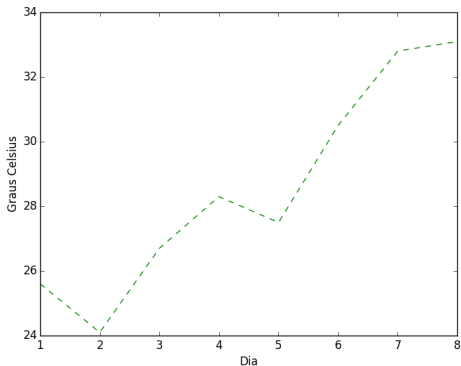
2 Matplotlib

- Partes da figura
- Marcadores
- **Nomes dos eixos**
- Intervalos dos eixos
- Estilos de linha
- Regiões com sombra
- Bordas dos gráficos
- Divisões dos eixos
- Etiquetas nos eixos
- Legendas
- Anotações

3 Material do curso

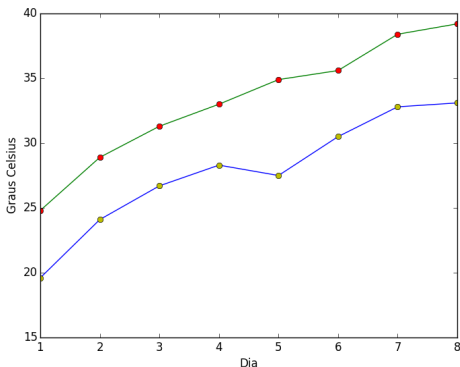
Nomes dos eixos

```
>>> import matplotlib.pyplot as plt
>>> dias = list(range(1, 9))
>>> temp = [25.6, 24.1, 26.7, 28.3, 27.5, 30.5, 32.8, 33.1]
>>> plt.plot(dias, temp, "g--")
>>> plt.xlabel('Dia')
>>> plt.ylabel('Graus Celsius')
>>> plt.show()
```



Nomes dos eixos

```
>>> dias = list(range(1, 9))
>>> temp_min = [19.6, 24.1, 26.7, 28.3, 27.5, 30.5, 32.8, 33.1]
>>> temp_max = [24.8, 28.9, 31.3, 33.0, 34.9, 35.6, 38.4, 39.2]
>>> plt.xlabel('Dia')
>>> plt.ylabel('Graus Celsius')
>>> plt.plot(dias, temp_min, dias, temp_min, "oy",
...          dias, temp_max, dias, temp_max, "or")
>>> plt.show()
```



1 NumPy

- Broadcasting
- copy
- savetxt
- loadtxt

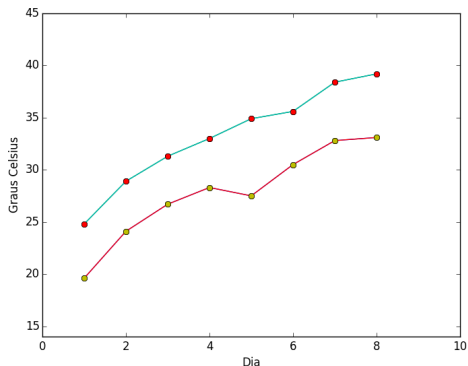
2 Matplotlib

- Partes da figura
- Marcadores
- Nomes dos eixos
- Intervalos dos eixos
- Estilos de linha
- Regiões com sombra
- Bordas dos gráficos
- Divisões dos eixos
- Etiquetas nos eixos
- Legendas
- Anotações

3 Material do curso

Intervalos dos eixos

```
>>> dias = list(range(1, 9))
>>> temp_min = [19.6, 24.1, 26.7, 28.3, 27.5, 30.5, 32.8, 33.1]
>>> temp_max = [24.8, 28.9, 31.3, 33.0, 34.9, 35.6, 38.4, 39.2]
>>> plt.xlabel('Dia')
>>> plt.ylabel('Graus Celsius')
>>> plt.plot(dias, temp_min, dias, temp_min, "oy",
...          dias, temp_max, dias, temp_max, "or")
>>> print plt.axis()
(1.0, 8.0, 15.0, 40.0)
>>> plt.axis([0, 10, 14, 45])
>>> plt.show()
```



1 NumPy

- Broadcasting
- copy
- savetxt
- loadtxt

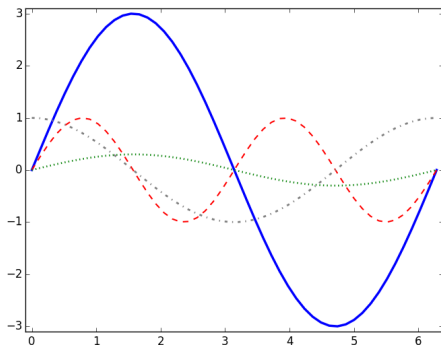
2 Matplotlib

- Partes da figura
- Marcadores
- Nomes dos eixos
- Intervalos dos eixos
- **Estilos de linha**
- Regiões com sombra
- Bordas dos gráficos
- Divisões dos eixos
- Etiquetas nos eixos
- Legendas
- Anotações

3 Material do curso

Estilos de linha

```
>>> import numpy as np
>>> X = np.linspace(0, 2*np.pi, 50, endpoint=True)
>>> F1, F2 = 3*np.sin(X), np.sin(2*X)
>>> F3, F4 = 0.3*np.sin(X), np.cos(X)
>>> plt.axis([-0.1, 2*np.pi + 0.1, -3.1, 3.1])
>>> plt.plot(X, F1, color="blue", linewidth=2.5, linestyle="-")
>>> plt.plot(X, F2, color="red", linewidth=1.5, linestyle="--")
>>> plt.plot(X, F3, color="green", linewidth=2, linestyle=":")
>>> plt.plot(X, F4, color="grey", linewidth=2, linestyle="-.")
>>> plt.show()
```



1 NumPy

- Broadcasting
- copy
- savetxt
- loadtxt

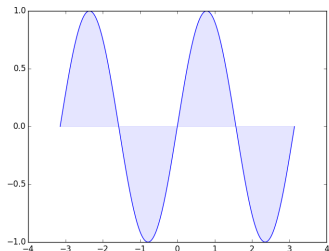
2 Matplotlib

- Partes da figura
- Marcadores
- Nomes dos eixos
- Intervalos dos eixos
- Estilos de linha
- **Regiões com sombra**
- Bordas dos gráficos
- Divisões dos eixos
- Etiquetas nos eixos
- Legendas
- Anotações

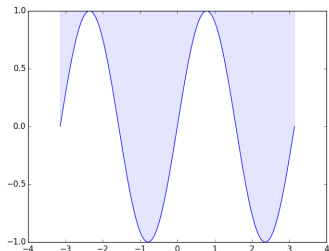
3 Material do curso

Regiões com sombra

```
n = 256
X = np.linspace(-np.pi, np.pi, n, endpoint=True)
Y = np.sin(2*X)
plt.plot(X, Y, color='blue')
plt.fill_between(X, 0, Y, color='blue', alpha=0.1)
plt.show()
```



```
n = 256
X = np.linspace(-np.pi, np.pi, n, endpoint=True)
Y = np.sin(2*X)
plt.plot(X, Y, color='blue')
plt.fill_between(X, Y, 1, color='blue', alpha=0.1)
plt.show()
```



1 NumPy

- Broadcasting
- copy
- savetxt
- loadtxt

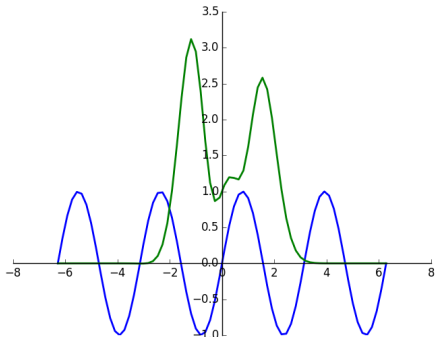
2 Matplotlib

- Partes da figura
- Marcadores
- Nomes dos eixos
- Intervalos dos eixos
- Estilos de linha
- Regiões com sombra
- **Bordas dos gráficos**
- Divisões dos eixos
- Etiquetas nos eixos
- Legendas
- Anotações

3 Material do curso

Bordas dos gráficos

```
>>> X = np.linspace(-2*np.pi, 2*np.pi, 70, endpoint=True)
>>> F1 = np.sin(2*X)
>>> F2 = (2*X**5 + 4*X**4 - 4.8*X**3 + 1.2*X**2 + X + 1)*np.exp(-X**2)
>>> ax = plt.gca()
>>> ax.spines['top'].set_color('none')
>>> ax.spines['right'].set_color('none')
>>> ax.xaxis.set_ticks_position('bottom')
>>> ax.spines['bottom'].set_position('zero')
>>> ax.yaxis.set_ticks_position('left')
>>> ax.spines['left'].set_position('zero')
>>> plt.plot(X, F1)
>>> plt.plot(X, F2)
>>> plt.show()
```



1 NumPy

- Broadcasting
- copy
- savetxt
- loadtxt

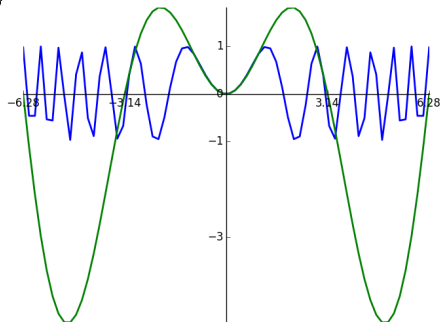
2 Matplotlib

- Partes da figura
- Marcadores
- Nomes dos eixos
- Intervalos dos eixos
- Estilos de linha
- Regiões com sombra
- Bordas dos gráficos
- **Divisões dos eixos**
- Etiquetas nos eixos
- Legendas
- Anotações

3 Material do curso

Divisões dos eixos

```
>>> X = np.linspace(-2*np.pi, 2*np.pi, 70, endpoint=True)
>>> F1 = np.sin(X**2)
>>> F2 = X*np.sin(X)
>>> ax = plt.gca()
>>> ax.spines['top'].set_color('none')
>>> ax.spines['right'].set_color('none')
>>> ax.xaxis.set_ticks_position('bottom')
>>> ax.spines['bottom'].set_position('zero')
>>> ax.yaxis.set_ticks_position('left')
>>> ax.spines['left'].set_position('zero')
>>> plt.xticks([-6.28, -3.14, 3.14, 6.28])
>>> plt.yticks([-3, -1, 0, 1, 3])
>>> plt.plot(X, F1, lw=2)
>>> plt.plot(X, F2, lw=2)
>>> plt.show()
```



1 NumPy

- Broadcasting
- copy
- savetxt
- loadtxt

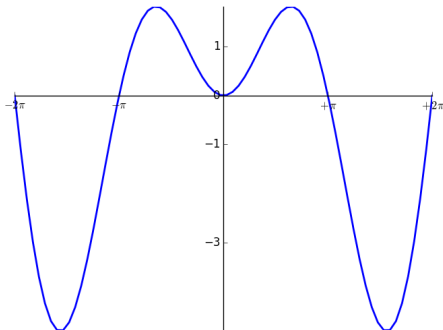
2 Matplotlib

- Partes da figura
- Marcadores
- Nomes dos eixos
- Intervalos dos eixos
- Estilos de linha
- Regiões com sombra
- Bordas dos gráficos
- Divisões dos eixos
- **Etiquetas nos eixos**
- Legendas
- Anotações

3 Material do curso

Etiquetas nos eixos

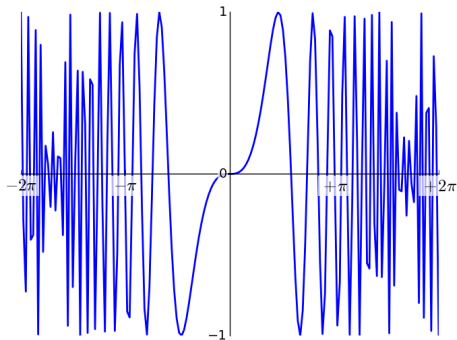
```
>>> X = np.linspace(-2*np.pi, 2*np.pi, 70, endpoint=True)
>>> F1 = X*np.sin(X)
>>> ax = plt.gca()
>>> ax.spines['top'].set_color('none')
>>> ax.spines['right'].set_color('none')
>>> ax.xaxis.set_ticks_position('bottom')
>>> ax.spines['bottom'].set_position('zero')
>>> ax.yaxis.set_ticks_position('left')
>>> ax.spines['left'].set_position('zero')
>>> plt.xticks([-6.28, -3.14, 3.14, 6.28],
...           [r'$-2\pi$', r'$-\pi$', r'$+\pi$', r'$+2\pi$'])
>>> plt.yticks([-3, -1, 0, 1, 3])
>>> plt.plot(X, F1)
>>> plt.show()
```



Etiquetas nos eixos

```
>>> X = np.linspace(-2*np.pi, 2*np.pi, 170, endpoint=True)
>>> F1 = np.sin(X**3/2)
.....
>>> for xtick in ax.get_xticklabels():
...     xtick.set_fontsize(18)
...     xtick.set_bbox(dict(facecolor='white', edgecolor='None', alpha=0.7))
>>> for ytick in ax.get_yticklabels():
...     ytick.set_fontsize(14)
...     ytick.set_bbox(dict(facecolor='white', edgecolor='None', alpha=0.7))
>>> plt.show()
```

```
>>> for xtick in ax.get_xticklabels():
...     print xtick
Text(-6.28,0,u'$$$-2\\pi$')
Text(-3.14,0,u'$$$-\\pi$')
Text(3.14,0,u'$$$+\\pi$')
Text(6.28,0,u'$$$+2\\pi$')
```



1 NumPy

- Broadcasting
- copy
- savetxt
- loadtxt

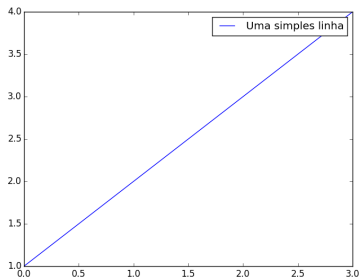
2 Matplotlib

- Partes da figura
- Marcadores
- Nomes dos eixos
- Intervalos dos eixos
- Estilos de linha
- Regiões com sombra
- Bordas dos gráficos
- Divisões dos eixos
- Etiquetas nos eixos
- **Legendas**
- Anotações

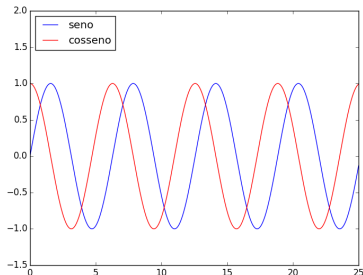
3 Material do curso

Legendas

```
>>> import matplotlib.pyplot as plt
>>> plt.plot([1, 2, 3, 4])
>>> plt.legend(['Uma simples linha'])
>>> plt.show()
```

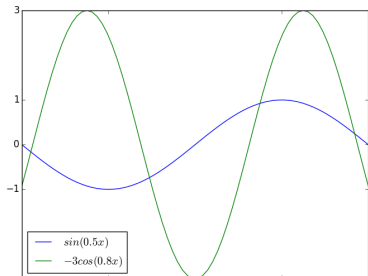


```
>>> import numpy as np
>>> x = np.linspace(0, 25, 1000)
>>> y1 = np.sin(x)
>>> y2 = np.cos(x)
>>> plt.plot(x, y1, '-b', label='seno')
>>> plt.plot(x, y2, '-r', label='cosseno')
>>> plt.legend(loc='upper left')
>>> plt.ylim(-1.5, 2.0)
>>> plt.show()
```

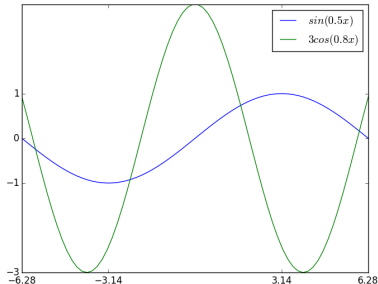


Legendas

```
>>> X = np.linspace(-2*np.pi, 2*np.pi, 70)
>>> F1 = np.sin(0.5*X)
>>> F2 = -3*np.cos(0.8*X)
>>> plt.xticks([-6.28, -3.14, 3.14, 6.28])
>>> plt.yticks([-3, -1, 0, 1, 3])
>>> plt.plot(X, F1, label='$sin(0.5x)$')
>>> plt.plot(X, F2, label='$-3cos(0.8x)$')
>>> plt.legend(loc='best')
>>> plt.show()
```



```
>>> X = np.linspace(-2*np.pi, 2*np.pi, 70)
>>> F1 = np.sin(0.5*X)
>>> F2 = 3*np.cos(0.8*X)
>>> plt.xticks([-6.28, -3.14, 3.14, 6.28])
>>> plt.yticks([-3, -1, 0, 1, 3])
>>> plt.plot(X, F1, label='$sin(0.5x)$')
>>> plt.plot(X, F2, label='$-3cos(0.8x)$')
>>> plt.legend(loc='best')
>>> plt.show()
```



1 NumPy

- Broadcasting
- copy
- savetxt
- loadtxt

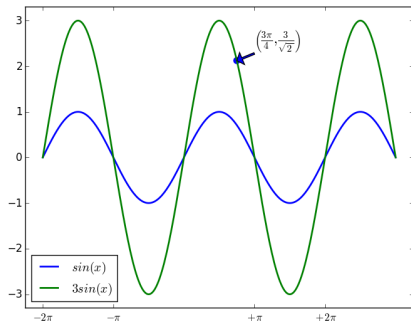
2 Matplotlib

- Partes da figura
- Marcadores
- Nomes dos eixos
- Intervalos dos eixos
- Estilos de linha
- Regiões com sombra
- Bordas dos gráficos
- Divisões dos eixos
- Etiquetas nos eixos
- Legendas
- **Anotações**

3 Material do curso

Anotações

```
>>> X = np.linspace(-2*np.pi, 3*np.pi, 300, endpoint=True)
>>> F1 = np.sin(X)
>>> F2 = 3*np.sin(X)
>>> plt.xticks([-6.28, -3.14, 3.14, 6.28],
...            [r'$-2\pi$', r'$-\pi$', r'$+\pi$', r'$+2\pi$'])
>>> plt.yticks([-3, -2, -1, 0, 1, 2, 3])
>>> x = 3*np.pi / 4
>>> plt.scatter(x, 3*np.sin(x), 50, color='blue')
>>> arrow = dict(facecolor='blue', headwidth=15, frac=0.3, width=2)
>>> plt.annotate(r'$\left(\frac{3\pi}{4}, \frac{3}{\sqrt{2}}\right)$',
...             xy=(x, 3*np.sin(x)),
...             xycoords='data',
...             xytext=(+20, +20),
...             textcoords='offset points',
...             fontsize=16,
...             arrowprops=arrow)
>>> plt.plot(X, F1, lw=2, label='$\sin(x)$')
>>> plt.plot(X, F2, lw=2, label='$3\sin(x)$')
>>> plt.legend(loc='lower left')
>>> plt.show()
```



- 1 NumPy
 - Broadcasting
 - copy
 - savetxt
 - loadtxt

- 2 Matplotlib
 - Partes da figura
 - Marcadores
 - Nomes dos eixos
 - Intervalos dos eixos
 - Estilos de linha
 - Regiões com sombra
 - Bordas dos gráficos
 - Divisões dos eixos
 - Etiquetas nos eixos
 - Legendas
 - Anotações

- 3 Material do curso

Aulas do Lázaro Camargo (INPE):

<https://gle35v.s.cld.pt>

material-python-02.tar.bz2

matplotlib_aula_01_fundamentos.pdf

matplotlib_aula_02_estilos_tipos_plotagem.pdf

python_aula_06_arquivos.pdf