Les courbes en Python
MatPlotLib

K. Boudjelaba
k.boudjelaba@carnus.fr

13 mars 2023

Table des matières

— Les détails de la librairie MatPlotLib sont disponibles ici
— Ou exécuter le code suivant dans Jupyter Notebook :

```
import matplotlib.pyplot as plt
help(plt)
```

```
Help on module matplotlib.pyplot in matplotlib:

NAME
    matplotlib.pyplot

DESCRIPTION
    `matplotlib.pyplot` is a state-based interface to matplotlib. It provides
    a MATLAB-like way of plotting.

    pyplot is mainly intended for interactive plots and simple cases of
    programmatic plot generation::

        import numpy as np
        import matplotlib.pyplot as plt

        x = np.arange(0, 5, 0.1)
        y = np.sin(x)
        plt.plot(x, y)

    The object-oriented API is recommended for more complex plots.
    ...
```

1 Plot

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

t = np.arange(0, 5+.01, .01)
y = np.exp(-t)

plt.figure()
plt.plot(t, y)
plt.xlabel('Abscisses')
plt.ylabel('Ordonnées')
plt.title('Figure')
plt.grid()
plt.show()
```

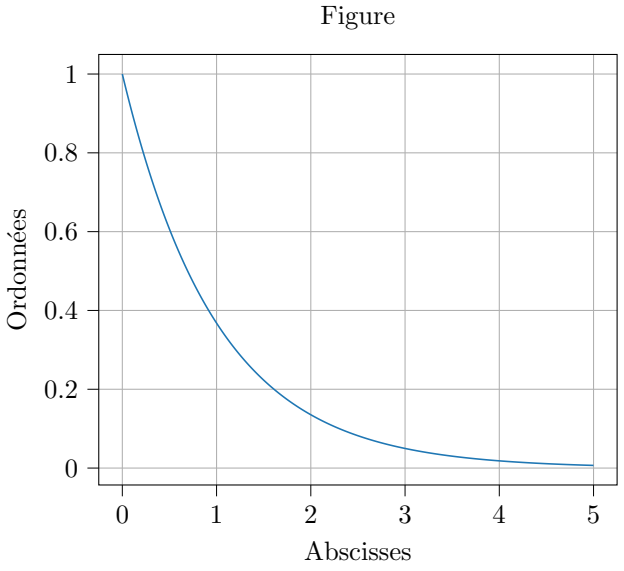


FIGURE 1 – Courbe 1

```
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline

t = np.arange(0, 5+.01, .01)
s = np.exp(-t)

fig, ax = plt.subplots()
ax.plot(t, s)

ax.set(xlabel='temps [s]', ylabel='Tension [V]',
       title='Signal s(t)')
ax.grid()

fig.savefig("Signal_S.png") # Pour sauvegarder la figure

plt.show()
```

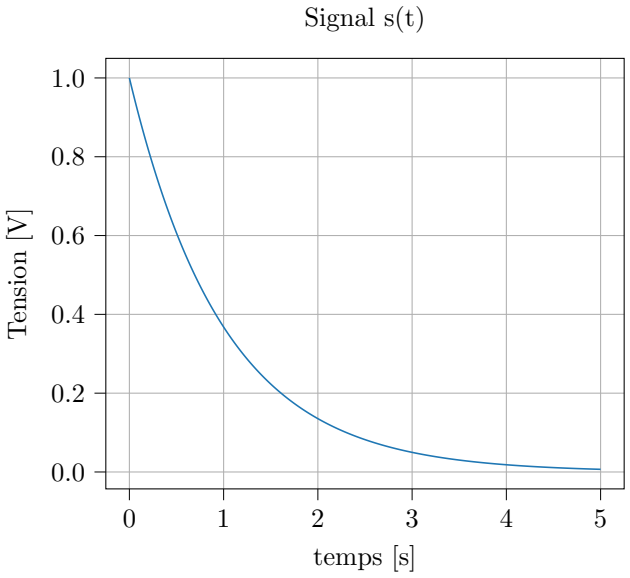


FIGURE 2 – Courbe 2

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

x = np.arange(0,4*np.pi+0.01,0.01)
y = 2.5 * np.sin(x)
z = 1.5 * np.cos(x)


fig = plt.figure(figsize=(8,4)) #figsize pour la taille de la figure

plt.plot(x, y, linewidth=2.0, label=r'$\sin(\omega t)$')
plt.plot(x, z, 'r--', label=r'$\cos(\omega t)$')

plt.xlabel(r'$x$')
plt.ylabel('Amplitude', fontsize=14, color='magenta')
plt.title('Titre en couleur', color='#f39c12')
plt.axis([x.min()-1, x.max()+1, -3, 3.5])
plt.legend(loc='lower left')
plt.grid()
plt.show()
```

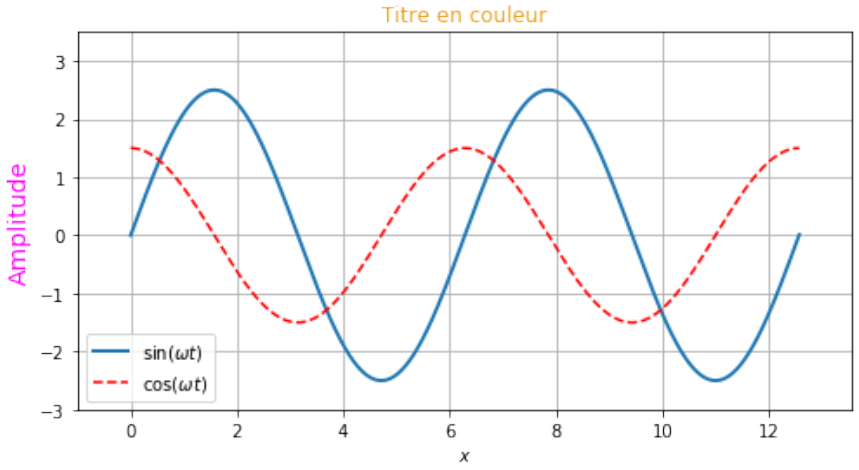


FIGURE 3 – Courbe 3

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

t = np.arange(0, 3+.1, .1)
y = np.exp(-t)

plt.figure()
plt.plot(t, y, 'o', color='green')
plt.xlabel('Abscisses')
plt.ylabel('Ordonnées')
plt.title('Figure')
plt.grid()
plt.show()
```

Figure

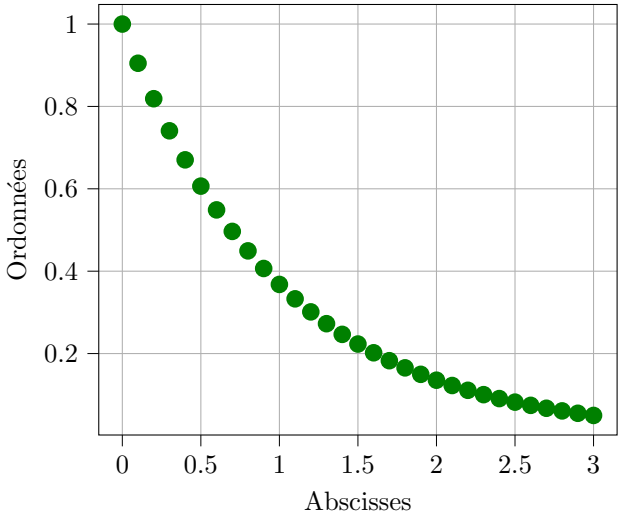


FIGURE 4 – Courbe 4

2 SubPlot

```
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt

t = np.arange(0, 1+.01, .01)
y = np.sin(2*np.pi*t)

fig = plt.figure(figsize=(8,6))

plt.subplot(221)
plt.plot(t,y)
plt.text(0.5, 0.5, "subplot(2,2,1)", ha='center', va='center', size='large')

plt.subplot(222)
plt.plot(t,y)
plt.text(0.5, 0.5, "subplot(2,2,2)", ha='center', va='center', size='large')

plt.subplot(223)
plt.plot(t,y)
plt.text(0.5, 0.5, "subplot(2,2,3)", ha='center', va='center', size='large')

plt.subplot(224)
plt.plot(t,y)
plt.text(0.5, 0.5, "subplot(2,2,4)", ha='center', va='center', size='large')

plt.show()
```

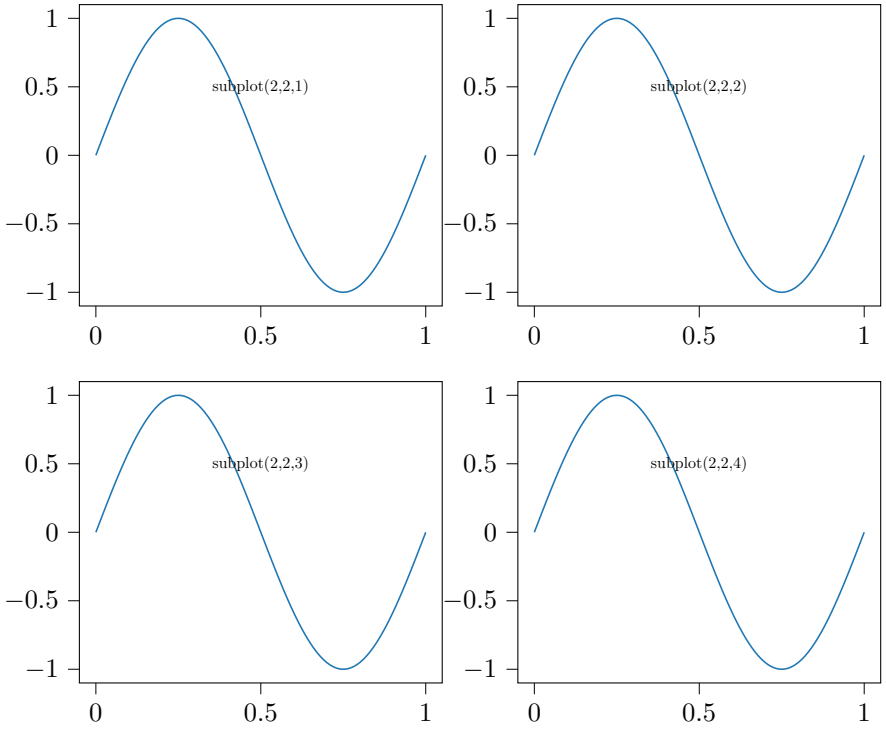


FIGURE 5 – Courbe 5

```
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt

t = np.arange(0, 1+.01, .01)
y = np.sin(2*np.pi*t)

fig = plt.figure(figsize=(8,6))

plt.subplot(221)
plt.plot(t,y)
plt.text(0.5, 0.5, "subplot(2,2,1)", ha='center', va='center', size='large')

plt.subplot(222)
plt.plot(t,y)
plt.text(0.5, 0.5, "subplot(2,2,2)", ha='center', va='center', size='large')

plt.subplot(234)
plt.plot(t,y)
plt.text(0.5, 0.5, "subplot(2,3,4)", ha='center', va='center', size='large')

plt.subplot(235)
plt.plot(t,y)
plt.text(0.5, 0.5, "subplot(2,3,5)", ha='center', va='center', size='large')

plt.subplot(236)
plt.plot(t,y)
plt.text(0.5, 0.5, "subplot(2,3,6)", ha='center', va='center', size='large')

plt.show()
```

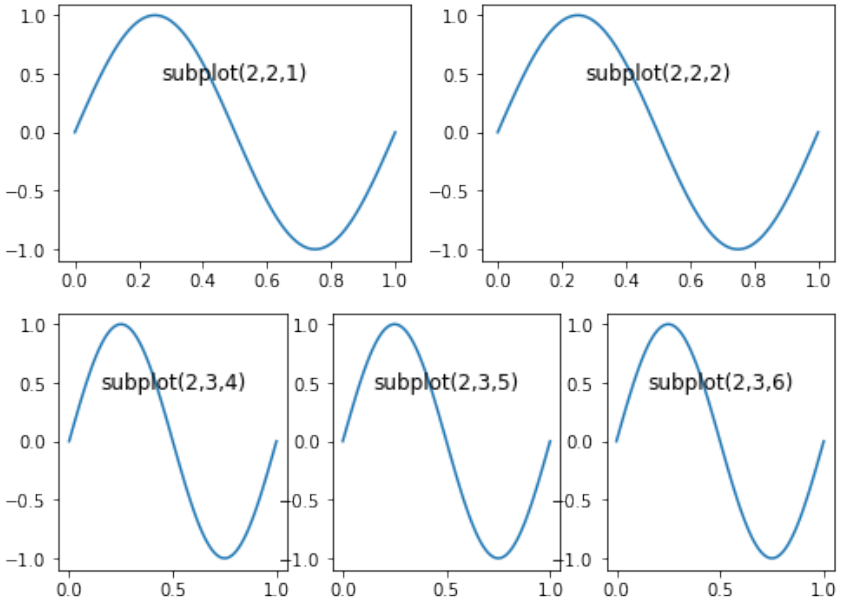


FIGURE 6 – Courbe 6

3 GridSpec

```
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.gridspec import GridSpec

t = np.arange(0, 1+.01, .01)
y = np.sin(2*np.pi*t)

fig = plt.figure(figsize=(8,6))
grid = GridSpec(3, 4)

plt.subplot(grid[0, :-1], xticks=[], yticks=[])
plt.plot(t,y)
plt.text(0.5, 0.5, "grid[0, :-1]", ha='center', va='center', size='large')

plt.subplot(grid[:-1, -1], xticks=[], yticks=[])
plt.plot(t,y)
plt.text(0.5, 0.5, "grid[:-1, -1]", ha='center', va='center', size='large')

plt.subplot(grid[1, 0], xticks=[], yticks=[])
plt.plot(t,y)
plt.text(0.5, 0.5, "grid[1, 0]", ha='center', va='center', size='large')

plt.subplot(grid[1, 1:3], xticks=[], yticks=[])
plt.plot(t,y)
plt.text(0.5, 0.5, "grid[1, 1:3]", ha='center', va='center', size='large')

plt.subplot(grid[-1, :-1])
plt.plot(t,y)
plt.text(0.5, 0.5, "grid[-1, :-1]", ha='center', va='center', size='large')
plt.grid(True)

plt.subplot(grid[-1, -1], xticks=[], yticks=[])
plt.plot(t,y)
plt.text(0.5, 0.5, "grid[-1, -1]", ha='center', va='center', size='large')

plt.show()
```

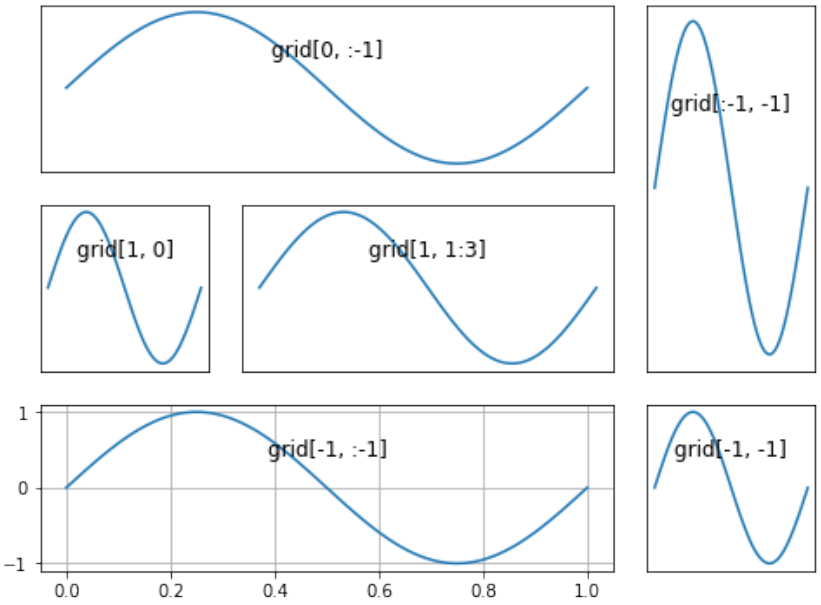


FIGURE 7 – Courbe 7

```
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.gridspec import GridSpec

t = np.arange(0, 1+.01, .01)
y = 0.5+np.sin(2*np.pi*t)

fig = plt.figure()
grid = GridSpec(3, 4)

ax1 = fig.add_subplot(grid[0, :-1], xticks=[], yticks=[])
ax1.plot(t,y)
ax1.text(0.5, 0.5, "grid[0, :-1]", ha='center', va='center', size='large')

ax2 = fig.add_subplot(grid[:-1, -1], xticks=[], yticks=[])
ax2.plot(t,y)
ax2.text(0.5, 0.5, "grid[:-1, -1]", ha='center', va='center', size='large')

ax3 = fig.add_subplot(grid[1, 0], xticks=[], yticks=[])
ax3.plot(t,y)
ax3.text(0.5, 0.5, "grid[1, 0]", ha='center', va='center', size='large')

ax4 = fig.add_subplot(grid[1, 1:3], xticks=[], yticks=[])
ax4.plot(t,y)
ax4.text(0.5, 0.5, "grid[1, 1:3]", ha='center', va='center', size='large')

ax5 = fig.add_subplot(grid[-1, :-1], xticks=[], yticks=[])
ax5.plot(t,y)
ax5.text(0.5, 0.5, "grid[-1, :-1]", ha='center', va='center', size='large')

ax6 = fig.add_subplot(grid[-1, -1], xticks=[], yticks=[])
ax6.plot(t,y)
ax6.text(0.5, 0.5, "grid[-1, -1]", ha='center', va='center', size='large')

plt.show()
```

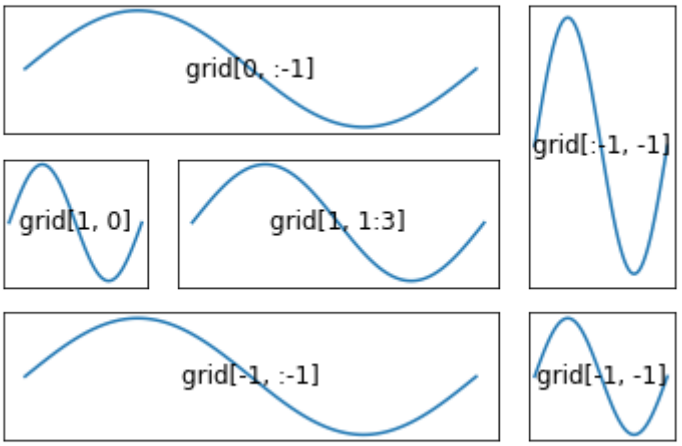


FIGURE 8 – Courbe 8

4 Multiple Plot

```
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
t = np.linspace(0, 2*np.pi)
fig, ax = plt.subplots(2, 3, sharex='col', sharey='row')
ax[1,1].plot(t,np.cos(t))
ax[1,1].grid()
plt.show()
```

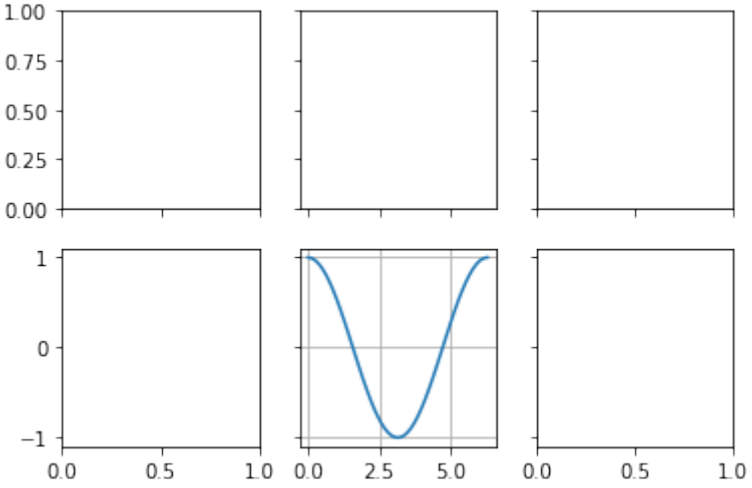


FIGURE 9 – Courbe 9

```
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt

def f(t):
    return np.sin(20*np.pi*t) * (1 + 0.75*np.cos(2*np.pi*t))

t = np.arange(0.0, np.pi, 0.01)

ax1 = plt.subplot(211)
ax1.margins(0.05)
ax1.plot(t, f(t))
ax1.set_title('Tracé normal')

ax2 = plt.subplot(223)
ax2.margins(.5, .5)
ax2.plot(t, f(t))
ax2.set_title('Tracé reculé')

ax3 = plt.subplot(224)
ax3.margins(x = -.1, y = -.3)
ax3.plot(t, f(t))
ax3.set_title('Tracé avancé')

plt.tight_layout()
plt.show()
```

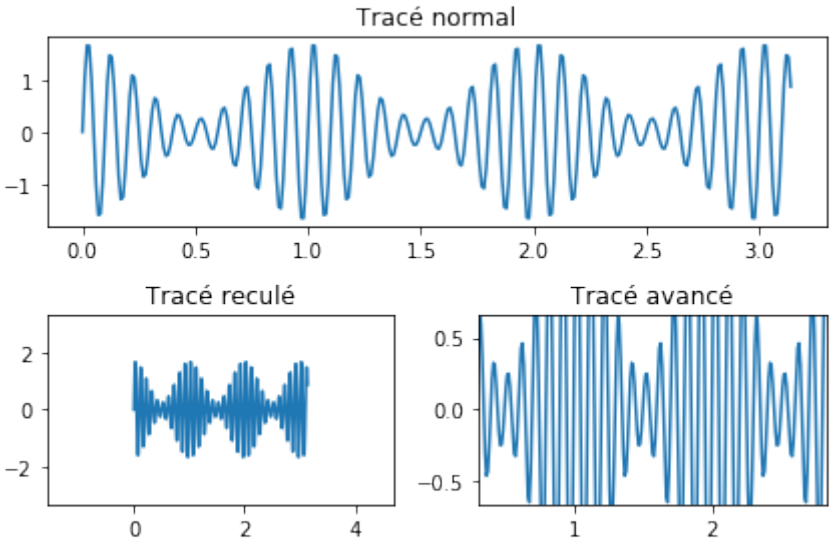


FIGURE 10 – Courbe 10

```
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline

ax = plt.subplot(111)
t = np.arange(.0, 2*np.pi+.01, .01)
for n in [1, 2, 3]:
    plt.plot(t, np.sin(n*t), label=r'$s_%d=sin(%d*t)$'%(n,n))
ax.set(ylim=(-1.6,1.1))
leg = plt.legend(loc='best', ncol=2, mode="expand", shadow=True, fancybox=True)
leg.get_frame().set_alpha(0.5)

plt.show()
```

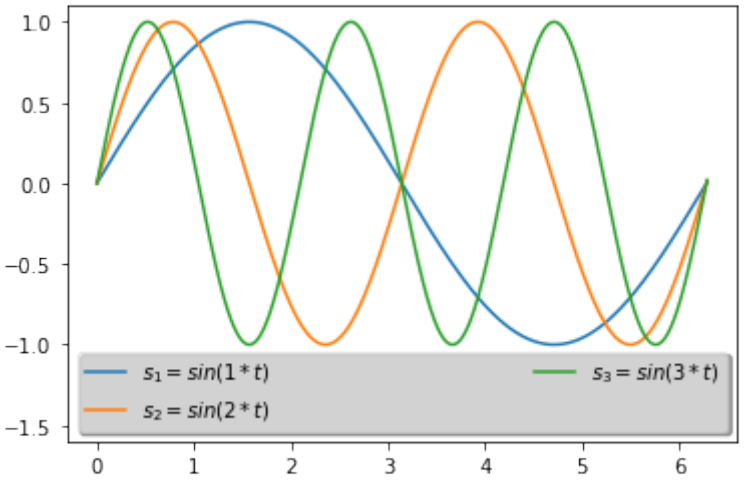


FIGURE 11 – Courbe 11

```
import matplotlib.pyplot as plt
import numpy as np
from matplotlib import colors
from matplotlib.ticker import PercentFormatter
%matplotlib inline

Nbp = 100
nb = 10
x = np.random.randn(Nbp)

fig, axs = plt.subplots(1, 2, tight_layout=True, figsize=(8,4))

axs[0].hist(x, bins=nb)
N, bins, patches = axs[1].hist(x, bins=nb)
fracs = N / N.max()
norm = colors.Normalize(fracs.min(), fracs.max())
for frac, patch in zip(fracs, patches):
    color = plt.cm.viridis(norm(frac))
    patch.set_facecolor(color)

axs[1].hist(x, bins=nb, density=True)
axs[1].yaxis.set_major_formatter(PercentFormatter(xmax=1))

plt.show()
```

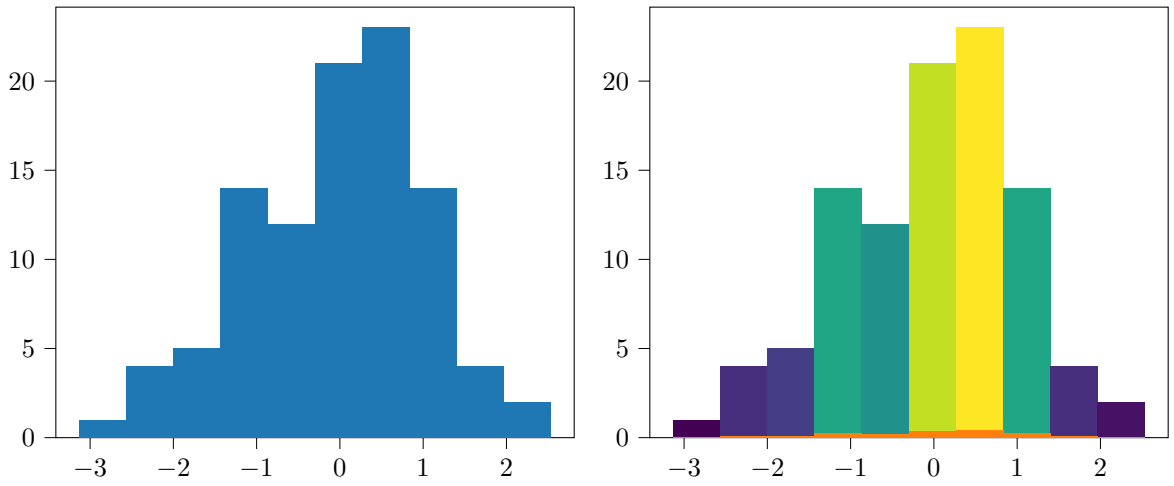


FIGURE 12 – Courbe 12

```
%matplotlib inline
import matplotlib.pyplot as plt
import matplotlib.transforms as mtransforms
import numpy as np

xs = np.arange(5)
ys = 2*xs

fig = plt.figure(figsize=(5, 10))
ax = plt.subplot(2, 1, 1)

trans_offset = mtransforms.offset_copy(ax.transData, fig=fig,
                                       x=0.05, y=0.10, units='inches')

for x, y in zip(xs, ys):
    plt.plot(x, y, 'ro')
    plt.text(x, y, '(%d, %d)' % (int(x), int(y)), transform=trans_offset)

plt.show()
```

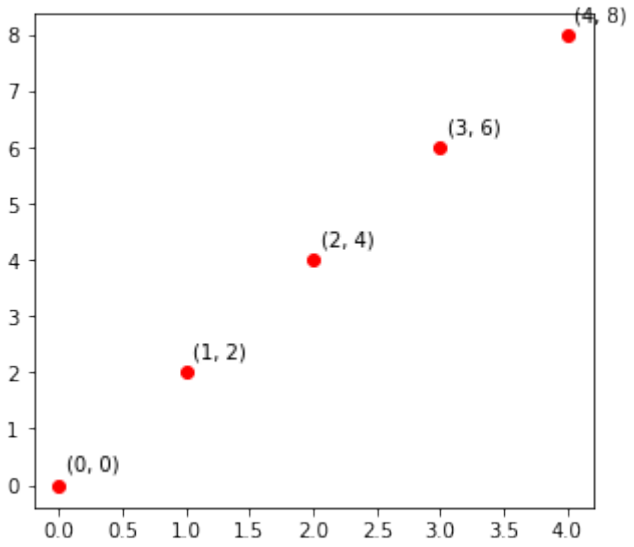


FIGURE 13 – Courbe 13

```
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt

t = np.arange(0, 5.0+.01, .01)
data1 = np.sin(2 * np.pi * t)
data2 = 0.8*np.exp(-t)*np.cos(2 * np.pi * t)

fig, ax1 = plt.subplots(figsize=(8,5))

color = 'tab:red'
ax1.set_xlabel('temps (s)')
ax1.set_ylabel(r'$sin(\omega t)$', color=color)
ax1.plot(t, data1, color=color)
ax1.tick_params(axis='y', labelcolor=color)

ax2 = ax1.twinx()

color = 'tab:blue'
ax2.set_ylabel(r'$a\times e^{-t}\times cos(\omega t)$', color=color)
ax2.plot(t, data2, color=color)
ax2.tick_params(axis='y', labelcolor=color)

fig.tight_layout()
plt.show()
```

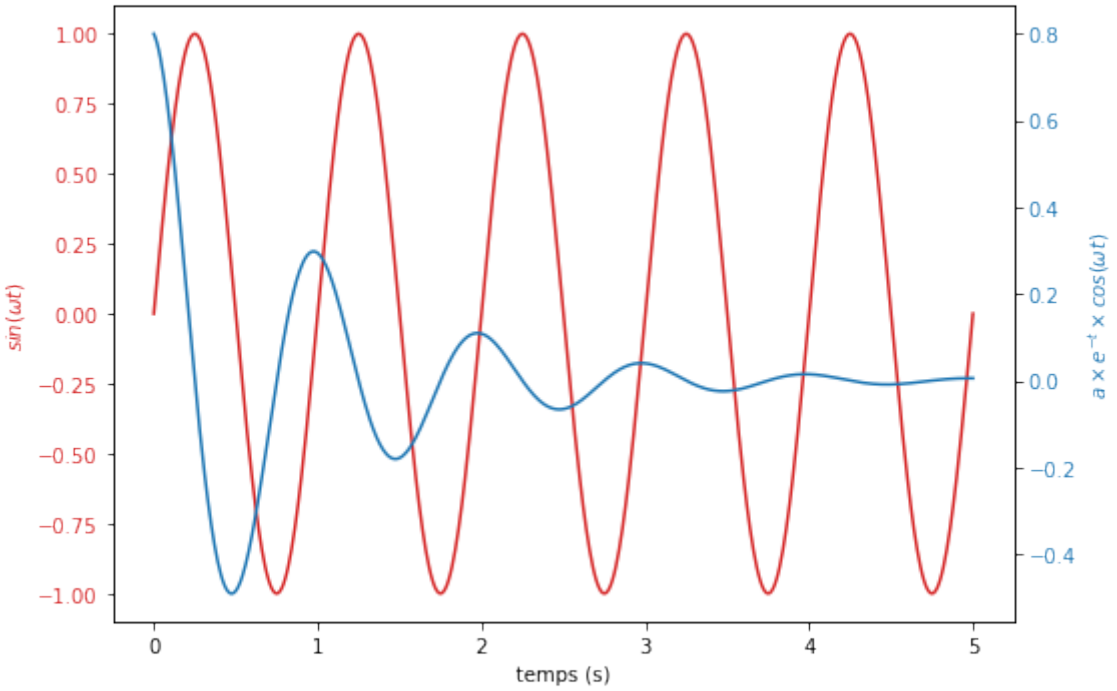


FIGURE 14 – Courbe 14

```
fig, ax = plt.subplots()

ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')

ax.xaxis.set_ticks_position('bottom')
ax.spines['bottom'].set_position(('data',0)) # Régler la position de l'axe x sur x=0

ax.yaxis.set_ticks_position('left')
ax.spines['left'].set_position(('data',0))   # Régler la position de l'axe y sur y=0

xx = np.linspace(-0.75, 1., 100)
ax.plot(xx, xx**3);

plt.show()
```

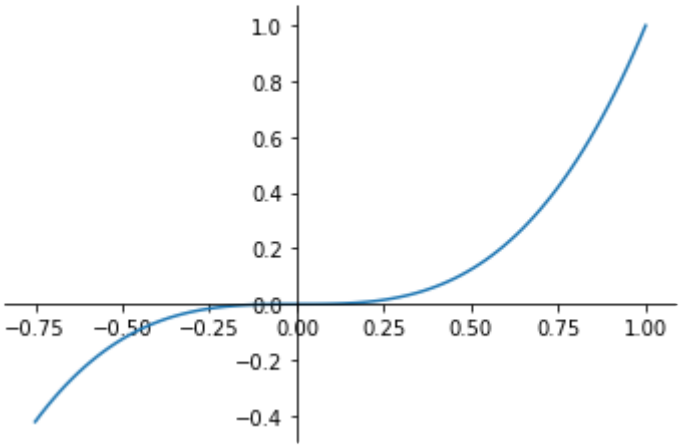


FIGURE 15 – Courbe 15

5 Stem

```
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt

n = np.arange(0,2*np.pi,0.2)
x = np.sin(n)

plt.figure(figsize=(8,5))
plt.stem(n,x,'r','rs',use_line_collection=True)
plt.axis([0,n.max(),-1.15,1.15])
plt.grid()
plt.show()
```

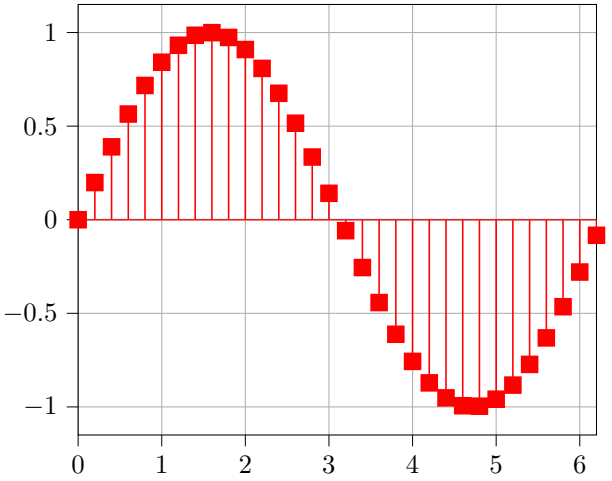


FIGURE 16 – Courbe 16

```
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt

n = np.arange(0,2*np.pi+0.2,0.2)
x = np.sin(n)

plt.figure(1)
markerline, stemlines, baseline = plt.stem(n,x, '--',use_line_collection=True)
plt.setp(stemlines, 'color', 'b', 'linewidth', 2)
#plt.setp(baseline, 'color', 'b', 'linewidth', 0.5)
plt.setp(baseline, visible=False)
plt.ylim([-1.2,1.2])
plt.xlim([n.min(),n.max()])
plt.xlabel('$n$')
plt.ylabel('$sin[n]$')
plt.title('$sin[n]$ avec stem')
plt.grid(True)

plt.show()
```

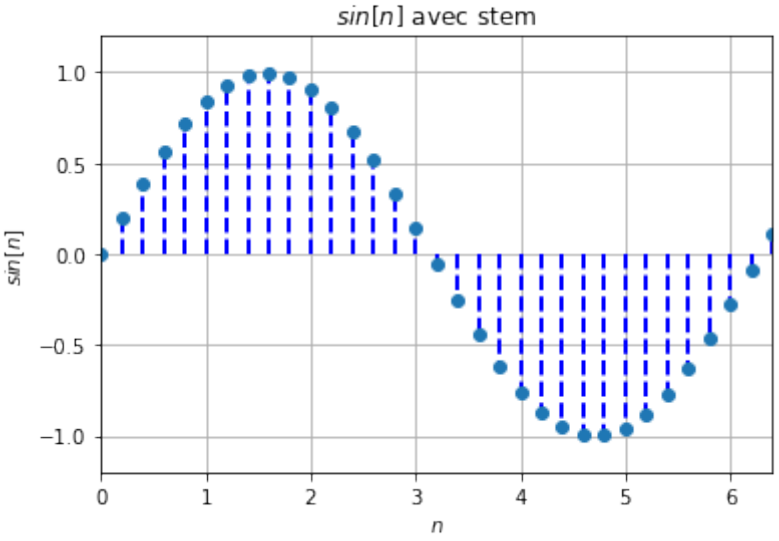


FIGURE 17 – Courbe 17

6 Scatter

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

t = np.arange(0, 3+.1, .1)
y = np.exp(-t)

plt.figure()
plt.scatter(t, y)
plt.xlabel('Abscisses')
plt.ylabel('Ordonnées')
plt.title('Figure')
plt.grid()
plt.show()
```

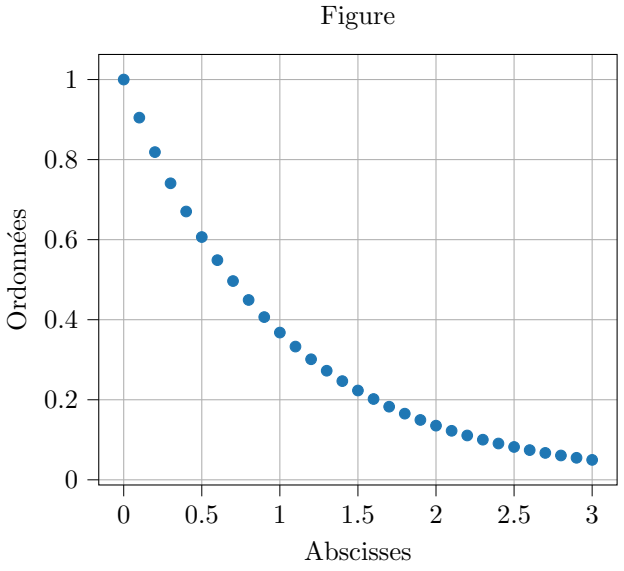


FIGURE 18 – Courbe 18

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

t = np.arange(0, 3+.1, .1)
y = np.exp(-t)

plt.figure()
plt.scatter(t, y, c=y, cmap='Spectral')
plt.colorbar()
plt.xlabel('Abscisses')
plt.ylabel('Ordonnées')
plt.title('Figure')
plt.grid()
plt.show()
```

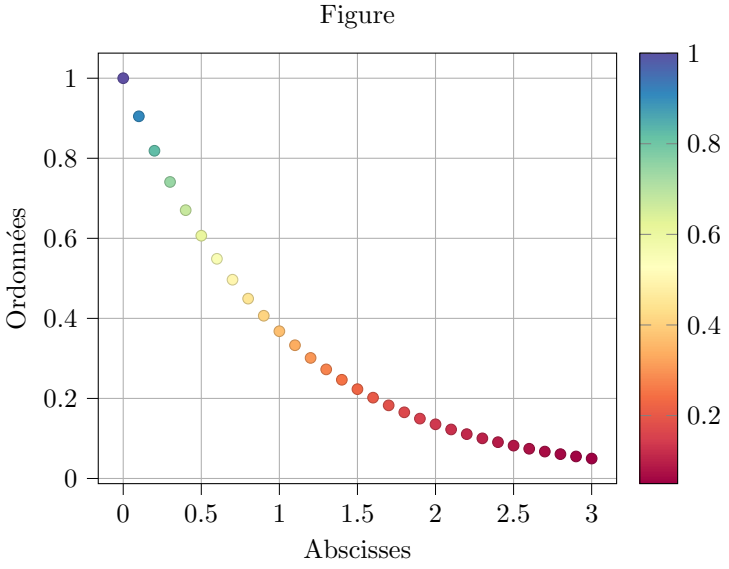


FIGURE 19 – Courbe 19

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

t = np.arange(0, 3+.1, .1)
y = np.exp(-t)
z = np.sin(t)

plt.figure()
plt.scatter(t, y)
plt.scatter(t,y,color='blue', marker= '*', label='Exp')
plt.scatter(t,z,color= 'red', marker='v', label='Sin')
plt.xlabel('Abscisses')
plt.ylabel('Ordonnées')
plt.title('Figure')
plt.legend(loc='best')
plt.grid()
plt.show()
```

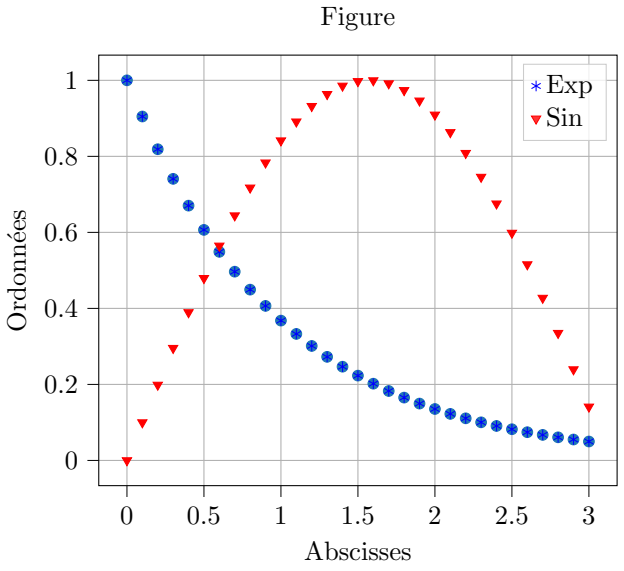


FIGURE 20 – Courbe 20

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

t = np.arange(0, 3+.1, .1)
y = np.exp(-t)
z = np.sin(t)

plt.figure()
plt.scatter(t, y)
plt.scatter(t, y, s=130, c='yellow', marker='*', edgecolors='green', label='Exp')
plt.scatter(t, z, s=110, c='red', marker='v', edgecolors='blue', label='Sin')
plt.xlabel('Abscisses')
plt.ylabel('Ordonnées')
plt.title('Figure')
plt.legend(loc='best')
plt.grid()
plt.show()
```

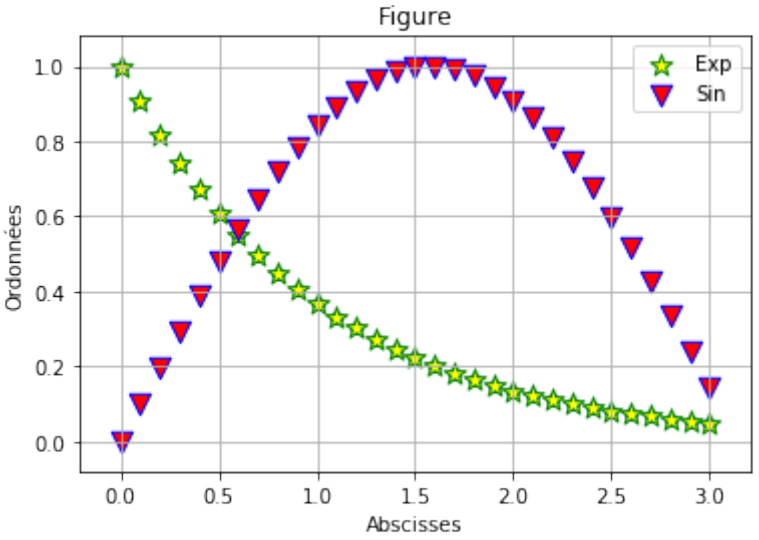


FIGURE 21 – Courbe 21

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

t = np.arange(0, 3+.1, .1)
y = np.exp(-t)
#s = np.random.rand(100)*200
plt.figure(figsize=(6,4))
plt.scatter(t, y, s=y*50,color='red')
plt.show()
```

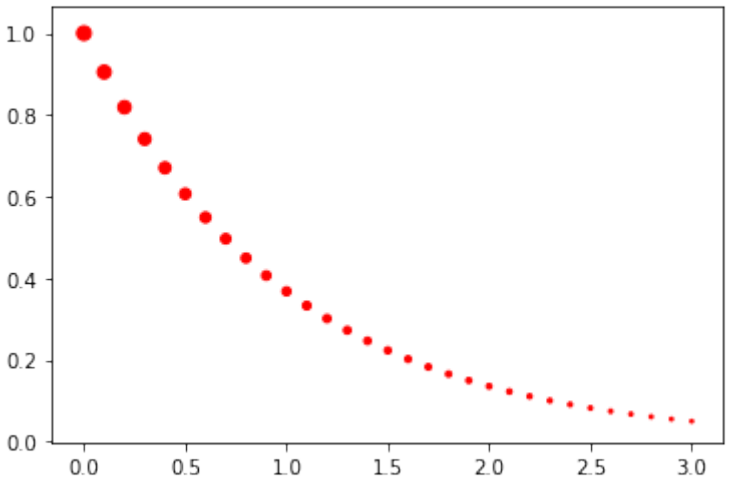


FIGURE 22 – Courbe 22

7 Echelle Log

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

x = np.logspace(-1,5,10000)
y = 1/(np.sqrt(1+(x**2/10000)))

fig = plt.figure(figsize=(10,5))

plt.subplot(121)
plt.plot(x,y)
plt.title("Echelle linéaire")
plt.grid()

plt.subplot(122)
plt.semilogx(x,y)
plt.title("Echelle Log (Axe des x)")
plt.grid()

plt.show()
```

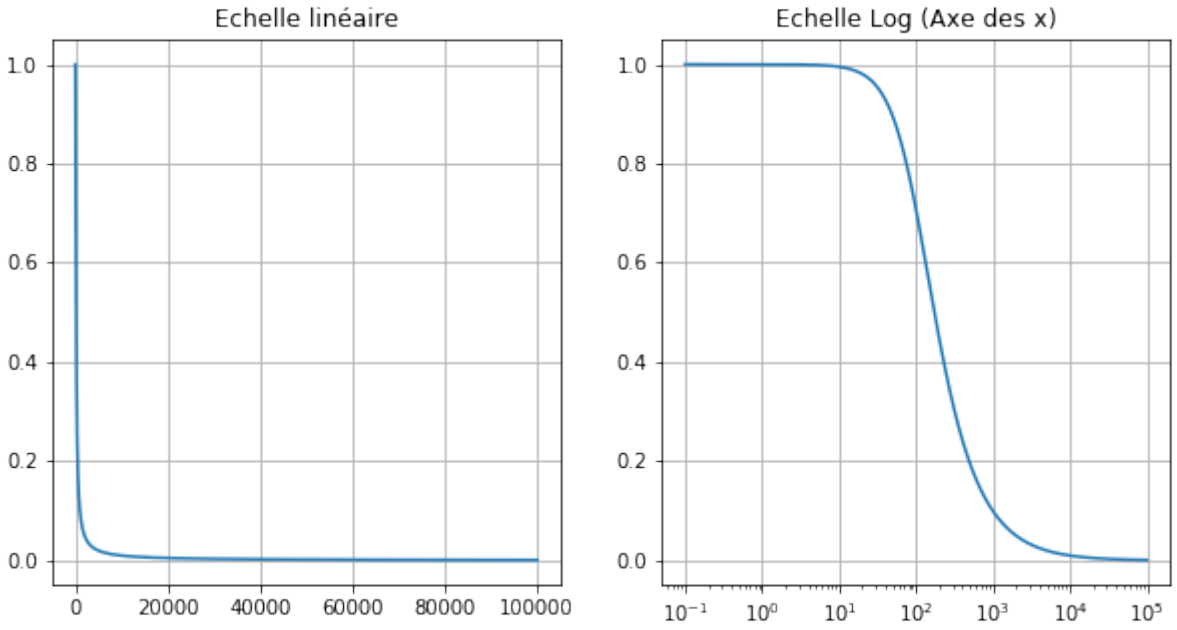


FIGURE 23 – Courbe 23

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

t = [0,1,2,3,4,5,6]
v = [1,0.1,1,10,30,10000,100]

fig = plt.figure(figsize=(10,5))

plt.subplot(121)
plt.plot(t,v)
plt.title("Echelle linéaire")
plt.grid()

plt.subplot(122)
plt.semilogy(t,v)
plt.title("Echelle Log (axe des y)")
plt.grid()

plt.show()
```

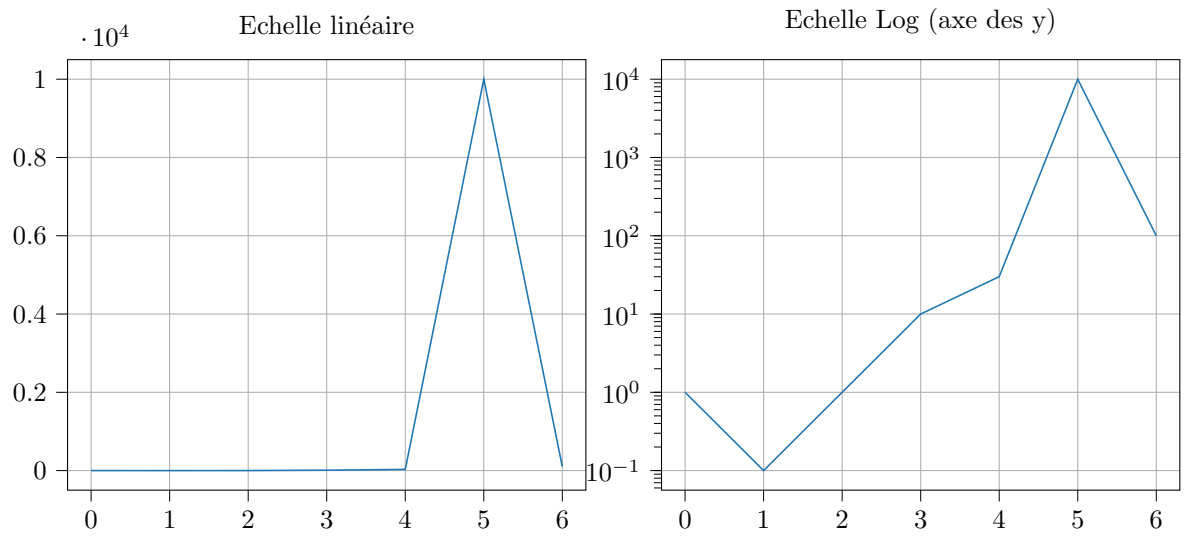


FIGURE 24 – Courbe 24

```
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline

x = np.linspace(1, 10, 10000)
y = x**8

fig = plt.figure(figsize=(10,5))

plt.subplot(121)
plt.plot(x,y)
plt.title("Echelle linéaire")
plt.grid()

plt.subplot(122)
plt.loglog(x,y)
plt.title("Echelle LogLog (axe des x et y)")
plt.grid()

plt.show()
```

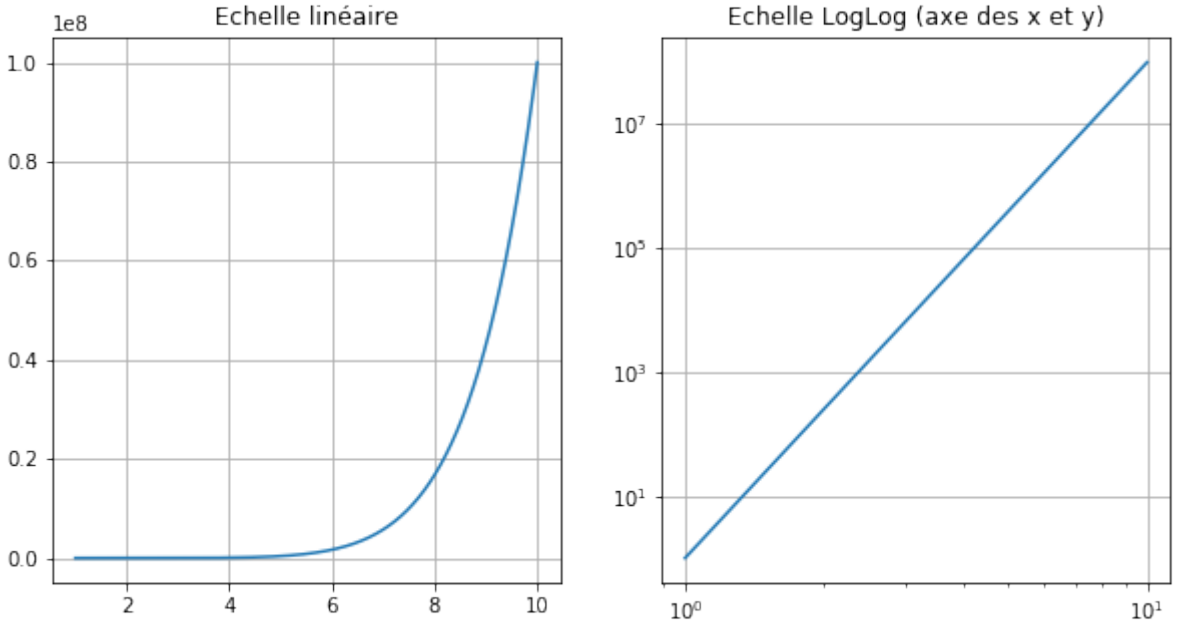


FIGURE 25 – Courbe 25

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.ticker as mtick
%matplotlib inline

x = np.linspace(1, 5, 100)
y = x**2

fig, ax = plt.subplots()
ax.loglog(x,y, basex=np.e, basey=np.e)

def ticks(y, pos):
    return r'$e^{:.0f}$'.format(np.log(y))

ax.xaxis.set_major_formatter(mtick.FuncFormatter(ticks))
ax.yaxis.set_major_formatter(mtick.FuncFormatter(ticks))

plt.show()
```

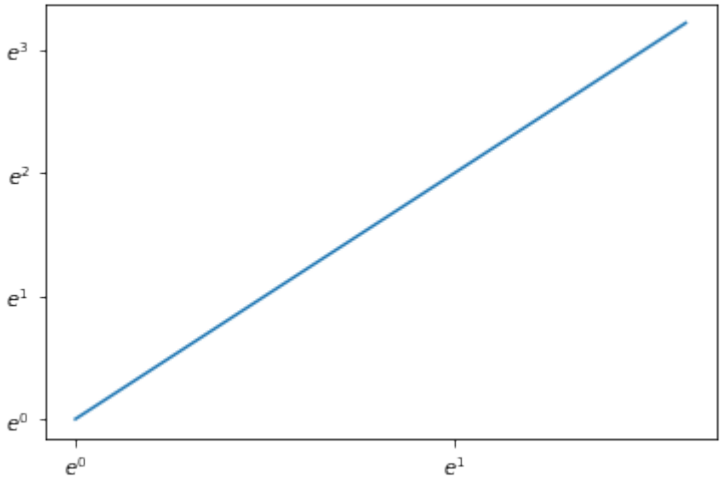


FIGURE 26 – Courbe 26

```
import matplotlib.pyplot as plt
import numpy as np
f, ax = plt.subplots()
ax.set_xscale('log')
ax.set_yscale('log')
ax.scatter(2**np.arange(10), 2**np.arange(10))

plt.show()
```

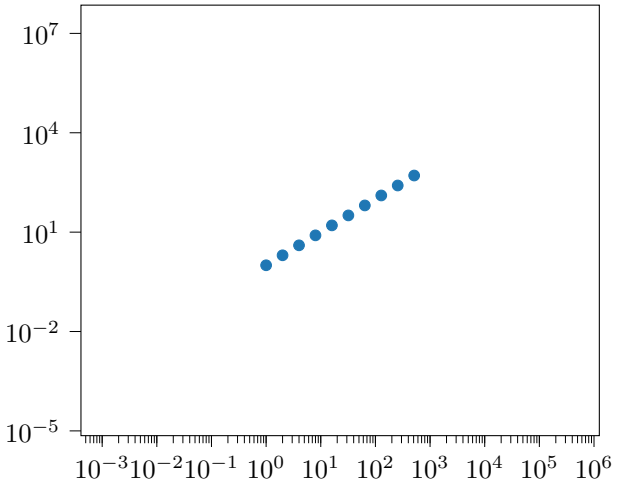


FIGURE 27 – Courbe 27

8 Annexe

Line properties

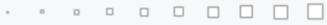
Property	Description	Appearance
alpha (or a)	alpha transparency on 0-1 scale	
antialiased	True or False - use antialised rendering	
color (or c)	matplotlib color arg	
linestyle (or ls)	see Line properties	
linewidth (or lw)	float, the line width in points	
solid_capstyle	Cap style for solid lines	
solid_joinstyle	Join style for solid lines	
dash_capstyle	Cap style for dashes	
dash_joinstyle	Join style for dashes	
marker	see Markers	
markeredgewidth (mew)	line width around the marker symbol	
markeredgecolor (mec)	edge color if a marker is used	
markerfacecolor (mfc)	face color if a marker is used	
markersize (ms)	size of the marker in points	

FIGURE 28 – Arguments de la fonction plot

Line styles

Symbol	Description	Appearance
`-`	solid line	
`--`	dashed line	
`-.`	dash-dot line	
`:`	dotted line	
`.`	points	
`,`	pixels	
`o`	circle	
`^`	triangle up	
`v`	triangle down	
`<`	triangle left	
`>`	triangle right	
`s`	square	
`+`	plus	
`x`	cross	
`D`	diamond	
`d`	thin diamond	
`1`	tripod down	
`2`	tripod up	
`3`	tripod left	
`4`	tripod right	
`h`	hexagon	
`H`	rotated hexagon	
`p`	pentagon	
`|`	vertical line	
`_`	horizontal line	

FIGURE 29 – Styles de ligne

Markers

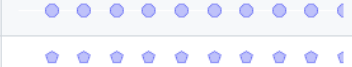
Symbol	Description	Appearance
0	tick left	
1	tick right	
2	tick up	
3	tick down	
4	caret left	
5	caret right	
6	caret up	
7	caret down	
o	circle	
D	diamond	
h	hexagon 1	
H	hexagon 2	
_	horizontal line	
1	tripod down	
2	tripod up	
3	tripod left	
4	tripod right	
8	octagon	
p	pentagon	
^	triangle up	
v	triangle down	
<	triangle left	
>	triangle right	
d	thin diamond	
,	pixel	
+	plus	
.	point	
s	square	
*	star	
|	vertical line	
x	cross	
r'$\sqrt{2}$'	any latex expression	

FIGURE 30 – Styles de marqueur