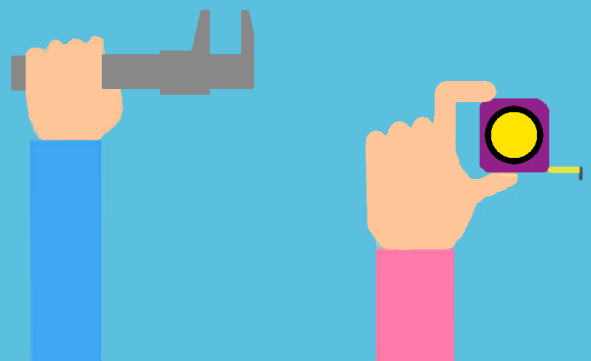
Calcul d'intégrales et tracé des résultats

Renaud Costadoat
Lycée Dorian

DORIAN



Méthode des rectangles

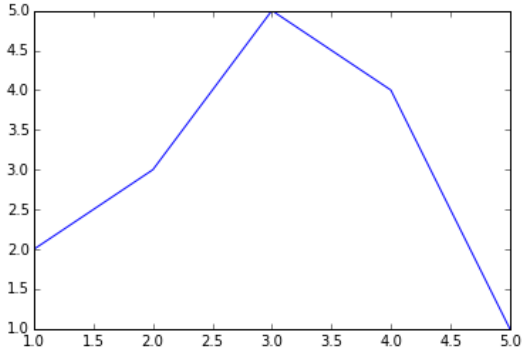
Dans cette méthode, on calcule l'intégrale numérique en réalisant une somme de surfaces de rectangles. Le domaine d'intégration est découpé en intervalles et on fait comme si la fonction restait constante sur chaque intervalle.

Sur chaque intervalle, on réalise ainsi l'approximation suivante :

$\int_a^b f(x).dx \approx (b-a).f(\alpha)$, où α est une abscisse appartenant à l'intervalle limité par a et b.

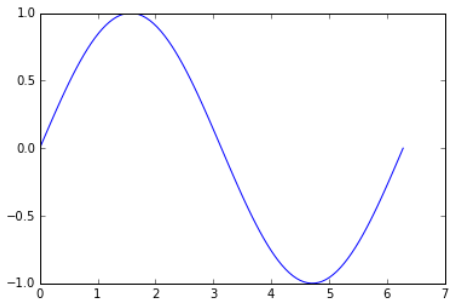
Nous nous limiterons ici aux cas où $\alpha = a$, $\alpha = b$ ou $\alpha = \frac{a+b}{2}$.

Comme exemple, nous allons réaliser un programme d'intégration pour $\alpha = a$ et nous visualiserons les rectangles. Pour tracer un rectangle ABCD (voir figure ci-dessous), il suffit de faire un plot avec les coordonnées de A, B, C, D et A. On termine par A pour fermer le tracé.

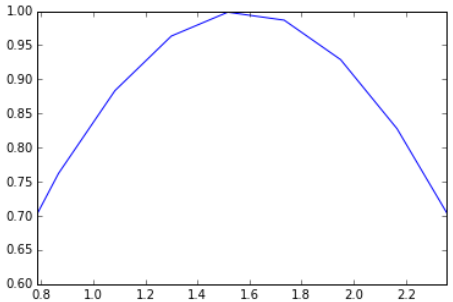


Méthode des rectangles

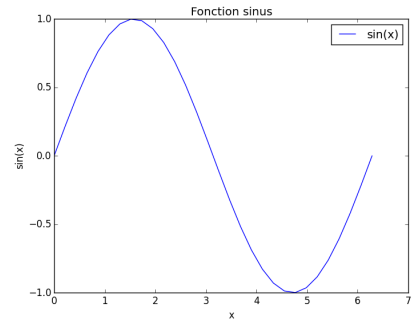
Cas $\alpha = a$



Cas $\alpha = b$



Cas $\alpha = \frac{a+b}{2}$



La méthode $\alpha = \frac{a+b}{2}$ est plus difficile à mettre en œuvre car, il faut alors découper en deux les intervalles d'intégration. Souvent a et b sont des entiers consécutifs et cela empêche cette division.

L'objectif de la méthode des rectangles est de faire la somme de la surface de ces carrés au fur et à mesure de l'avancement du programme.

Méthode des rectangles ($\alpha = a$)

Création de la fonction `y` et de la variable `x`.

```
xmin, xmax, nbx = 0, 3*pi/2, 2006
x = linspace(xmin, xmax, nbx)
y = cos(x)
plot(x,y,"bo-")
```

Calcul de l'intégrale `int`.

```
nbi = nbx - 1 # nombre d'intervalles
int = linspace(1, 1, nbx)
int[0]=0

for i in range(nbi):
    int[i+1] = int[i] + y[i]*(x[i+1]-x[i])
```

Méthode des rectangles ($\alpha = b$)

Modifier le programme pour prendre $\alpha = b$.

```
xmin, xmax, nbx = 0, 3*pi/2, 2006
x = linspace(xmin, xmax, nbx)
y = cos(x)
plot(x,y,"bo-")

nbi = nbx - 1 # nombre d'intervalles
int = linspace(1, 1, nbx)
int[0]=0

for i in range(nbi):
    int[i+1] = int[i] + y[i+1]*(x[i+1]-x[i])
```

Méthode des trapèzes

Coder la méthode des trapèzes.

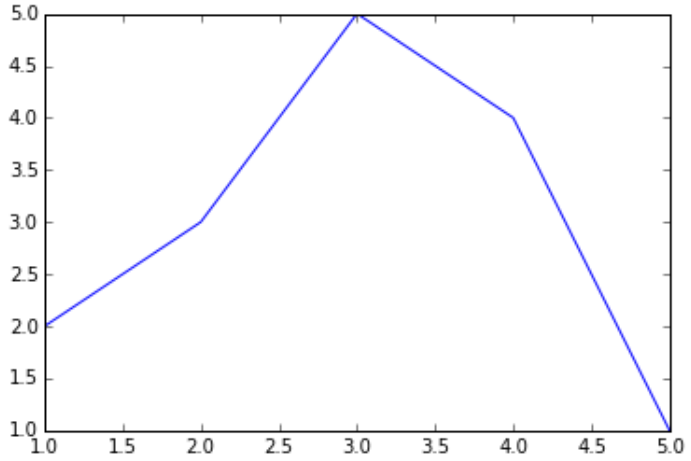
Création de courbes

L'instruction `plot()` permet de tracer des courbes qui relient des points dont les abscisses et ordonnées sont fournies dans des tableaux.

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

x = np.array(range(1,6))
y = np.array([2, 3, 5, 4, 1])
plt.plot(x, y)

plt.show() # affiche la figure
```



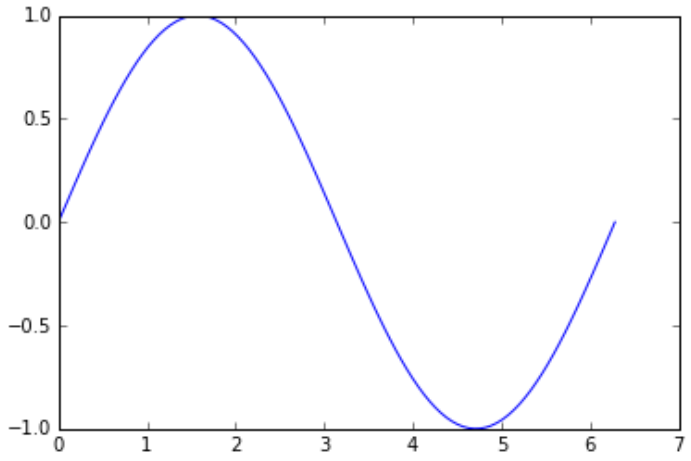
Création de courbes

De la même manière, il est possible de tracer une fonction `sin(x)`.

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

x = np.linspace(0, 2*np.pi, 100)
y = np.sin(x)
plt.plot(x, y)

plt.show()
```



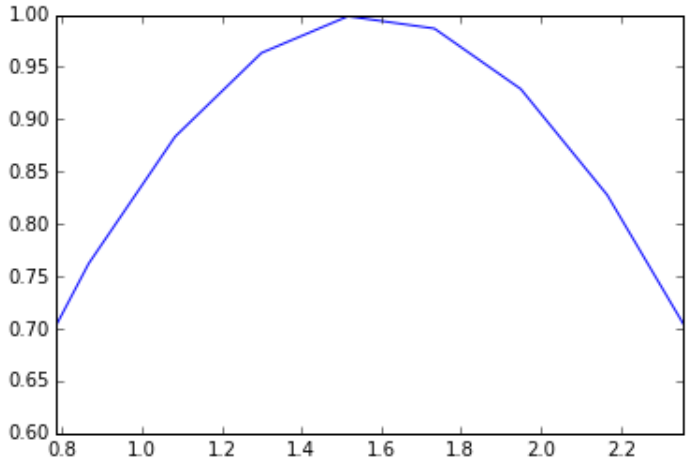
Définition du domaine des axes

Les fonction `xlim(xmin, xmax)` et `ylim(ymin, ymax)` permettent de fixer indépendamment les domaines des abscisses et des ordonnées.

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

x = np.linspace(0, 2*np.pi, 30)
y = np.sin(x)
plt.plot(x, y)
plt.xlim(np.pi/4, 3*np.pi/4)
plt.ylim(0.6, 1)

plt.show()
```



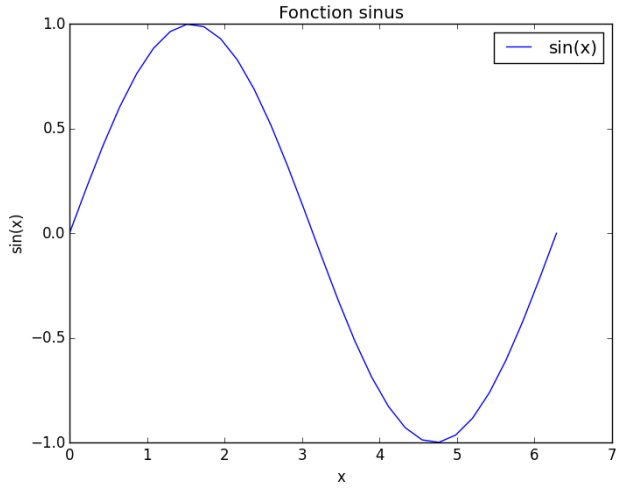
Ajouter un titre, une légende ou des labels

Les courbes tracées doivent être accompagnée d'informations afin de pouvoir être interprétées.

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

x = np.linspace(0, 2*np.pi, 30)
y = np.sin(x)
plt.plot(x, y, label="sin(x)")
plt.legend()
plt.title("Fonction sinus")
plt.xlabel("x")
plt.ylabel("sin(x)")

plt.show()
```



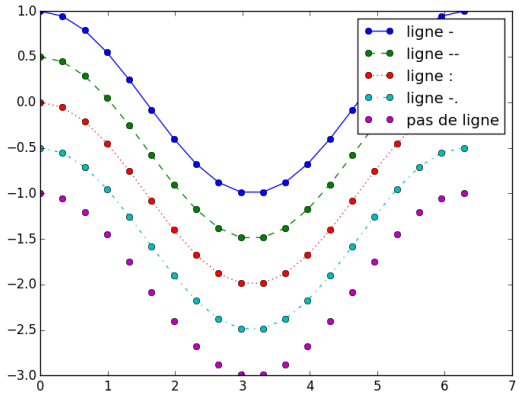
Style des lignes

Il est nécessaire de modifier le style des lignes afin de les identifier.

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

x = np.linspace(0, 2*np.pi, 20)
y = np.cos(x)
plt.plot(x, y, "o-", label="-")
plt.plot(x, y-0.5, "o--", label="--")
plt.plot(x, y-1, "o:", label=":")
plt.plot(x, y-1.5, "o-.", label="-.")
plt.plot(x, y-2, "o", label="no line")
plt.legend()

plt.show()
```

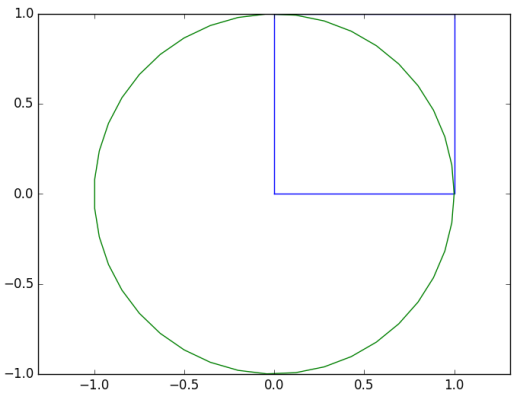


Tracé de formes

Comme la fonction `plot()` ne fait que relier des points, il est possible de lui fournir plusieurs points avec la même abscisse.

```
x = np.array([0, 1, 1, 0, 0])
y = np.array([0, 0, 1, 1, 0])
plt.plot(x, y)

theta = np.linspace(0, 2*np.pi, 40)
x = np.cos(theta)
y = np.sin(theta)
plt.plot(x, y)
plt.axis("equal")
plt.show()
```



Tracé de formes

Déterminer les k_i, ω_i, φ_i et c_i et coder le tracé du profil de vitesse $v(t) = \dot{x}(t)$ suivant:

- $0 \leq t \leq 1 : \dot{x}(t) = k_1.sin(\omega_1.t + \varphi_1) + c_1$,
- $1 < t \leq 3 : \dot{x}(t) = v_{max}$,
- $3 < t \leq 5 : \dot{x}(t) = k_2.sin(\omega_2.t + \varphi_2) + c_2$.