

Utilisation de python en physique

Savoir faire 1 : utilisation de fonctions

Exemple:

```
def g(t):
    —return 3*np.sin(t)
```

Exemple:

```
def f(x,y):
    —return x**2+4*y
f(2,3) renvoie
f(x,1) renvoie
f(2,y) renvoie
```

Savoir faire 2 : utilisation de listes et des vecteurs

<i>Syntaxe concernant les listes</i>	<i>Illustration</i>
Une liste l s'écrit de la forme l=[...,...,...]	l=[1,3,5,7]
len(l) : désigne le nombre de termes dans la liste	len(l)=
l[0] : désigne le 1er terme de la liste	l[0]=
l[i] : désigne le i+1 terme de la liste	l[2]=
l.append(i) : sert à ajouter un terme dans la liste, ce terme a pour valeur i	l.append(9) renvoie l=

<i>Syntaxe concernant les vecteurs</i>	<i>Illustration</i>
Un vecteur s'écrit v=np.array([...,...,...])	v=array([2,4,6,8])
len(v) : désigne le nombre de termes dans le vecteur	len(v)=
v[0] : désigne le 1er terme du vecteur	v[0]=
v[i] : désigne le i+1 terme de la liste	v[2]=
v=np.zeros(N) crée un vecteur contenant N zéros	v=np.zeros(2) renvoie

Remarque: intérêt d'une liste: on peut à tout moment ajouter des termes, c'est très utile lorsque l'on ne connaît pas à l'avance le nombre de termes de la liste.

Remarque: intérêt d'un vecteur: on peut appliquer une fonction f aux différents termes d'un vecteur (on ne peut pas le faire avec les termes d'une liste).

v est un vecteur de la forme array([v[0],v[1],...]), f(v) renvoie un vecteur de la forme array([f(v[0]),f(v[1]),...])

Exemple 1:

```
lt=[0]
dt=0.1
N=100
for i in range(N):
    —lt.append(lt[i]+dt)
```

Exemple 2:

```
lx=[0]
lx=[1]
g,dt,N=10,0.2,500
for i in range(N):
    —lx.append((i+1)*dt)
    —lx.append(lx[i]+g*lx[i]**2)
```

Exemple 3:

```
v=np.zeros(4)
for i in range(len(v)):
    v[i]=3*i
```

Exemple 4:

```
v=np.zeros(3)
v[0]=2
for i in range(len(v)-1):
    v[i+1]=v[i]+2*i+3
```

Exemple 5:

```
v=np.array([1,2,3,4])
def f(x):
    return x**2
v1=f(v)
```

Savoir faire 3 : tracer une courbe

La syntaxe à utiliser, donnée dans l'énoncé d'un sujet de concours, est:

`plt.plot(x,y)` : x et y sont deux listes (ou deux vecteurs) contenant le même nombre de termes. Python trace les points de coordonnées $[x_i, y_i]$ et les relie entre eux par des segments

`plt.xlabel('variable en x')` : permet d'écrire le nom de la variable sur l'axe des abscisses

`plt.ylabel('variable en y')` : permet d'écrire le nom de la variable sur l'axe des ordonnées

`plt.title('titre')` : permet d'écrire le titre du graphe

`plt.grid()` : permet d'ajouter un quadrillage pratique pour la lecture des coordonnées des points

`plt.show()` : permet d'afficher la courbe

Cas 1: tracé de points de mesures expérimentaux

$x=[x_1, x_2, \dots, x_N]$: liste de valeurs de la variable x

$y=[y_1, y_2, \dots, y_N]$: liste de valeurs de la variable y

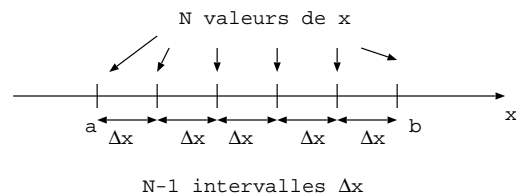
`plt.plot(x,y,'o')` : trace les points expérimentaux représentés par des •

`plt.plot(x,y,'*')` : trace les points expérimentaux représentés par des *

Cas 2: tracé d'une fonction $f(x)$ sur un intervalle $[a, b]$

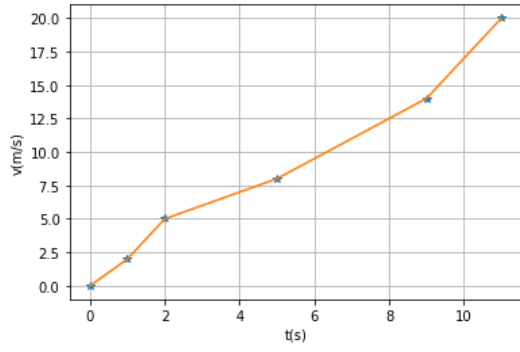
Pour créer le vecteur avec les valeurs en x , les deux possibilités les plus fréquentes reposent sur l'utilisation de `np.linspace` ou `np.arange`. Ensuite on obtient le vecteur contenant les valeurs de y en appliquant la fonction f au vecteur x .

`x=np.linspace(a,b,N)` : crée un vecteur contenant N valeurs régulièrement espacées entre a et b . L'intervalle Δx entre deux valeurs consécutives est $\Delta x = \frac{b-a}{N-1}$: on parle d'échantillonnage.

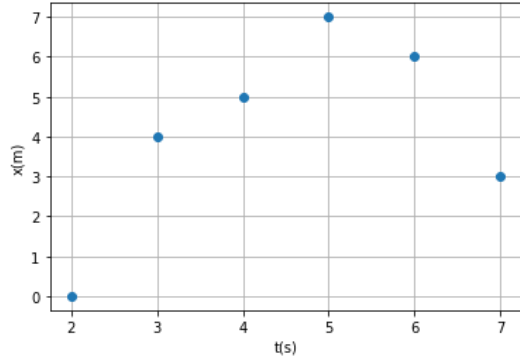


`x=np.arange(a,b,p)` : crée un vecteur dont le premier terme est a , le dernier terme est inférieur ou égal à b et où p désigne le pas. Le vecteur est donc de la forme `array([a,a+p,a+2p,...,a+Np ≤ b])`

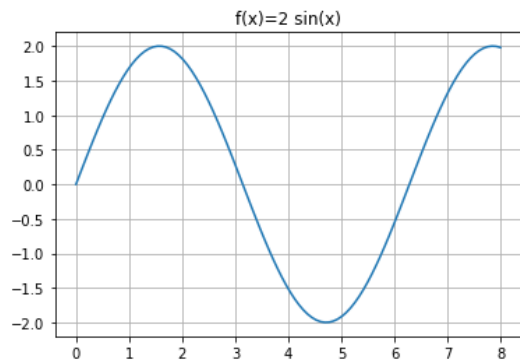
Exemple 1:



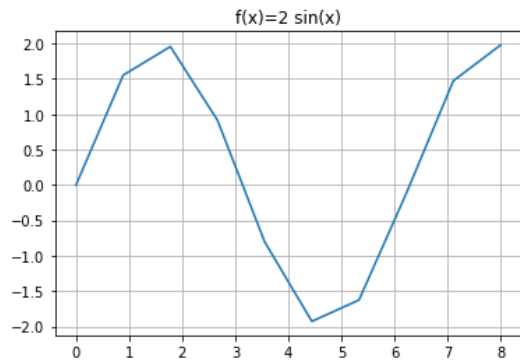
Exemple 2:



Exemple 3:



Exemple 4:



Savoir faire 4: méthode d'Euler pour une résolution numérique d'équations différentielles

La méthode d'Euler repose sur le DL à l'ordre 1 en dt : $f(t + dt) = f(t) + f'(t)dt = f(t) + \frac{df}{dt}dt$ où dt est le pas de temps (petit).

Dans le cas où $f(t)$ est la fonction vitesse: $v(t + dt) =$

d'où $v(t_{i+1}) =$

Dans le cas où $f(t)$ est la fonction position: $x(t + dt) =$

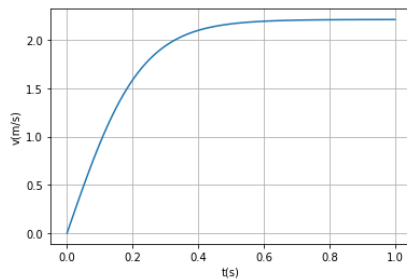
d'où $x(t_{i+1}) =$

Exemple 1:

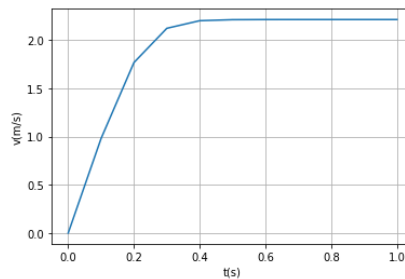
```
9 import matplotlib.pyplot as plt
10 import numpy as np
11
12 lt,lv=[0],[0]
13 g,k=9.8,2
14 tau,N=0.01,100
15 for i in range(N):
16     lt.append(lt[i]+tau)
17     a=-k*lv[i]**2+g
18     lv.append(lv[i]+tau*a)
19
20 plt.plot(lt,lv)
21 plt.grid()
22 plt.xlabel('t(s)')
23 plt.ylabel('v(m/s)')
24 plt.show()
```

L'exécution du code donne les courbes suivantes en modifiant la ligne 14:

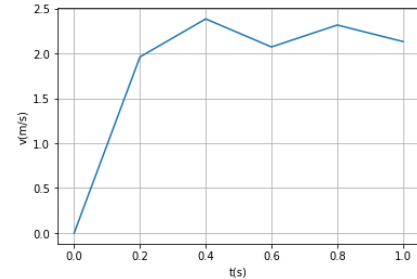
tau,N=0.01,100



tau,N=0.1,10



tau,N=0.2,5



Commenter le code et les courbes obtenues.

Exemple 2:

```
27 N,tau=500,0.02
28 lt=np.zeros(N)
29 lv=np.zeros(N)
30 lx=np.zeros(N)
31 lt[0],lv[0],lx[0]=0,10,0
32 for i in range(len(lt)-1):
33     lt[i+1]=lt[i]+tau
34     a=-lv[i]**3
35     lv[i+1]=.....
36     lx[i+1]=.....
```

Compléter le code et l'expliquer.