

Utilisation de listes ou de tableaux et tracé de courbes

1. Donnée en annexe d'un sujet de concours

Bibliothèque NUMPY

Dans les exemples ci-dessous, la bibliothèque `numpy` a préalablement été importée à l'aide de la commande : `import numpy as np`. On peut alors utiliser les fonctions de la bibliothèque, dont voici quelques exemples :

- **`np.linspace(start, stop, N point)` :**
 - Description : renvoie un nombre d'échantillons espacés uniformément, calculés sur l'intervalle `[start, stop]` ;
 - Argument d'entrée : début, fin et nombre d'échantillons dans l'intervalle ;
 - Argument de sortie : un tableau.

Commande	Résultat
<code>np.linspace(1, 4, 5)</code>	<code>[1., 1,75, 2,5, 3,25, 4.]</code>

- **`np.zeros(i)` :**
 - Description : renvoie un tableau de taille `i` rempli de zéros ;
 - Argument d'entrée : un scalaire
 - Argument de sortie : un tableau.

Commande	Résultat
<code>np.zeros(5)</code>	<code>[0, 0, 0, 0, 0]</code>

- **`np.array(liste)` :**
 - Description : crée une matrice (de type tableau) à partir d'une liste.
 - Argument d'entrée : une liste définissant un tableau à 1 dimension (vecteur) ou 2 dimensions (matrice).
 - Argument de sortie : un tableau (matrice).

Commande	Résultat
<code>np.array([4, 3, 5])</code>	<code>[4, 3, 5]</code>

Bibliothèque MATPLOTLIB.PYLOT

Cette bibliothèque permet de tracer des graphiques. Dans les exemples ci-dessous, la bibliothèque `matplotlib.pyplot` a préalablement été importée à l'aide de la commande : `import matplotlib.pyplot as plt`.

- Description : fonction permettant de tracer un graphique de `n` points dont les abscisses sont contenues dans le vecteur `x` et les ordonnées dans le vecteur `y`. Cette fonction doit être suivie de la fonction `plt.show()` pour que le graphique soit affiché.
- Argument d'entrée : un vecteur d'abscisses `x` (tableau de `n` éléments) et un vecteur d'ordonnées `y` (tableau de `n` éléments). La chaîne de caractères 'SC' précise le style et la couleur de la courbe tracée. Des valeurs possibles pour ces deux critères sont :

Valeurs possibles pour S (style) :				
Description	Ligne continue	Ligne traitillée	Marqueur rond	Marqueur plus
Symbole S	-	--	o	+

Valeurs possibles pour C (couleur) :				
Description	bleu	rouge	vert	noir
Symbole C	b	r	g	k

- Argument de sortie : un graphique.

```
x = np.linspace(3, 25, 5)
y = np.sin(x)
plt.plot(x,y,'-b') # tracé d'une ligne bleue continue
plt.title('titre_graphique') # titre du graphe
plt.xlabel('x') # titre de l'axe des abscisses
plt.ylabel('y') # titre de l'axe des ordonnées
plt.show()
```

2. Savoir faire 1 : utilisation de listes ou de tableaux

Syntaxe concernant les listes

Une liste l s'écrit de la forme $l=[\dots,\dots]$

$\text{len}(l)$: désigne le nombre de termes dans la liste

$l[0]$: désigne le 1er terme de la liste

$l[1]$: désigne le 2ième terme de la liste

$l[2]$: désigne le 3ième terme de la liste...

$l.\text{append}(b)$: sert à ajouter un terme dans la liste, ce terme a pour valeur b

Syntaxe concernant un tableau à une dimension

Un tableau à une dimension (ou vecteur) v se note $v=\text{np.array}([\dots,\dots])$

$\text{len}(v)$: désigne le nombre de termes dans le vecteur

$v[0]$: désigne le 1er terme du vecteur

$v[1]$: désigne le 2ième terme du vecteur

$v[2]$: désigne le 3ième terme du vecteur

$v=\text{np.zeros}(N)$: sert à créer un vecteur contenant N termes tous égaux à 0

Exemple 1:

$l=[1,3,7,8,11]$

$\text{len}(l)=5$

$l[2]=7$

$l.\text{append}(15)$ renvoie $l=[1,3,7,8,11,15]$

Exemple 2:

$v=\text{np.zeros}(4)$

$v = \text{np.array}([0,0,0,0])$

$\text{len } v$

for i in $\text{range}(\text{len}(v))$:

— $v[i]=2*i$

L'exécution de ce code donne $v=$

Exemple 3:

$l=[0]$

for i in $\text{range}(3)$:

— $l.\text{append}(l[i]+2)$

L'exécution de ce code donne $l=$

Exemple 4:

$x=\text{np.linspace}(0,3,4)$

def $f(a,b)$:

— return $3*b**2+a$

$y=f(2,x)$

L'exécution de ce code donne:

$x=$

$y=$

Exemple 5:

```
l=[1,3,7,8]
```

```
s=0
```

```
for i in range(len(l)):
```

```
—s=s+l[i]
```

L'exécution de ce code donne s=

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

Exemple: `v=np.array([0,2,4,6])`

`w=v**2` renvoie

```
def f(x):
```

```
—return 3*x+2
```

`y=f(v)` renvoie

3. Savoir faire 2: tracer une courbe

Cas 1: tracé de points de mesures expérimentaux:

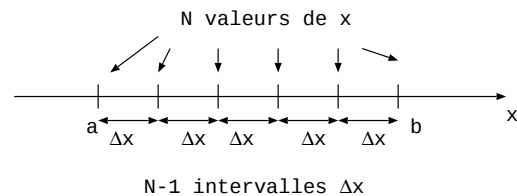
`x=[x1,x2,...,xN]` ou `x=np.array([x1,x2,...,xN])` : on écrit les valeurs de x dans une liste ou un vecteur

`y=[y1,y2,...,yN]` ou `y=np.array([y1,y2,...,yN])` : on écrit les valeurs de y dans une liste ou un vecteur

`plt.plot(x,y,'*')` : cela permet de tracer les points expérimentaux ici représentés par le symbole *

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

`x=np.linspace(a,b,N)` : cela 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.

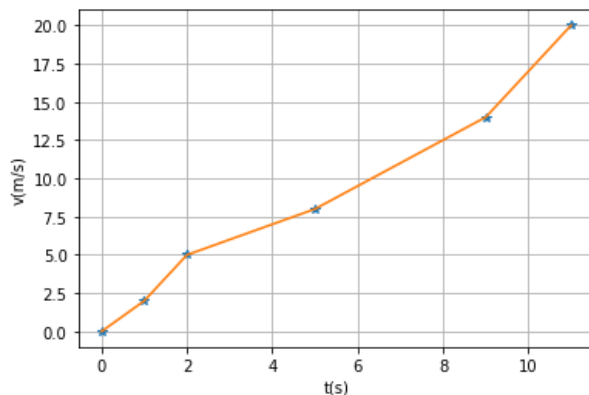


```
def f(x):
```

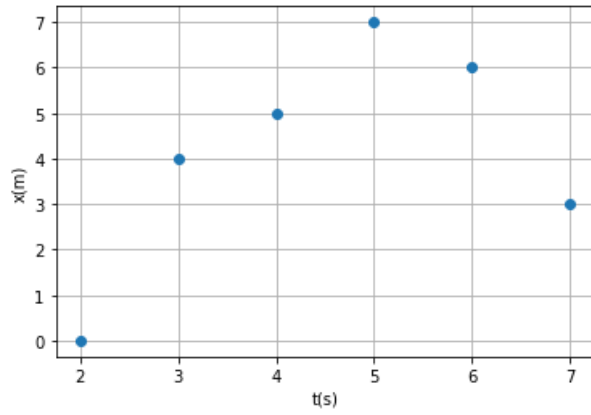
```
—return ..... : on définit la fonction f(x)
```

`plt.plot(x,f(x))` : permet de tracer la courbe $f(x)$, les points sont reliés entre eux sur cette courbe.

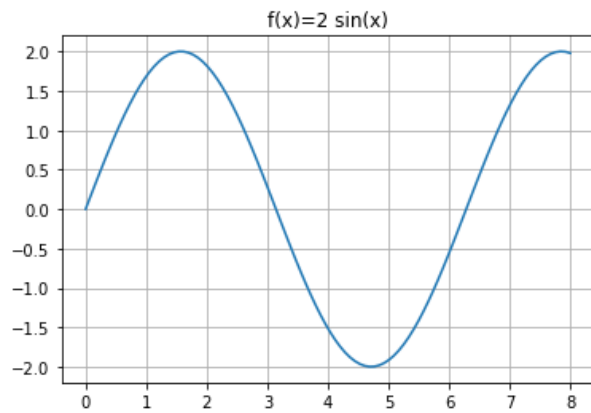
Exemple 1:



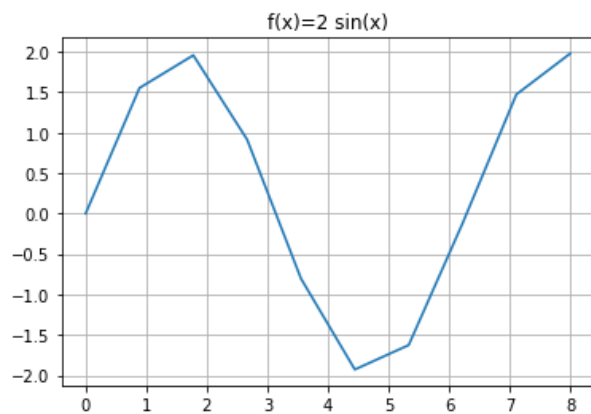
Exemple 2:



Exemple 3:



Exemple 4:



4. Exercice

Soit un projectile que l'on lance vers le haut depuis une altitude $h = 1 \text{ m}$ avec une vitesse verticale $v_0 = 3 \text{ m.s}^{-1}$. On note Oz l'axe verticale ascendant et $z(t)$ l'altitude du projectile au cours du temps. On néglige tout frottement. Donnée: $g = 9,8 \text{ m.s}^{-2}$.

1- Etablir l'équation de la trajectoire $z(t)$.

On désire tracer la courbe donnant $z(t)$ jusqu'à ce que le projectile tombe sur le sol.

2- Dans un premier temps, on utilise sous python, une liste temps de la forme $t_i = idt$ où dt est le pas de temps choisi et une liste $z_i = z(t_i)$. Compléter le code suivant:

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 #Données
5 g,h,v0=.....,.....,....
6 #Lancer d'un projectile: utilisation de liste
7 lt,lz=[...],[...] #on initialise les listes t(temps) et z(hauteur)
8 dt=20e-3 #c'est le pas de temps
9 i=0
10 while .....: #condition d'arrêt du calcul de t et de z(t) |
11     lt.append(....
12     lz.append(....
13     i=i+1
14 plt.plot(...,...) #on trace z(t)
15 plt.grid()
16 plt.xlabel('.....')
17 plt.ylabel('.....')
18 plt.show()
```

3- Déterminer, en fonction de v_0 , g et h , l'expression théorique du temps t_f pour lequel le projectile tombe sur le sol. Dans un second temps, on utilise sous python, des vecteurs t pour le temps et z pour l'altitude du projectile. Compléter le code suivant:

```
20 #Lancer d'un projectile: utilisation d'un vecteur
21 tf=.....
22 def z(t):
23     return .....
24 t=np.linspace(...,...,1000)
25 plt.plot(...,...) #on trace z(t)
26 plt.grid()
27 plt.xlabel('.....')
28 plt.ylabel('.....')
29 plt.show()
```