

Méthodes pour python en physique

Annexe donnée sur le sujet de concours de l'épreuve de modélisation de CCINP.

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()
```

Savoir faire 1 : utilisation de listes ou de vecteurs

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...

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

Syntaxe concernant un vecteur

Un vecteur v se note $v=np.array([\dots,\dots])$

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

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

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

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

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

Syntaxe pour réaliser la somme des termes d'une liste ou la sommes des termes d'un vecteur:

Soit l une liste ou un vecteur:

$s=0$

for i in range(len(l)):

— $s=s+l[i]$

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.

Exemple: Soit un projectile lancé à l'instant $t = 0$ depuis la hauteur $h = 1$ m avec une vitesse verticale $v_0 = 3$ m/s. On néglige tout frottement. On cherche sa position $z(t)$ aux instants $t_i = i dt$ jusqu'à ce qu'il tombe sur le sol. Donnée: $dt = 20$ ms.

$g,h,v_0,dt=$

$lt,lz=[0],[h]$

$i=0$

while

— $lt.append($

— $lz.append($

— $i=i+1$

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

Exemple:

$v=np.linspace(0,10,11)$

$w=3*v**2$

L'exécution du code renvoie:

Exemple 1:

```
l=[1,3,7,8,11]
len(l)=
l[2]=
l.append(15) renvoie l=
```

Exemple 2:

```
v=np.zeros(4)
for i in range(len(v)):
    v[i]=2*i
L'exécution de ce code donne v=
```

Exemple 3:

```
l=[0]
for i in range(3):
    l.append(l[i]+2)
L'exécution de ce code donne l=
```

Exemple 4:

```
x=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=
```

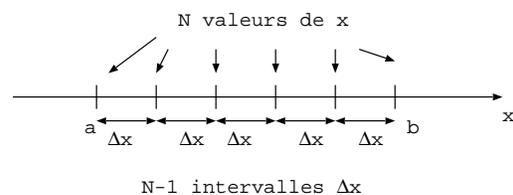
Savoir faire 2: tracer une courbe

Cas 1: tracé de points de mesures expérimentaux:

$x=[x_1, x_2, \dots, x_N]$: on écrit les valeurs de x dans une liste
 $y=[y_1, y_2, \dots, y_N]$: on écrit les valeurs de y dans une liste
`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.

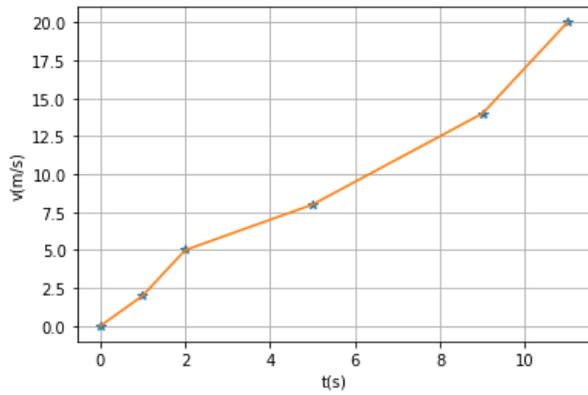


def f(x):

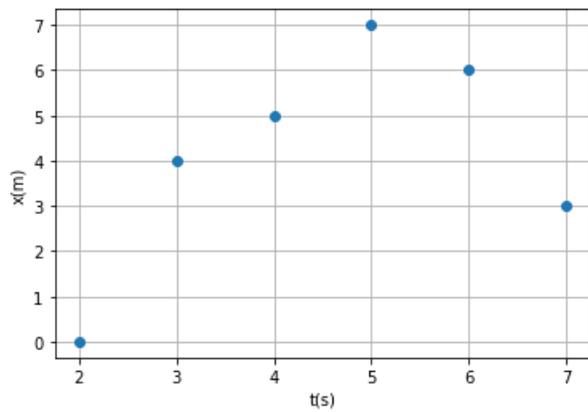
——return : on définit la fonction f(x)

y=f(x) : on applique la fonction f aux N valeurs du vecteur x et on obtient donc un vecteur contenant N valeurs des ordonnées associées aux valeurs des abscisses x

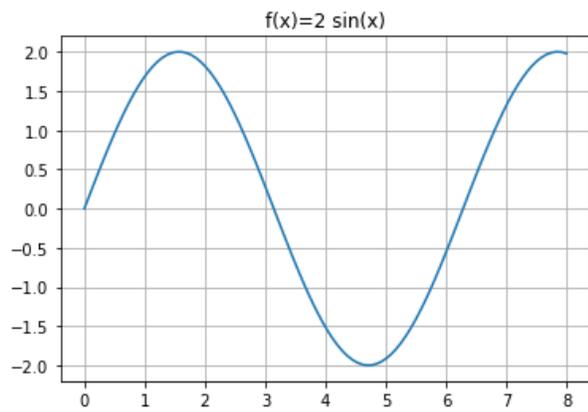
Exemple 1:



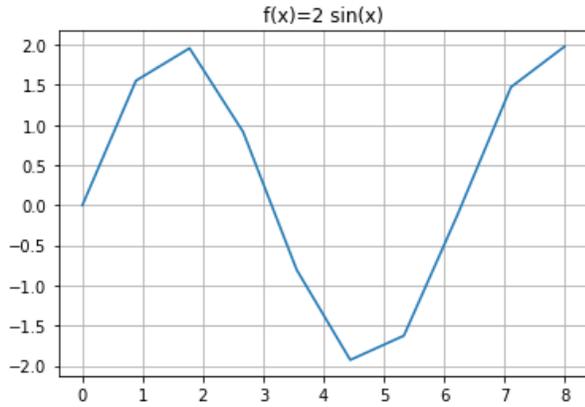
Exemple 2:



Exemple 3:



Exemple 4:



Savoir faire 3 : Méthode d'Euler

La méthode d'Euler repose sur le DL à l'ordre 1 en dt: $f(t + dt) = f(t) + f'(t)dt$
ou $f(t_{i+1}) = f(t_i) + (t_{i+1} - t_i)f'(t_i)$.

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

ou $v(t_{i+1}) =$

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

ou $x(t_{i+1}) =$

Principe de la méthode et code de base:

On initialise les listes de temps, position et vitesse.

On rentre les valeurs numériques de l'échantillon et du nombre de points de mesures.

On utilise une boucle for ou une boucle while pour compléter les listes de temps, vitesse et position en utilisant les relations de récurrence:

$$t_{i+1} = t_i + dt$$

$$v_{i+1} = v_i + a_i * dt$$

$$x_{i+1} = x_i + v_i * dt$$

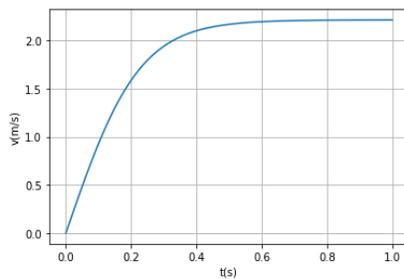
.

Exemple:

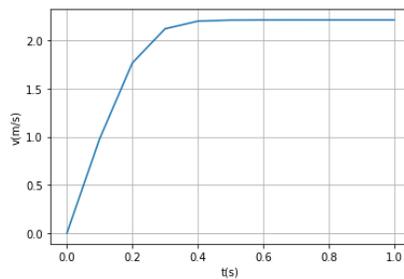
```
1 lt,lv,lx=[0],[0],[0]
2 g,k=9.8,2
3 dt,N=...,...
4 for i in range(N):
5     lt.append(lt[i]+dt)
6     a=-k*lv[i]**2+g
7     lv.append(lv[i]+dt*a)
8 plt.plot(lt,lv)
9 plt.grid()
10 plt.xlabel('t(s)')
11 plt.ylabel('v(m/s)')
12 plt.show()
```

L'exécution du code donne les courbes suivantes en modifiant la ligne 3:

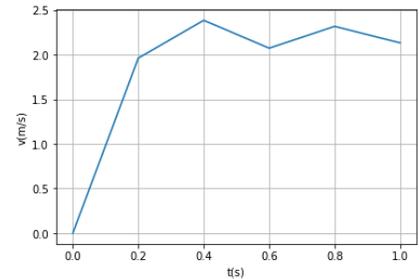
dt,N=0.01,100



dt,N=0.1,10



dt,N=0.2,5



Commenter le code et les courbes obtenues.

Compléter le code pour en déduire les valeurs de $x(t_i)$ aux instants $t_i = idt$ jusqu'à ce que $x(t)$ atteigne $x_i = 10$ m. Tracer la courbe $x(t)$.