éduSCOL
Informer et accompagner les professionnels de l'éducation

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

MINISTÈRE DE L'ÉDUCATION NATIONALE ET DE LA JEUNESSE

VOIE GÉNÉRALE ET TECHNOLOGIQUE

2DE | 1RE | TLE

Physique-chimie

ENSEIGNEMENT COMMUN

PROGRAMMER EN PYTHON
FICHE N°2 : PREMIÈRE EXPLOITATION, PREMIER MODÈLE

L'étude de la chronophotographie de la chute d'une bille est ici reprise avec le début du script précédent.

Capacité numérique mise en œuvre : représenter les positions successives d'un système modélisé par un point lors d'une évolution unidimensionnelle ou bidimensionnelle à l'aide d'un langage de programmation.

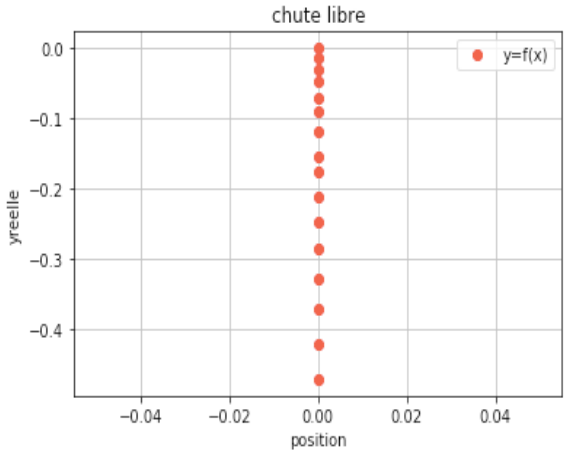
```
import numpy as np
import matplotlib.pyplot as plt
```

```
ymes=np.array([-0,-0.7,-1.5,-2.3,-3.5,-4.5,-5.9,-7.7,-8.8,-10.6,-12.3,-14.2,-16.4,-18.5,-21,-23.5])
yreelle=ymes*2/100
t=np.linspace(0,1/4,16)
```

Représentation de la chute

En affectant une abscisse constante x, il est possible d'obtenir le tracé des positions successives de la bille et ainsi recréer la chronophotographie. Le script précédent est repris en indiquant que x est nul pour tout t. Cela s'écrit sous la forme x = 0*t. Cette écriture est rendue nécessaire car la fonction plt.plot doit disposer de deux listes ayant le même nombre de valeurs. La liste yreelle contient 14 valeurs. L'abscisse doit également en contenir 14, ce qui ne serait pas le cas si l'on écrivait seulement x = 0. Une manière d'écrire cela est :

```
x=0*t
plt.plot(x,yreelle,'ro',label="y=f(x)")
plt.xlabel("position")
plt.ylabel("yreelle")
plt.grid()
plt.legend()
plt.title("chute libre")
plt.show()
```



Premier modèle automatique

Dans un premier temps, il est possible d'appeler la fonction np.polyfit qui modélise une fonction par un polynôme de degré laissé au choix de l'utilisateur. En choisissant un polynôme d'ordre 2, np.polyfit recherche la fonction qui à t associe $a.t^2+b.t+c$. En nommant la fonction mod, np.polyfit renvoie un tableau avec les valeurs de a, b et c. Il est alors possible d'afficher l'une des valeurs de ce tableau (par exemple, la première) grâce à print(mod[0]) (la numérotation de python commence toujours à 0 et non à 1). Ainsi, par comparaison avec les expressions littérales du mouvement (modélisation de la chute libre verticale au moyen de la seconde loi de Newton), il est possible d'obtenir les valeurs de la constate de gravitation g = - 2.a, de la vitesse verticale initiale v0 = b et de la position initiale z0 = c. Cela peut être écrit ainsi :

```
mod=np.polyfit(t,yreelle,2)
print('mod :', mod)
```

```
mod : [-4.64117647e+00 -7.09235294e-01 -6.81372549e-04]
```

```
print('g = ',-mod[0]*2)
print('vo = ', mod[1])
print('yo = ',mod[2])
```

```
g =  9.282352941176475
vo =  -0.709235294117647
yo =  -0.000681372549019614
```

Conclusion : l'ordre de grandeur de la valeur modèle de g ne paraît pas trop éloigné de celui de la valeur tabulée. Une réflexion peut être amorcée sur les chiffres significatifs.

Représentation du modèle

Il est ensuite possible de superposer la courbe issue de la modélisation et les points expérimentaux :

```
ymod=mod[0]*t**2+mod[1]*t+mod[2]
plt.plot(t,yreelle,'ro',label="yréelle=f(t)")
plt.plot(t,ymod,'b-',label="ymod=f(t)")
plt.xlabel("temps")
plt.ylabel("yreelle")
plt.grid()
plt.legend()
plt.title("chute libre et son modèle")
plt.show()
```