



## PROGRAMMER EN PYTHON

### FICHE N°3 : VECTEURS DÉPLACEMENT ET BOUCLE FOR

L'étude de la chronophotographie de la chute d'une bille est complétée par un tracé des vecteurs « variation de position » au cours du temps.

Capacité numérique mise en œuvre : représenter des vecteurs vitesse d'un système modélisé par un point lors d'un mouvement à l'aide d'un langage de programmation.

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

yimes=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=yimes*2/100
t=np.linspace(0,1/4,16)
```

#### *Présentation de la boucle for*

L'utilisation d'une boucle for permet d'automatiser le tracé des vecteurs déplacement. Pour cela, il faut d'abord générer une liste d'entiers de 0 à 15. On peut directement calculer la longueur de la liste t en Python, grâce à la fonction len(t), cette dernière option est intéressante, car elle est directement transposable à d'autres longueurs de vidéo.

Cela permettra d'appeler successivement des valeurs de la liste yreelle qui contient 16 valeurs. Il n'y a cependant que 15 valeurs de différences d'ordonnée du type yreelle[i+1] - yreelle[i] : l'indice i ne doit donc pas dépasser la valeur 14.

```
n=np.arange(len(t)-1)
print('n:',n)

n: [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14]
```

Comme précédemment, la liste des valeurs successives de x est générée par x=0\*t pour faire tracer les vecteurs en fonction de la position x=0.

Ensuite, la boucle for permet faire défiler les différentes positions de la bille grâce à l'instruction « for i in n ».

Attention à ne pas oublier les « : » après l'instruction for, car python renverrait un message d'erreur sans cela.

## Tracé des vecteurs

Le tracé de vecteur s'effectue avec la fonction `plt.arrow` qui a pour arguments successifs :

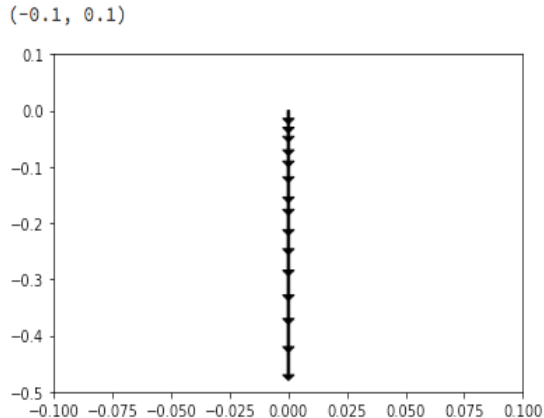
- abscisse de départ du vecteur : ici, 0 ;
- ordonnée de départ du vecteur : ici, `yreelle[i]`, c'est-à-dire le *i*ème élément de la liste des `yreelle` ;
- longueur horizontale de la flèche : ici, 0 ;
- longueur verticale de la flèche: ici, `yreelle[i+1] - yreelle[i]`, c'est-à-dire le déplacement vertical lors du *i*ème pas.

L'allure des flèches peut ensuite être améliorée grâce à des paramètres optionnels de couleur de fond (`fc='k'`), de couleur du tour (`ec='k'`), de largeur de la tête (`head_width=0.05`), de longueur de la tête (`head_length=0.02`).

## Script et rendu

Le rendu le plus simple est obtenu avec les lignes de script suivantes :

```
x=0*t
for i in n :
    plt.arrow(0,yreelle[i],0,yreelle[i+1]-yreelle[i],fc="k", ec="k",head_width=0.005, head_length=0.01)
plt.ylim(-0.5,0.1)
plt.xlim(-0.1,0.1)
```



Retrouvez éducol sur :

