



PROGRAMMER EN PYTHON

FICHE N°4 : VECTEURS VITESSE ET BOUCLE FOR

L'étude de la chronophotographie de la chute d'une bille est complétée par le tracé des vecteurs vitesse au cours du temps.

Capacité numérique mise en oeuvre : 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

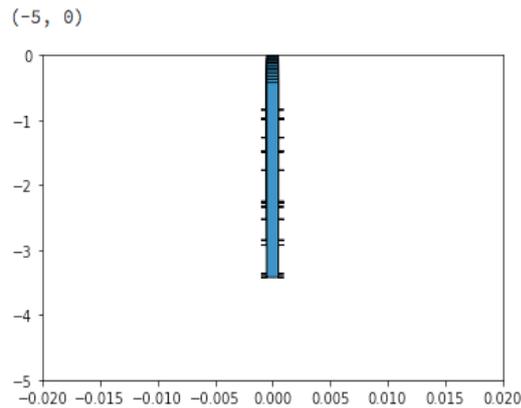
ymes=np.array([-0,-0.7,-1.5,-2.3,-3.5,-4.5,-5.9,-7.7,-8.8,-10.6,-12.3,-14.2,-16.4,-18.5,-21,-23.5])
yreelle=ymes*2/100
t=np.linspace(0,1/4,16)
```

Présentation de la boucle for

Comme précédemment, la boucle for permet de faire défiler les positions successives (valeurs de la liste yreelle) et le tracé des vecteurs s'effectue avec la fonction plt.arrow qui a pour arguments successifs :

- abscisse de départ de la flèche : ici, 0 ;
- ordonnée de départ de la flèche : ici, yreelle[i], c'est-à-dire le ième élément de la liste des yreelle ;
- longueur horizontale de la flèche, ici 0 ;
- longueur verticale de la flèche : ici, (yreelle[i+1]- yreelle[i])/(t[i+1]-t[i]), c'est-à-dire la vitesse verticale lors du ième pas (on assimile ici la vitesse en i au déplacement entre les positions i et i+1 rapporté à l'intervalle de temps correspondant, assimilation qui n'est pas la plus précise).

```
y = 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])
y_reelle = y * 2 / 100
t = np.linspace(0, 1/4, 16)
n = np.arange(len(t) - 1)
x = 0 * t
for i in n :
    plt.arrow(0, y_reelle[i], 0, (y_reelle[i+1] - y_reelle[i]) / (t[i+1] - t[i]), head_width = 0.002)
plt.xlim(-0.02, 0.02)
plt.ylim(-5, 0)
```



Conclusion

Le rendu est peu lisible et difficilement exploitable. L'activité suivante permettra, en utilisant le même code pour l'étude d'une trajectoire bidimensionnelle, de proposer un mode de tracé plus visible du vecteur vitesse.

Retrouvez éducol sur :

