

## Utiliser Matplotlib

### Tracer une courbe avec Matplotlib

On s'intéresse à la trajectoire d'un ballon au cours du temps. Pour cela, on dispose de coordonnées en abscisses et en ordonnées de la position du ballon toutes les 0,08 secondes.

Les données sont fournies dans le tableau suivant :

|   |      |      |      |      |      |      |      |      |      |      |      |      |      |
|---|------|------|------|------|------|------|------|------|------|------|------|------|------|
| X | 0,00 | 0,43 | 0,87 | 1,28 | 1,68 | 2,06 | 2,40 | 2,76 | 3,09 | 3,41 | 3,74 | 4,07 | 4,36 |
| Y | 0,00 | 0,45 | 0,82 | 1,10 | 1,28 | 1,43 | 1,49 | 1,50 | 1,44 | 1,30 | 1,14 | 0,90 | 0,62 |

Pour représenter ces données en Python, nous allons utiliser le module Matplotlib que l'on importe à l'aide la commande suivante, à entrer en début de programme : `from matplotlib.pyplot import *`.

Les données doivent être entrées dans deux listes X et Y, dont le contenu doit être défini entre crochets :

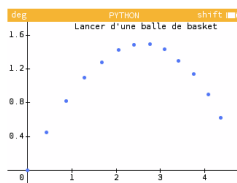
```
X = [0, 0.43, 0.86, 1.28, 1.68, 2.06, 2.40, 2.77, 3.09, 3.41, 3.75, 4.07, 4.36]
```

```
Y = [0, 0.45, 0.82, 1.10, 1.28, 1.44, 1.49, 1.50, 1.44, 1.31, 1.14, 0.90, 0.62]
```

On peut maintenant utiliser la commande `scatter(X, Y)` afin de faire apparaître le nuage de points utilisant la liste X en abscisses et Y en ordonnées.

On peut aussi remplacer par la commande `plot(X, Y)` afin de relier les points entre eux. Dans les deux cas, le programme doit être terminé par la commande `show()` afin de faire apparaître les points ou la courbe à l'écran.

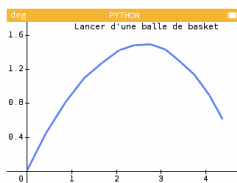
On pourra aussi, par exemple, ajouter une légende à l'aide de `text(x, y, "texte")` qui fait apparaître le "texte" au point de coordonnées (x; y)



```
from math import *
from matplotlib.pyplot import *

X=[0,0.43,0.87,1.28,1.68,2.06,2.40,2.76,3.09,3.41,3.74,4.07,4.36]
Y=[0,0.45,0.82,1.10,1.28,1.43,1.49,1.50,1.44,1.30,1.14,0.90,0.62]

scatter(X,Y)
text(1,1.6,"Lancer d'une balle de basket")
show()
```



```
from math import *
from matplotlib.pyplot import *

X=[0,0.43,0.87,1.28,1.68,2.06,2.40,2.76,3.09,3.41,3.74,4.07,4.36]
Y=[0,0.45,0.82,1.10,1.28,1.43,1.49,1.50,1.44,1.30,1.14,0.90,0.62]

plot(X,Y)
text(1,1.6,"Lancer d'une balle de basket")
show()
```

## Tracer les vecteurs vitesses avec Matplotlib

En affichant les résultats sous forme de nuage de points avec **scatter**, nous allons maintenant faire apparaître les vecteurs vitesses tout au long du mouvement.

Pour cela, nous allons utiliser la commande **arrow(x, y, dx, dy)** qui fait apparaître un vecteur de coordonnées (dx; dy) dont l'origine est le point de coordonnées (x; y).

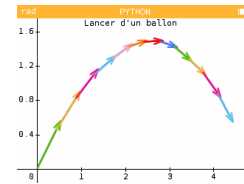
Nous disposons déjà des coordonnées de chaque point à l'aide des listes X et Y. Pour désigner, par exemple, le terme de rang i dans la liste X, on utilisera la commande **X[i]** (le premier terme de la liste étant de rang 0). Ainsi, pour tracer le vecteur de rang [i], on utilisera les coordonnées des points de rang [i] et [i+1].

```
from math import *
from matplotlib.pyplot import *

X=[0,0.43,0.87,1.28,1.68,2.06,2.40,2.76,3.09,3.41,3.74,4.07,4.36]
Y=[0,0.45,0.82,1.10,1.28,1.43,1.49,1.50,1.44,1.30,1.14,0.90,0.62]
dt=0.08

scatter(X,Y)
for t in range (len(X)-1):
    arrow(X[t],Y[t],(X[t+1]-X[t])/(10*dt),(Y[t+1]-Y[t])/(10*dt))

text(1,1.6,"Lancer d'un ballon")
show()
```



Pour contrôler la longueur des vecteurs, on pourra appliquer un facteur d'échelle, d'où la division par 10 dans le code ci-dessus.