

## TP 20 : Étude du mouvement



### Objectifs :

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



En 1604, Galilée pose les bases de la mécanique en étudiant le mouvement de chute des corps. Pour ses travaux, il ne dispose pas d'instruments précis de mesure du temps. Depuis, les avancées technologiques et numériques permettent une analyse précise des mouvements. Comment analyser le mouvement de la chute d'un corps avec les outils numériques actuels ?

### I- Réaliser et exploiter une chronophotographie d'un système en mouvement

#### Chronophotographie :

Succession de photographies prises à intervalles de temps réguliers, permettant de décomposer un mouvement.

#### Matériel :

Vidéo : Chutelibre  
Ordinateur avec le logiciel Aviméca  
Notice Aviméca en annexe

- 1- Émettre une hypothèse sur la nature du mouvement de chute d'une balle lâchée sans vitesse initiale dans le référentiel terrestre.
- 2- Réaliser le protocole suivant :
  - Ouvrir Aviméca.
  - Suivre les instructions de l'annexe pour réaliser le pointage de la vidéo intitulée chutelibre.
  - Imprimer la trajectoire.
- 3- Numéroté les positions successives de la balle.  $M_0, M_1, M_2, \dots$
- 4- Représenter, les vecteurs vitesses de la balle  $\vec{v}_2, \vec{v}_3$  et  $\vec{v}_4$ .
- 5- Vérifier l'hypothèse émise sur la nature du mouvement de la balle.
- 6- Comparer les caractéristiques des vecteurs vitesses représentés.

## II- Représenter les vecteurs vitesse d'un système à l'aide d'un langage de programmation

Le mouvement d'un système modélisé par un point M peut être analysé grâce à l'utilisation d'un langage de programmation Python. Dans le script du programme utilisé pour tracé des vecteurs vitesse, les premières instructions consistent à représenter un nuage de points représentant les positions pour ensuite représenter les vecteurs vitesse.

Voici un screenshot du programme :

```

1 import matplotlib.pyplot as plt
2 # Positions
3 # Tableaux contenant les différentes abscisses et ordonnées des positions de la balle
4 X = [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ]
5 Y = [ 0.0018, -0.021, -0.062, -0.12, -0.19, -0.28, -0.38, -0.50, -0.64, -0.79, -0.96]
6 # On trace la trajectoire
7 #Représentation des positions de la balle, le symbole 'r+' permet de représenter les positions de la balle avec des croix rouges
8 plt.plot(X,Y,'r+' )
9 # Paramètres du graphe (taille, légendes, titres des axes et du graphique)
10 plt.axis([-1,1,-1,0], 'equal')
11 plt.xlabel(" x (en m)")
12 plt.ylabel(" y (en m)")
13 plt.title("Trajectoire de la balle dans le référentiel Terrestre")
14 # Afficher la trajectoire de la balle
15 plt.show()
16 #Tracé des vecteurs vitesse
17 # Nombre de positions adoptées par le système
18 N=
19 # Durée (en seconde) entre deux positions successives. Attention: Ne pas mettre de virgule mais un point.
20 dt=
21 # Boucle permettant de ...
22 for k in range(N-1) :
23     |   Vy=(Y[k+1]- Y[k])/dt
24     |   plt.quiver(X[k],Y[k],0,0.4*Vy, color="green",scale=12,angles='xy',scale_units='xy')
25 #Affichage
26 plt.show()

```

### 1- Tes missions :

- Prendre connaissance du script.
  - Aller sur internet et ouvrir le lien suivant qui permet d'ouvrir une console puis d'exécuter un script : <https://www.lelivrescolaire.fr/console-python>
  - Ouvrir et copier le script (format .odt).
  - Coller le script dans la partie console.
- 2- Compléter les valeurs manquantes de N et dt à l'aide des valeurs obtenues lors du pointage.
  - 3- Que permettent de faire les lignes 22 à 24 ?
  - 4- Exécuter le script. Pourquoi le vecteur vitesse n'est-il pas tracé pour la dernière position de la balle.

## Script

```
import matplotlib.pyplot as plt
# Positions
# Tableaux contenant les différentes abscisses et ordonnées des positions de la balle
X = [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ]
Y = [ 0.0018, -0.021, -0.062, -0.12, -0.19, -0.28, -0.38, -0.50, -0.64, -0.79, -0.96]
# On trace la trajectoire
#Représentation des positions de la balle, le symbole 'r+' permet de représenter les positions de la balle
avec des croix rouges
plt.plot(X,Y,'r+' )
# Paramètres du graphe (taille, légendes, titres des axes et du graphique)
plt.axis([-1,1,-1,0],'equal')
plt.xlabel(" x (en m)")
plt.ylabel(" y (en m)")
plt.title("Trajectoire de la balle dans le référentiel Terrestre")
# Afficher la trajectoire de la balle
plt.show()
#Tracé des vecteurs vitesse
# Nombre de positions adoptées par le système
N=
# Durée (en seconde) entre deux positions successives. Attention : Ne pas mettre de virgule mais un point.
dt=
# Boucle permettant de ....
for k in range(N-1) :
    Vy=(Y[k+1]- Y[k])/dt
    plt.quiver(X[k],Y[k],0,0.4*Vy, color="green",scale=12,angles='xy',scale_units='xy')
#Affichage
plt.show()
```