

Thème : Mouvements et interactions
P6 : description d'un mouvement
Activité 3 : étudier le mouvement de chute d'un corps

Objectif : utiliser un langage de programmation pour représenter des vecteurs vitesse

Document 0 : Télécharger éditeur PYTHON (pour ouvrir des fichiers.py)
Sous WINDOWS :télécharger edupython, gratuit

<http://edupython.tuxfamily.org>

onglet téléchargement

SOUS MAC : télécharger IDLE, gratuit

<https://www.python.org/downloads/>

Download the latest version for Mac OS X

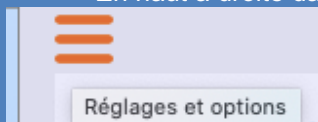
Document 1 : simulation de chute verticale + protocole expérimental

https://www.hatier-clic.fr/2019/01/9782401020658/Newton_seconde/index.htm

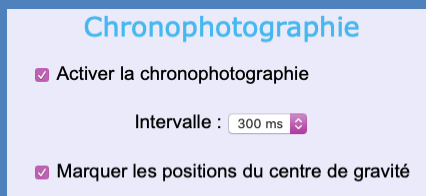
- Aller dans « activités »



- Cliquer sur « étudier le mouvement de la chute d'un corps »
- En haut à droite dans réglages et options :



- Activer la chronophotographie , intervalle 300ms.



- Cliquer sur tableaux de valeurs (pour avoir les valeurs de x et y en fonction du temps)
- Cliquer ensuite sur exporter > copier les positions pour python (cela vous permettra d'avoir directement les valeurs de x, y et t pour python (sinon, vous pourrez les copier manuellement).
- Lancer la simulation.

Document 2 : le code, nommé vecteurs_vitesses.py

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 #Positions du point M
5 x=np.array([..., ..., ..., ...])
6 y=np.array([..., ..., ..., ...])
7
8 #Tracé de la chronophotographie
9 plt.plot(x,y, 'o', markersize=4)
10 plt.xlabel("x (en m)")
11 plt.ylabel("y (en m)")
12 plt.title("Chronophotographie du mouvement")

```

Tableaux contenant les abscisses des positions du système et les ordonnées des positions du système.

Représentation graphique des positions du point M (x ; y) au cours du temps par le symbole 'o'. « markersize » règle la taille du symbole.

On peut ensuite tracer des vecteurs vitesse en utilisant les instructions suivantes.

```

14 #Tracé des vecteurs vitesse
15 N=...
16 dt=...
17 for k in range(0, N-1) :
18     Vy=(y[...]-y[...])/dt
19
20     echelle=0.2
21     Vy=Vy*echelle
22
23     plt.quiver(x[...], y[...], 0, ...,
24               color="red", scale=1, scale_units='xy')
25 #Affichage
26 plt.show()

```

Nombre total de positions adoptées par le système.

Durée (en secondes) entre deux positions successives.

① 1^{re} itération (k = 0) : $V_y = (y[1] - y[0]) / dt$
 2^e itération (k = 1) : $V_y = (y[2] - y[1]) / dt$
 3^e itération (k = 2) : $V_y = (y[...] - y[...]) / dt$...
 (k+1)^e itération : $V_y = (y[...] - y[...]) / dt$...

Ajustement de la longueur du vecteur vitesse.

② (k+1)^e itération : $plt.quiver(x[...], y[...], 0, ...)$

Les instructions dans la boucle **for** s'exécutent une première fois pour $k = 0$ (1^{re} itération*), puis pour $k = 1$ (2^e itération), puis $k = 2$ (3^e itération) jusqu'à $k = N - 2$ ((N - 1)^e itération).

À chaque itération un vecteur vitesse est tracé : \vec{v}_1 (1^{re} itération), puis \vec{v}_2 (2^e itération) puis \vec{v}_3 (3^e itération) jusqu'à \vec{v}_{N-1} ((N - 1)^e itération).

La fonction `quiver(xM, yM, Vx, Vy)` permet de tracer un vecteur \vec{v} dont l'origine est M(x_M ; y_M) et dont les coordonnées sont $\begin{pmatrix} V_x \\ V_y \end{pmatrix}$. Les arguments facultatifs `color="red"`, `scale=1` et `scale_units='xy'` dans la fonction permettent la mise en forme du vecteur.

VOCABULAIRE

Itération : répétition d'un calcul ou d'un bloc d'instructions informatiques.

Questions

- Ouvrir le simulateur, document 1 : quel est le système étudié ? {balle}
- Quel est le référentiel d'étude ? Terrestre.
- Compléter les lignes 5 et 6 du code grâce aux coordonnées du point M obtenues grâce à la simulation (vous pouvez soit faire clic-droit>coller soit les ajouter à la main) .
Lignes 5 et 6 : le point M1 a pour coordonnées M₁(10 ; 20) , M₂(10 ; 19,56)

On remplit des listes :

```

4 #Positions du point M
5 x=[10,10,10,10,10,10,10]# a compléter
6 y=[20,19.56,18.24,16.04,12.96,9.03,4.25] # a compléter
7

```

- Compléter les lignes 15 et 16 du code :
Ligne 15 : combien y a-t'il de positions adoptées par le système ?
N= 7 positions , de M₁ jusqu'à M₇

Ligne 16 : combien vaut Δt , en secondes ? $\Delta t = 300ms = 0,3 s$

```
N= 7 # a compléter
dt=0.3 # a compléter
```

5. A) Compléter l'encadré ① du document 2.

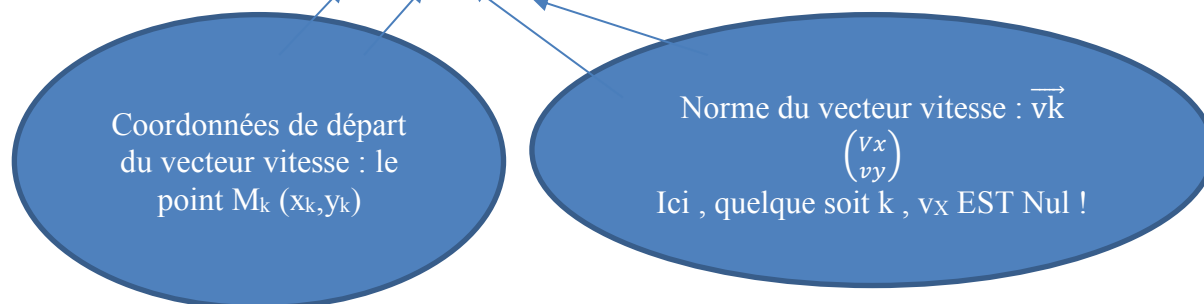
A la (k+1)^e itération : $V_y = (y[k+1] - y[k]) / dt$

B) Quel vecteur aura été tracé lorsque toutes les instructions de la boucle auront été exécutées ?

6. Compléter la ligne 18 du code.

7. A) Compléter l'encadré ② du document 2.

(k+1)^e itération : `plt.quiver(x[k], y[k], 0, Vy)`



B) Compléter la ligne 23 du code (aide fin du document 2).

Lignes 18 et 23

```
17 for k in range(0, N-1) :
18     Vy=(y[k+1]- y[k])/dt # a compléter
19
20     echelle=0.02
21     Vy=Vy*echelle
22
23     plt.quiver(x[k],y[k],0,Vy, color="red",scale=1,scale_units='xy')# a compléter
24
```

8. Exécuter le code.

9. Pourquoi les instructions de la boucle for ne peuvent pas être exécutées une N^e-ème fois ?

Les instructions ne peuvent pas être réalisées une N^e fois car il n'y a pas de position N+1 pour calculer la valeur de la vitesse en position N.