

Quelques exemples d'utilisation de Python: exercice de chute libre

Entrée [6]:

```
## Importation des bibliothèques pour Le notebook
import numpy as np
import matplotlib.pyplot as plt
```

Entrée [22]:

```
## POUR MODIFIER UNE CELLULE, DOUBLE-CLIQUE DESSUS
## APPUIES SUR Exécuter POUR CONFIRMER LES MODIFICATIONS

## Tu peux modifier toutes Les cellules à partir de là =>
```

Exercice 1. Trajectoire de la chute libre

#On définit les paramètres de départ :

```
#-1) accélération de la pesanteur : g = 9.18

#-2) On note la position toutes les secondes : dt = 0.1

#-3) Hauteur de l'objet avant qu'il tombe (en mètres) : hauteur_0 = 15

#-4) La vitesse de l'objet est nulle au début de sa chute : vitesse_0 = 0
```

Entrée [9]:

```
# Données de l'exercice
```

```
g = 9.18

dt = 0.1

hauteur_0 = 15

vitesse_0 = 0
```

Entrée [17]:

```
# On écrit Le programme qui va calculer la trajectoire de l'objet qui tombe
```

```
hauteur = hauteur_0

vitesse = vitesse_0

i = 0
temps = 0

while(hauteur > 0) : #tant que la hauteur de l'objet est plus grande que 0 ( si la hauteur vaut 0, il touche le sol )

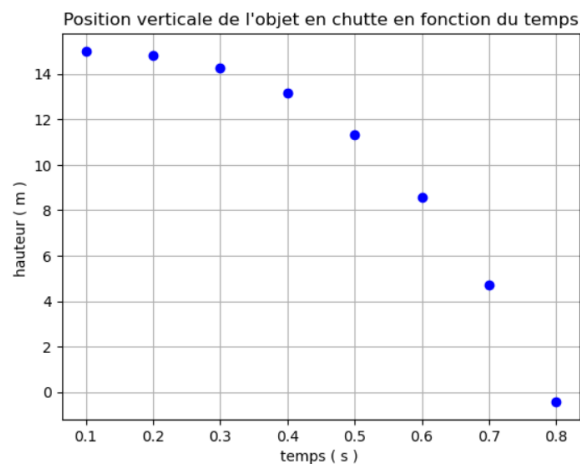
    vitesse = vitesse - g * dt
    hauteur = hauteur + vitesse * temps

    i = i + 1
    temps = temps + dt

plt.plot(temps, hauteur , "bo") # fonction python pour tracer une courbe
# essaie de remplacer "bo" par "ro" ou "yo" ou "go" pour voir...
# essaie aussi de remplacer "bo" par "b+" ou "r+" ect.
```

```
plt.title("Position verticale de l'objet en chute en fonction du temps") #titre
plt.xlabel("temps ( s ) ")
plt.ylabel("hauteur ( m ) ")
plt.grid(True)
print("l'objet touche le sol en ",round(temps,2)," secondes")
print("calcul en ",i," étapes espacées de ",dt," secondes")
```

l'objet touche le sol en 0.8 secondes
calcul en 8 étapes espacées de 0.1 secondes



```
Entrée [ ]: ## On peut changer Les paramètres de départ

## -1) change dt

dt = 0.01
## Que se passe-t-il? On voit plus ou moins de points ?
```

```
Entrée [20]: ## On peut changer Les paramètres de départ

## -2) change hauteur_0

hauteur_0 = 30

## L'objet tombe en combien de temps ? Le tems de chute a augmenté ?
```

Tu peux noter ici tes résultats d'expérience :

Si hauteur_0 = 15 mètres Alors l'objet tombe en ____ secondes.

Si hauteur_0 = 30 mètres Alors l'objet tombe en ____ secondes.

Si hauteur_0 = 1500 mètres Alors l'objet tombe en ____ secondes.

Exercice 2. Vitesse lors de la chute

#On définit les mêmes paramètres de départ :

```
#-1) accélération de la pesanteur : g = 9.18

#-2) On note la position toutes les secondes : dt = 0.1

#-3) Hauteur de l'objet avant qu'il tombe (en mètres ): hauteur_0 = 15

#-4) La vitesse de l'objet est nulle au début de sa chute : vitesse_0 = 0

.
```

```
Entrée [31]: # Données de l'exercice
```

```
g = 9.18

dt = 0.1

hauteur_0 = 15

vitesse_0 = 0
```

```
Entrée [36]: # On écrit Le programme qui va calculer La trajectoire de l'objet qui tombe
```

```
hauteur = hauteur_0
position_h = [hauteur]
```

```
vitesse = vitesse_0
vitesse_v = [vitesse]

i = 0
temps = 0

while(hauteur > 0 ) : #tant que La hauteur de l'objet est plus grande que 0 ( si la hauteur vaut 0, il touche le sol )

    vitesse = vitesse -g * dt
    vitesse_v = vitesse_v + [vitesse]

    hauteur = hauteur + vitesse * dt
    position_h = position_h + [hauteur]

    i = i + 1

temps = np.linspace(0,i,i+1)*dt
print("l'objet touche le sol en ",round(temps[i],2)," secondes")
print("calcul en ",i," étapes espacées de ",dt," secondes")

l'objet touche le sol en 1.8 secondes
calcul en 18 étapes espacées de 0.1 secondes
```

```
Entrée [42]: # On s'intéresse à La vitesse de l'objet
```

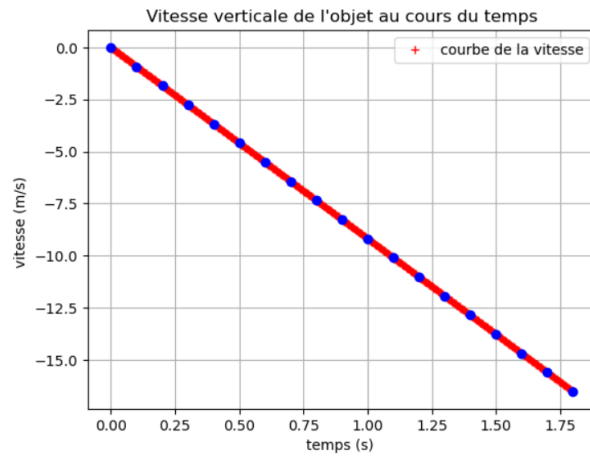
```
nb_pts = int(1000/temps[i])
temps_lissé = np.linspace(0,temps[i],nb_pts)
vitesse_lissé = - g * temps_lissé + vitesse_0

vitesse = - g * temps + vitesse_0

plt.plot(temps_lissé, vitesse_lissé, "r+", lw=0.5, label="courbe de la vitesse")
plt.legend()
plt.title("Vitesse verticale de l'objet au cours du temps")
plt.xlabel("temps (s) ")
plt.ylabel("vitesse (m/s) ")
plt.grid(True)

plt.plot(temps, vitesse_v, "bo",label="vitesse calculée")
```

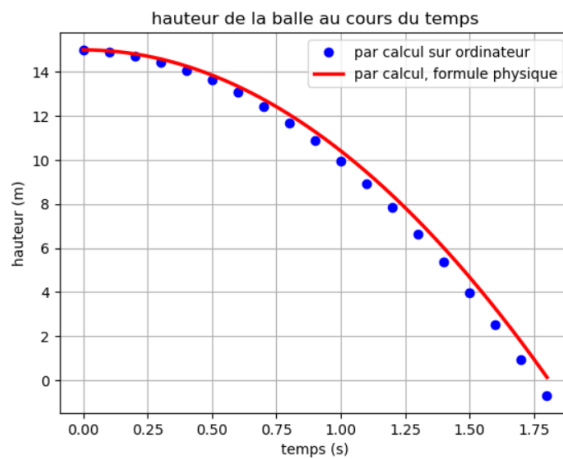
```
Out[42]: [<matplotlib.lines.Line2D at 0x18ba51c03d0>]
```



```
Entrée [46]: # On représente à nouveau la trajectoire de l'objet :
plt.plot(temps, position_h, "bo", label="par calcul sur ordinateur")

nb_pts = int(11000/temps[i])
temps_lissé = np.linspace(0, temps[i], nb_pts)
hauteur_lissé = -1/2 * g * temps_lissé**2 + vitesse_0 * temps_lissé + hauteur_0

plt.plot(temps_lissé, hauteur_lissé, "-r", lw=2.5, label="par calcul, formule physique")
plt.legend()
plt.title("hauteur de la balle au cours du temps")
plt.xlabel("temps (s)")
plt.ylabel("hauteur (m)")
plt.grid(True)
```



Voilà c'est fini !

Maintenant, tu sais programmer en python et tu sais comment la chute d'objets fonctionne physiquement.