VECTEURS VITESSE ET VARIATION DE VITESSE

Les données sont obtenues à partir de l'exploitation d'une chronophotographie du lancer d'un ballon.

Programme Python

Méthode 1

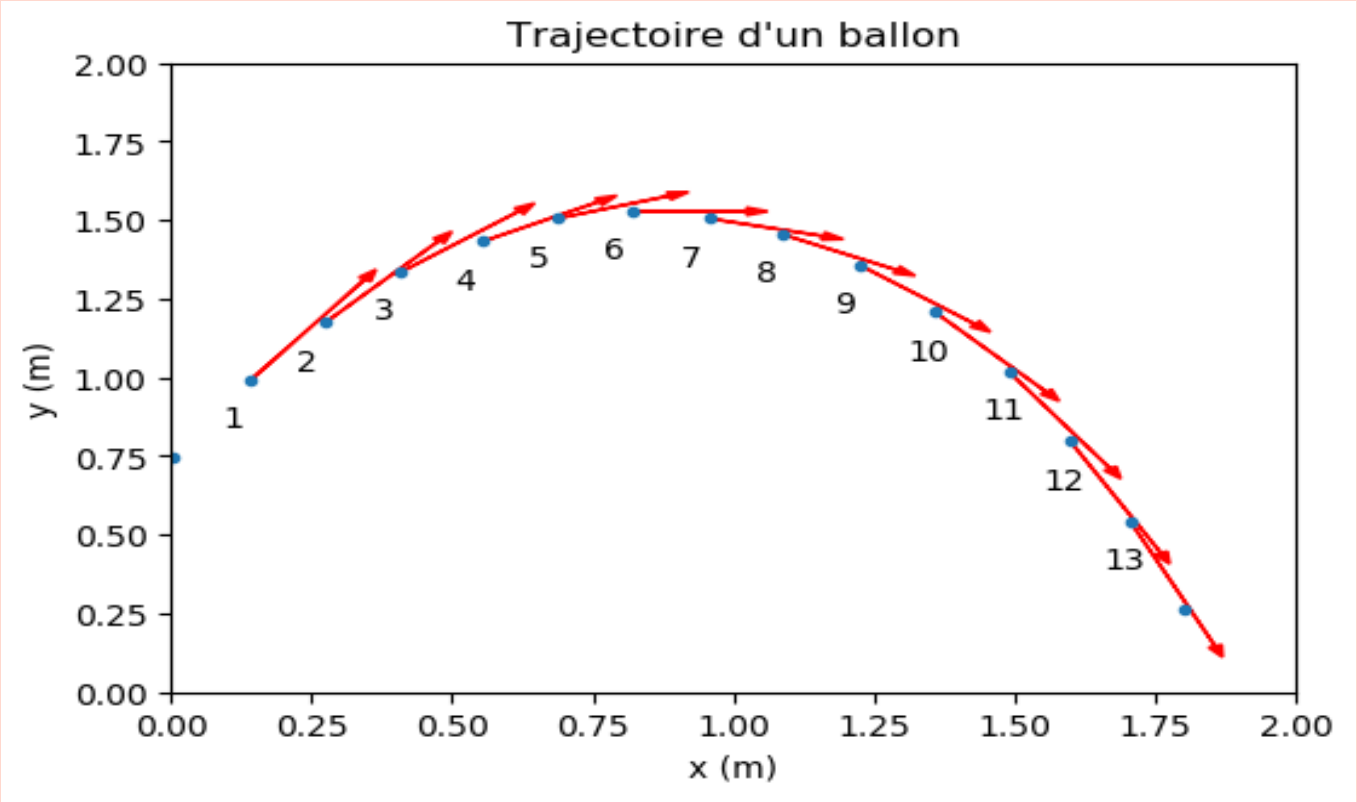
La fonction `matplotlib.pyplot.arrow()` permet de tracer des flèches pour illustrer les graphiques.

```
import numpy as np
import matplotlib.pyplot as plt
# Données
dt = 0.0667
x = np.array([0.003, 0.141, 0.275, 0.410, 0.554, 0.686, 0.820, 0.958, 1.089, 1.227, 1.359, 1.490, 1.599,
1.705, 1.801])
y = np.array([0.746, 0.990, 1.175, 1.336, 1.432, 1.505, 1.528, 1.505, 1.454, 1.355, 1.207, 1.018, 0.797,
0.544, 0.266])
# Calculs des vecteurs vitesses
N = len(x)          # Nombre de points de mesures
vx = np.zeros(N)    # Initialisation d'un tableau vide
vy = np.zeros(N)    # Idem
for i in range(1, N-1):
    vx[i]=(x[i+1]-x[i-1])/(2*dt)
    vy[i]=(y[i+1]-y[i-1])/(2*dt)
# Calcul des vitesses
v = np.sqrt(vx**2+vy**2)
plt.plot(x, y, '.')
plt.xlabel("x (m)")
```

```
plt.xlim(0, 2)
plt.ylabel("y (m)")
plt.ylim(0, 2)
plt.title("Trajectoire d'un ballon")
for i in range(1,N-1):
    plt.arrow(x[i], y[i], vx[i]/10, vy[i]/10, head_width=0.025, color='r')
    plt.annotate(i, (x[i]-0.05,y[i]-0.15))
    print('Point', i, '-> v=', round(v[i],2), " m/s")
plt.show()
```

Résultats :

```
Point 1 -> v= 3.81  m/s
Point 2 -> v= 3.29  m/s
Point 3 -> v= 2.84  m/s
Point 4 -> v= 2.43  m/s
Point 5 -> v= 2.12  m/s
Point 6 -> v= 2.04  m/s
Point 7 -> v= 2.09  m/s
Point 8 -> v= 2.31  m/s
Point 9 -> v= 2.74  m/s
Point 10 -> v= 3.2  m/s
Point 11 -> v= 3.56  m/s
Point 12 -> v= 3.9  m/s
Point 13 -> v= 4.26  m/s
```



Méthode 2

La fonction `matplotlib.pyplot.quiver()` a l'avantage d'être itérative. Elle permet de tracer un champ de vecteur.

```
import numpy as np
import matplotlib.pyplot as plt
# Données
dt = 0.0667
x = np.array([0.003, 0.141, 0.275, 0.410, 0.554, 0.686, 0.820, 0.958, 1.089, 1.227, 1.359, 1.490, 1.599, 1.705, 1.801])
y = np.array([0.746, 0.990, 1.175, 1.336, 1.432, 1.505, 1.528, 1.505, 1.454, 1.355, 1.207, 1.018, 0.797, 0.544, 0.266])
# Calculs des vecteurs vitesses
N = len(x)          # Nombre de points de mesures
vx = np.zeros(N)    # Initialisation d'un tableau vide
```

```
vy = np.zeros(N)    # Idem
for i in range(1, N-1):
    vx[i]=(x[i+1]-x[i-1])/(2*dt)
    vy[i]=(y[i+1]-y[i-1])/(2*dt)
# Calcul des vitesses
v = np.sqrt(vx**2+vy**2)
plt.plot(x, y, ".")
plt.xlabel("x (m)")
plt.xlim(0, 2)
plt.ylabel("y (m)")
plt.ylim(0, 2)
plt.title("Trajectoire d'un ballon")
plt.quiver(x, y, vx, vy, angles='xy', scale_units='xy', scale=10, color='red', width=0.005)
for i in range(1, N-1):
    plt.annotate(i, (x[i]-0.05,y[i]-0.15))
    print('Point', i, '-> v=', round(v[i],2), " m/s")
plt.show()
```

Résultats:

```
Point 1 -> v= 3.81  m/s
Point 2 -> v= 3.29  m/s
Point 3 -> v= 2.84  m/s
Point 4 -> v= 2.43  m/s
Point 5 -> v= 2.12  m/s
Point 6 -> v= 2.04  m/s
Point 7 -> v= 2.09  m/s
```

Point 8 -> v= 2.31 m/s

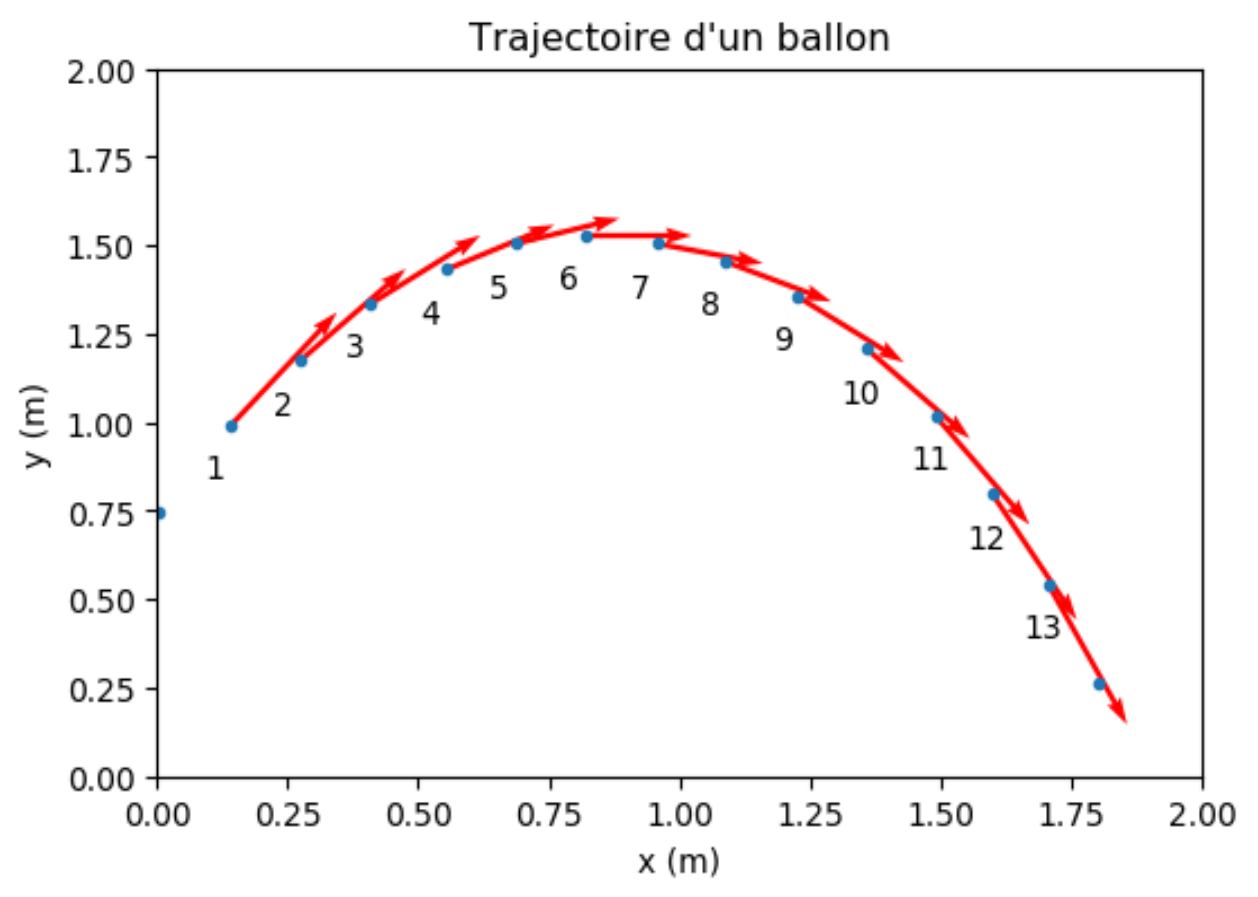
Point 9 -> v= 2.74 m/s

Point 10 -> v= 3.2 m/s

Point 11 -> v= 3.56 m/s

Point 12 -> v= 3.9 m/s

Point 13 -> v= 4.26 m/s



Mouvement d'un point : vecteur variation vitesse

Programme Python

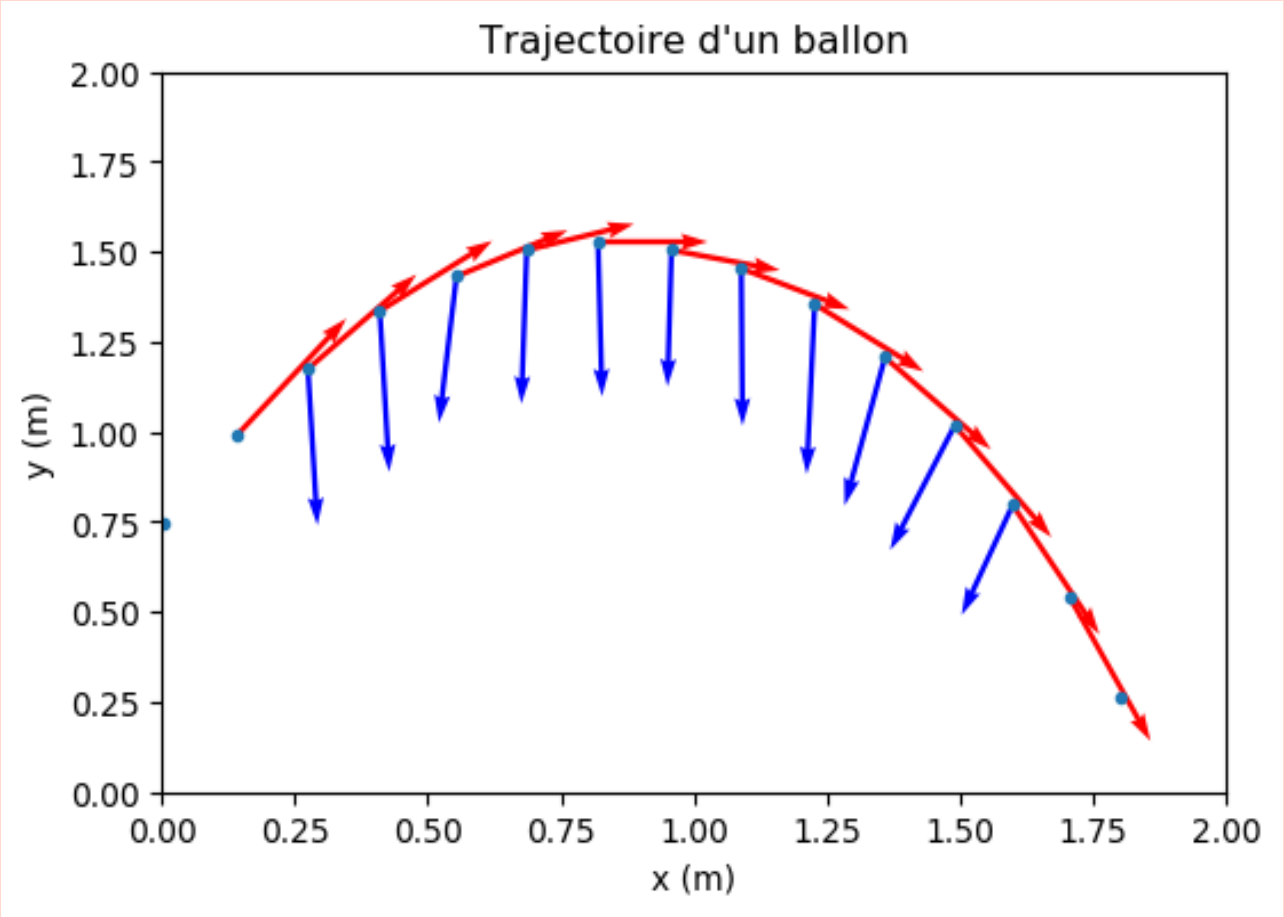
```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
plt.rcParams['figure.dpi'] = 100
dt = 0.066
x = np.array([0.003, 0.141, 0.275, 0.410, 0.554, 0.686, 0.820, 0.958, 1.089, 1.227, 1.359, 1.490, 1.599,
1.705, 1.801])
y = np.array([0.746, 0.990, 1.175, 1.336, 1.432, 1.505, 1.528, 1.505, 1.454, 1.355, 1.207, 1.018, 0.797,
0.544, 0.266])
N = x.size
vx = np.zeros(N)
vy = np.zeros(N)
for i in range(1, N-1):
    vx[i] = (x[i+1]-x[i-1])/(2*dt)
    vy[i] = (y[i+1]-y[i-1])/(2*dt)
#print(vx.round(2))
#print(vy.round(2))
dvx = np.zeros(N)
dvy = np.zeros(N)
for i in range(2, N-2):
    dvx[i] = vx[i+1]-vx[i-1]
    dvy[i] = vy[i+1]-vy[i-1]
#print(dvx.round(2))
#print(dvy.round(2))
plt.plot(x, y, ".")
plt.quiver(x, y, vx, vy, angles='xy', scale_units='xy', scale=10, color='red', width=0.005)
plt.quiver(x, y, dvx, dvy, angles='xy', scale_units='xy', scale=3, color='blue', width=0.005)
plt.xlabel("x (m)")
```

```
plt.xlim(0, 2)
plt.ylabel("y (m)")
plt.ylim(0, 2)
plt.title("Trajectoire d'un ballon")
plt.show()
```

Résultats:



La fonction `quiver()` permet de tracer un vecteur ou un champ de vecteur.