Partie VI : Mécanique
TP

Correction – TP 7 : chute d'une bille dans un fluide visqueux

1 - C'est le PFD qui mène facilement à :

$$m\frac{\mathrm{d}v}{\mathrm{d}t} + 6\pi\eta R v = \frac{4}{3}\pi g R^3 \left(\rho_b - \rho_f\right).$$

Ensuite, la vitesse limite est telle que $\frac{\mathrm{d}v}{\mathrm{d}t} = 0$, donc il reste à isoler v ci-dessus. On trouve bien : $\boxed{v_{\text{lim}} = \frac{2}{9}\frac{gR^2}{\eta}\left(\rho_b - \rho_f\right).}$

En exploitant les formules théoriques, on s'attend à (avec $T = 30\,\text{cm}/v_{\text{lim}}$; avec une viscosité de $1\,\text{Pa}\cdot\text{s}$) :

R	v_{lim}	T (s) (pour $d = 30\,\text{cm}$)	5τ et $v_{\text{lim}} \times 5\tau$
2 mm	5,5 cm/s	5,4 s	35 ms et 2 mm
3 mm	12 cm/s	2,4 s	75 ms et 10 mm
4 mm	22 cm/s	1,4 s	135 ms et 30 mm

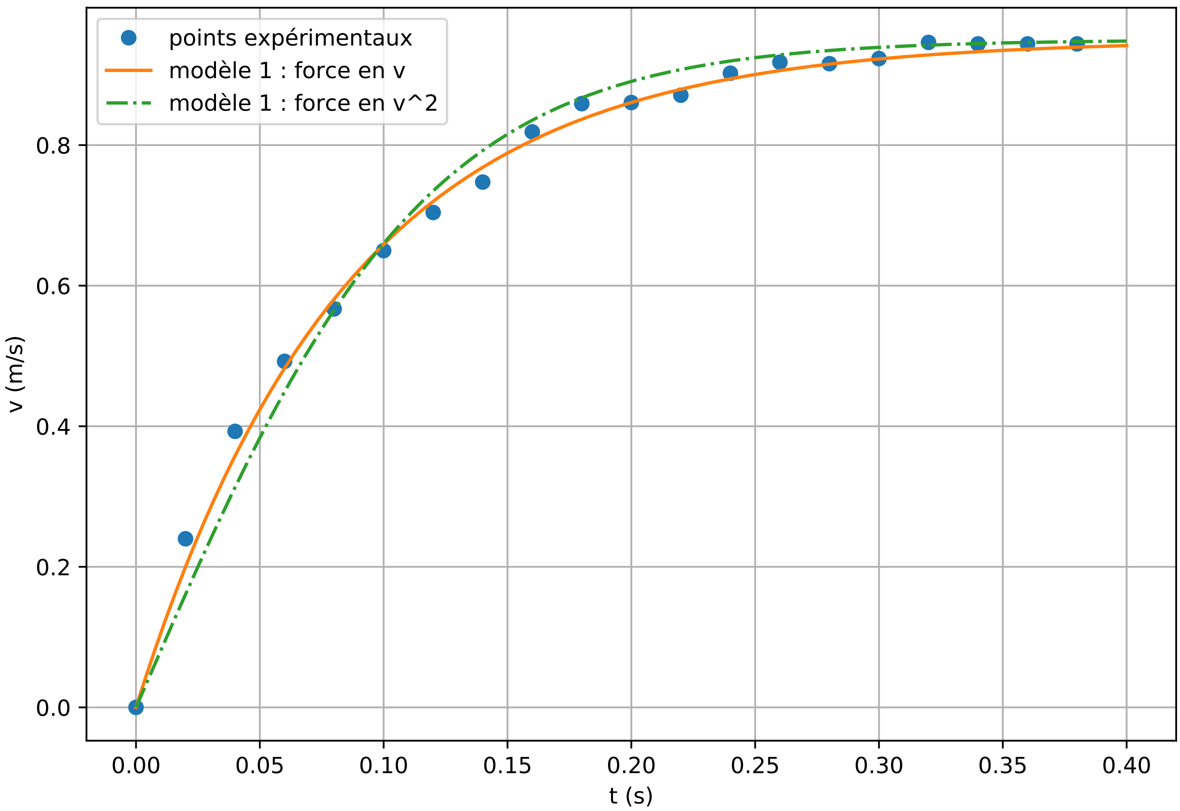
Remarque : la constante de temps de l'équation différentielle du premier ordre est

$$\tau = \frac{m}{6\pi\eta R} = \frac{2}{9}\frac{R^2\rho_b}{\eta}.$$

On atteint 95% de v_{lim} en 5τ, cf tableau ci-dessus pour les valeurs. (Les étudiants peuvent faire ce calcul.)

Première expérience : mesure de la viscosité

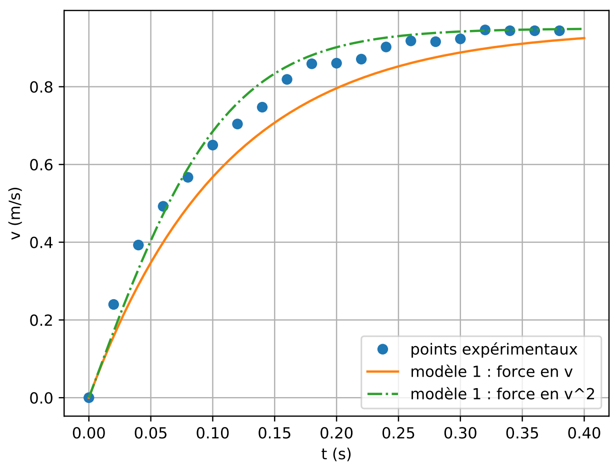
Seconde expérience : exploitation d'un enregistrement vidéo



Le meilleur accord pour le modèle en v^2 semble être autour de $a = 9\,\mathrm{m}^{-1}$.

On constate (graphe ci-dessus) que le modèle en v correspond mieux dans sa forme aux données expérimentales.

Remarque : Néanmoins, les paramètres expérimentaux ne correspondent pas tout à fait à τ et v_{lim}. On peut en effet écrire $\dot{v} = g_{\mathrm{eff}}(1 - v/v_{\mathrm{lim}})$ ou $\dot{v} = g_{\mathrm{eff}}(1 - v^2/v_{\mathrm{lim}}^2)$ avec $g_{\mathrm{eff}} = g(1 - \rho_{\mathrm{fluide}}/\rho_{\mathrm{bille}}) = 8{,}61\,\mathrm{m/s^2}$ et $v_{\mathrm{lim}} = 0{,}95\,\mathrm{m/s}$ d'après le graphe, et on obtient le graphe suivant :



Le code complété :

```
#------------ PARTIE 1 : DONNEES EXPERIMENTALES -------------#

import numpy as np
import matplotlib.pyplot as plt

# position relevée sur la vidéo (metres)
z_exp = (np.array([2.3,3.7,6.9,11.23,16.34,22.1,28.8,35.6,43.13,51.3,59.6,67.8,
76.3,85.1,93.9,102.66,111.6,120.8,129.7,138.9,147.8])-2.3)*242/116 * 1.e-3
dt_exp = 1/50    # temps entre deux images du film (seconde)

#-- à partir de z_exp et dt_exp, construction d'une liste des vitesses et des temps--#
t_exp = [0]
v_exp = [0]
for i in range(1,len(z_exp)-1):
    v_exp.append((z_exp[i+1]-z_exp[i-1])/(2*dt_exp))
    t_exp.append(i*dt_exp)

#--tracé--#
plt.figure(1)
plt.plot(t_exp,v_exp,'o',label='points expérimentaux')
plt.xlabel("t (s)")
plt.ylabel("v (m/s)")
plt.legend()
plt.grid()
plt.show()


#------------ PARTIE 2 : MODELE 1 : FORCE EN V -------------#
fin = 0.4        # durée simulation (secondes)
dt = fin/500     # pas de temps, assez court
nb_iterations = int(fin/dt)    # calcul du nombre total d'itérations nécessaires

vlim = 0.95      # vitesse limite mesurée sur le graphe expérimental (m/s)   A COMPLETER
tau  = 0.085     # constante de temps mesurée sur le graphe expérimental (s) A COMPLETER

# Initialisation des listes :
liste_t = [0]
liste_v = [0]
t = 0
v = 0

for i in range(nb_iterations):
    # on dispose des variables de l'itération précédente : t et v
    # et on calcule leurs nouvelles valeurs :
    v = v + dt*(vlim-v)/tau                                   # A COMPLETER
    t = t + dt

    # On ajoute ces nouvelles valeurs à chaque liste :
    liste_t.append(t)
    liste_v.append(v)

plt.plot(liste_t,liste_v,'-',label='modèle 1 : force en v')    # A COMPLETER
plt.legend()
plt.show()


#------------ PARTIE 3 : MODELE 2 : FORCE EN V^2 -------------#
a = 9   # paramètre en facteur de -v^2    # A AJUSTER AU MIEUX

# Initialisation des listes :
liste_t = [0]
liste_v = [0]
```

```python
t = 0
v = 0

for i in range(nb_iterations):
    # on dispose des variables de l'itération précédente : t et v
    # et on calcule leurs nouvelles valeurs :
    v = v + dt*(vlim**2-v**2)*a                               # A COMPLETER
    t = t + dt

    # On ajoute ces nouvelles valeurs à chaque liste :
    liste_t.append(t)
    liste_v.append(v)

plt.plot(liste_t,liste_v,'-.',label='modèle 1 : force en v^2')  # A COMPLETER
plt.legend()
plt.show()
```