

## Python | TD n°2 (D3)

- Mettre en œuvre la méthode d'Euler explicite afin de résoudre une ED d'ordre 1.
- Utiliser la fonction « odeint » de la bibliothèque « scipy.integrate » afin de résoudre une ED d'ordre 1 ou 2.

### Exercice n°1 - ED d'ordre 1, avec solution analytique

On considère une réaction chimique d'ordre 2. L'équation différentielle peut être résolue analytiquement (cf. cours de chimie).

$$v = -\frac{dC}{dt} = kC^2 \Rightarrow C(t) = \left(\frac{1}{C_0} + kt\right)^{-1}$$

Données :  $C_0 = 0,1 \text{ mol} \cdot \text{L}^{-1}$  et  $k = 0,1 \text{ mol}^{-1} \cdot \text{L} \cdot \text{min}^{-1}$

- 1) Écrire un script Python permettant de visualiser la solution analytique.
- 2) Résoudre l'ED à l'aide de la méthode d'Euler. Superposer les graphes.
- 3) Résoudre l'ED à l'aide de la fonction « odeint ». Superposer les graphes.

### Exercice n°2 - ED d'ordre 1, sans solution analytique

On considère l'équation différentielle suivante :

$$\frac{df}{dt} \cdot \exp\left(\frac{t}{\tau}\right) + \sin(\omega t) \cdot f^2(t) = t^2$$

Données :  $\tau = 5 \text{ s}$ ,  $\omega = 20 \text{ rad} \cdot \text{s}^{-1}$  et  $f(0) = 0$ .

- 1) Résoudre numériquement l'ED à l'aide de la méthode de votre choix.

### Exercice n°3 - ED d'ordre 2

On considère un pendule amorti en régime sinusoïdal forcé, dont l'équation du mouvement est donnée par :

$$\frac{d^2\theta}{dt^2} + \frac{\omega_0}{Q} \frac{d\theta}{dt} + \omega_0^2 \sin(\theta) = \omega_0^2 \cos(\omega t)$$

Données :  $\omega_0 = 50 \text{ rad} \cdot \text{s}^{-1}$ ,  $Q = 0,02$ ,  $\omega = 10 \text{ rad} \cdot \text{s}^{-1}$ ,  $\theta(0) = -\frac{\pi}{4} \text{ rad}$  et  $\dot{\theta}(0) = 0 \text{ rad} \cdot \text{s}^{-1}$ .

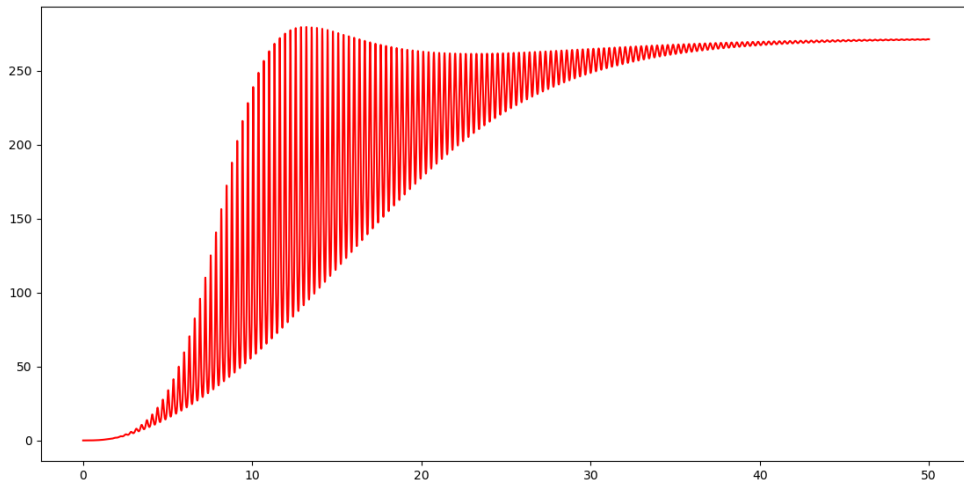
- 1) Résoudre l'ED à l'aide de la fonction « odeint ».

### Exercice n°1 - ED d'ordre 1, avec solution analytique

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 # Paramètres
5 C0 = 0.1
6 k = 0.1
7
8 # Solution analytique
9 t_max = 10/(k*C0) # t_max = 5 x Temps de demi-réaction
10 t1 = np.linspace(0, t_max, 10000)
11 C1 = (1/C0 + k*t1)**(-1)
12
13 # Solution numérique : Euler
14 dt = t_max/1e4
15 t2 = np.arange(0, t_max, dt)
16 N = len(t2)
17 C2 = np.zeros(N)
18 C2[0] = C0
19 for i in range(N-1):
20     C2[i+1] = C2[i] - k*dt*C2[i]**2
21
22 # Solution numérique : odeint
23 from scipy.integrate import odeint
24
25 def deriv(y,t):
26     dydt = -k*y**2
27     return dydt
28
29 t3 = np.linspace(0, t_max, 10000)
30 C3 = odeint(deriv, C0, t3)
31
32 # Graphiques
33 fig, ax = plt.subplots()
34 ax.plot(t1, C1, 'g-')
35 ax.plot(t2, C2, 'r--')
36 ax.plot(t3, C3, 'k--')
```

## Exercice n°2 - ED d'ordre 1, sans solution analytique

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from scipy.integrate import odeint
4
5  $\tau = 5$ 
6  $\omega = 20$ 
7  $f_0 = 0$ 
8
9 def deriv(y,t):
10     dydt = (t**2 - np.sin( $\omega$ *t)*y**2) * np.exp(-t/ $\tau$ )
11     return dydt
12
13 t = np.linspace(0, 50, 10000)
14 f = odeint(deriv, f0, t)
15
16 fig, ax = plt.subplots()
17 ax.plot(t, f, 'r-')
```



## Exercice n°3 - ED d'ordre 2

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from scipy.integrate import odeint
4
5  $\omega_0 = 50$ 
6  $\omega = 10$ 
7  $Q = 0.02$ 
8  $Cl = [-np.pi/4, 0]$ 
9
10 def deriv(y,t):
11     dydt = [y[1],  $\omega_0$ **2*np.cos( $\omega$ *t) -  $\omega_0$ **2*np.sin(y[0]) -  $\omega_0$ / $Q$ *y[1]]
12     return dydt
13
14 t_max = 2*np.pi/ $\omega$  * 20 # t_max = 20 périodes
15 t = np.linspace(0, t_max, 10000)
16 sol = odeint(deriv, Cl, t)
17
18  $\theta$ , d $\theta$ dt = sol[:,0], sol[:,1]
19
20 fig, ax = plt.subplots()
21 ax.plot(t,  $\theta$ , 'r-')
```

