9

Simulation d'un système en physique, chimie, SII

Exemple. On considère la réaction :

$$Fe^{2+} + Co^{3+} = Fe^{3+} + Co^{2+}$$

t (s)	C ($\mathrm{mol \cdot L^{-1}}$)
20	$2.78 \cdot 10^{-4}$
40	$1.92 \cdot 10^{-4}$
60	$1.47 \cdot 10^{-4}$
80	$1.19 \cdot 10^{-4}$
100	$1.0 \cdot 10^{-4}$
120	$0.86 \cdot 10^{-4}$

C'est une réaction d'ordre 2 et l'évolution de la concentration en ions Fe^{2+}, notée C, est décrite par l'équation différentielle :

$$\frac{\mathrm{d}C}{\mathrm{d}t} = -kC^2$$

On donne $k = 80.2\ \mathrm{mol^{-1} L \cdot s^{-1}}$ et $C_0 = 5.0 \cdot 10^{-4}\ \mathrm{mol \cdot L^{-1}}$. Ci-dessus : mesures expérimentales de C en fonction du temps.

◇ But de la simulation :

- Pour des instants donnés $t_0, t_1, \ldots, t_{N-1}$, déterminer approximativement les valeurs de la concentration C, notées $C_0, C_1, \ldots, C_{N-1}$;
- On a $C_0 = 5 \cdot 10^{-4}$ et $C_1, \ldots, C_{N-1}$ sont à calculer ;
- Pour pouvoir mener le calcul, il faut que t_{i+1} soit « proche » de t_i, on pose donc $t_i = i\Delta_t$ avec Δ_t fixé « assez petit » (c'est le *pas* de la simulation, par ex. $\Delta_t = 0.1$ s).

	t	C
0	0	$5 \cdot 10^{-4}$
1	Δ_t	?
2	$2\Delta_t$	?
$\vdots$	$\vdots$	$\vdots$
$N-2$	$(N-2)\Delta_t$	?
$N-1$	$(N-1)\Delta_t$	?

◇ Avec PYTHON, C et t sont des listes : $t = [t_0, t_1, \ldots, t_{N-1}]$ et $C = [C_0, C_1, \ldots, C_{N-1}]$ avec $t[i] = i\Delta_t$ et $C[0] = 5 \cdot 10^{-4}$.

⚠ Bien comprendre les différentes notations :

- $C(t)$ est la concentration en ions Fe^{2+} à l'instant t ;
- $t_0, t_1, \ldots, t_{N-1}$ sont les instants où l'on va calculer numériquement la concentration ;
- $t[i]$ représente, en PYTHON, ce que l'on note mathématiquement t_i : $t[i] = t_i = i\Delta_t$;
- $C[i]$ représente, en PYTHON, C_i (ou encore $C(t_i)$), donc la concentration à l'instant t_i.

◇ Principe de l'approximation numérique : comme $t_{i+1} - t_i$ est « petit »

$$\frac{C(t_{i+1}) - C(t_i)}{t_{i+1} - t_i} \simeq C'(t_i) = -kC(t_i)^2$$

◇ Traduction avec PYTHON, on *définit* $C[i+1]$ *de sorte que* :

$$\frac{C[i+1]-C[i]}{t[i+1]-t[i]} = -kC[i]^2$$

c'est à dire que l'on *calcule* $C[i+1]$ à partir de $C[i]$ en *posant* :

$$\forall i \in [\![0, N-2]\!],\ C[i+1] = C[i] - (t[i+1]-t[i])kC[i]^2$$

Pour remplir la liste C, on répète cette instruction pour i compris entre 0 et $N-2$.

```
# Paramètres de la simulation numérique :
Delta_t=0.1 # Delta_t est "petit"
N=1200      # N*Delta_t=120 secondes
# La liste t :
t=[i*Delta_t for i in range(N)]
C=[0]*N # Liste de taille N, pour l'instant C=[0,0,...,0]
# Données numériques de l'énoncé :
C[0]=5e-4
k=80.2
for i in range(N-1): # On a bien : i varie de 0 à N-2
    C[i+1]=C[i]-(t[i+1]-t[i])*k*C[i]**2
```

◇ On représente graphiquement C en fonction de t pour les valeurs calculées. Sur le même graphique, on représente également les valeurs mesurées (enregistrées dans deux listes notées **t_mes** et **C_mes**) :

```
import matplotlib.pyplot as plt
plt.plot(t,C)
t_mes=[20,40,60,80,100,120]
C_mes=[2.78e-4,1.92e-4,1.47e-4,1.19e-4,1.0e-4,0.86e-4]
plt.plot(t_mes,C_mes,'+',markersize=12)
```