Annexe 1 : Utilisation de la fonction odeint pour résoudre une équation différentielle ou un système d'équations différentielles

▪La fonction odeint doit être chargée à partir du module scipy.integrate :

Chargement du module : from scipy.integrate **import** odeint

▪Cette méthode de résolution numérique ne concerne que des équations différentielles du 1^er^ ordre que l'on peut mettre sous la forme : **f'(t) = F (f (t) , t)** .
f est la fonction de la variable t que l'on cherche à déterminer , f'(t) désigne la fonction dérivée(par rapport à la variable t) de f .
La valeur de f à l'instant t_0 doit être connue (***condition initiale***)

▪Principe de la résolution :

On se donne un vecteur t = (t_0 , t_1, ... , t_n) qui donne l'échantillon de temps sur lequel on veut résoudre le problème . Partant de $f_0 = f(t_0)$ qui est imposé, on calcule de proche en proche des approximations f_1 , f_2 ,…f_n de $f(t_1)$, $f(t_2)$, ... , $f(t_n)$ en utilisant les valeurs de la dérivée de f , c'est-à-dire la fonction F .

▪Utilisation de odeint

① Créer un tableau numpy des valeurs de la variable dont dépend la fonction f cherchée , solution d e l'équation différentielle . Ainsi , par exemple , pour l'équation différentielle décrite ci-dessus , la variable est le temps t ; on crée le tableau numpy T =np.linspace (min , max , nombre de valeurs)
② Définir la fonction F
③ Introduire la fonction odeint sachant qu'elle renvoie un tableau numpy , les paramètres d'entrée de la fonction odeint sont la fonction F , la condition initiale et le tableau numpy des valeurs de la variable selon la syntaxe

Odeint(F , [f_0] , T)

Conditions initiales

Le tableau numpy renvoyé contient les valeurs de la fonction f pour les différentes valeurs de la variable contenues dans l'intervalle choisi.

La fonction odeint peut être utilisée pour résoudre un système d'équations en vectorisant .

▪Exemples

Exemple 1 : résolution de l'équation différentielle $\frac{dC}{dt} = -kC$ *avec C (t=0) =0.01 molL*$^{-1}$

```
T=np.linspace ( 0, 500 , 1000)
def F(C,t):
  return (-k*C)
sol = odeint ( F, [0.01],T)
```

Exemple 2 : résolution du système d'équations $\begin{cases} \frac{dC1}{dt} = -k * C1 \\ \frac{d\,C2}{dt} = k * C1 - k2 * C2 + 4 * t \end{cases}$

avec les conditions initiales C1 (t=0) =0.01 molL^{-1} *et C2(t=0)=0*

```
T=np.linspace ( 0, 500 , 1000)
def F(C,t):
   C1, C2 = C[0] , C[1]
   Return (-k*C1 , k*C1-k2*C2+ 4*t)
sol = odeint ( F, [0.01,0],T)
```

C1=sol[:,0] tableau contenant les valeurs de C1 pour les différentes valeurs de la variable t
C2=sol[:,1] tableau contenant les valeurs de C2 pour les différentes valeurs de la variable t