ON 5 - Résolution d'une équation différentielle avec Odeint

I. Résolution d'une équation différentielle du 1er ordre

On cherche une solution approchée du problème de Cauchy sur l'intervalle I = [a,b] :

$$\begin{cases} y'(t) = f(y(t),t) \\ y(a) = y_0 \end{cases}$$

La fonction **`odeint`** qui se trouve dans le module `scipy.integrate` sert à résoudre toutes les équations différentielles du 1er ordre, et cela plus précisément que la méthode d'Euler.

On lui donne la **fonction $\boldsymbol{f(y(t),t)}$** qui est telle que $y'(t) = f(y(t),t)$, **une condition initiale**, et **un tableau des valeurs $\boldsymbol{t_k}$** pour lesquelles on souhaite calculer une valeur approchée de $y(t_k)$.

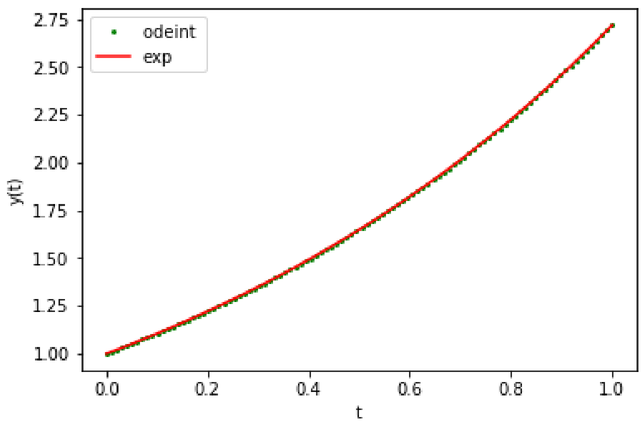
La fonction retourne un tableau de valeurs qu'on peut représenter graphiquement.

Exemple : y' = y avec y(0) = 1 dont la solution est connue : y(t) = e^t, sur l'intervalle [0,1].

```
from scipy.integrate import odeint
import matplotlib.pyplot as plt
import numpy as np

# fonction de l'équation différentielle y' = y
def f(y,t):
       return y
t = np.linspace (0 ,1 ,100)
y = odeint (f,1,t)
plt.plot (t,y,'go',ms=2,label='odeint ')

# solution exacte
plt.plot (t,np.exp (t),'r',label='exp')
plt.xlabel('t')
plt.ylabel('y(t)')
plt.legend()
```



II. Résolution d'une équation différentielle du 2nd ordre

On cherche une solution approchée du problème de Cauchy sur l'intervalle I = [a,b] :

$$\begin{cases} y''(t) = f(y(t), y'(t), t) \\ y(a) = y_0 \\ y'(a) = y'_0 \end{cases}$$

Odeint ne permet d'intégrer que des équations différentielles du premier ordre.
Pour intégrer une équation différentielle du deuxième ordre, il faut la convertir en un système de deux équations différentielles du premier ordre.

On introduit la variable vectorielle $Y = (y(t), y'(t))$.

Si on dérive Y, on obtient la variable vectorielle $Y' = (y'(t), f(y(t), y'(t), t))$

Y est donc solution du système différentiel :

$$\begin{cases} Y'(t) = F(Y, t) \\ Y(a) = (y_0, y'_0) \end{cases}$$

Où $F(Y,t)$ est une fonction qui renvoie le vecteur $(y'(t), f(y(t), y'(t), t))$

Exemple d'implémentation :

```
# Exemple : OHNA tel que y'' = −y sur l'intervalle [0,30] avec
y(0)=0,y'(0)=1, n = 100

import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint

# fonction du système différentiel
def F(Y,t):
    return np.array([Y[1],-Y[0]])

# définition du tableau temps
t = np.linspace(0, 30, 100)

# intégration du système
Y = odeint(F, [0,1],t)

# représentation graphique de la solution
plt.plot(t, Y[:,0])   #on ne veut que la 1ère colonne pour tracer y(t)
plt.xlabel('t')
plt.ylabel('y')
```