ON 4 - Résolution d'une équation différentielle par la méthode d'Euler

I. Equation différentielle du 1er ordre

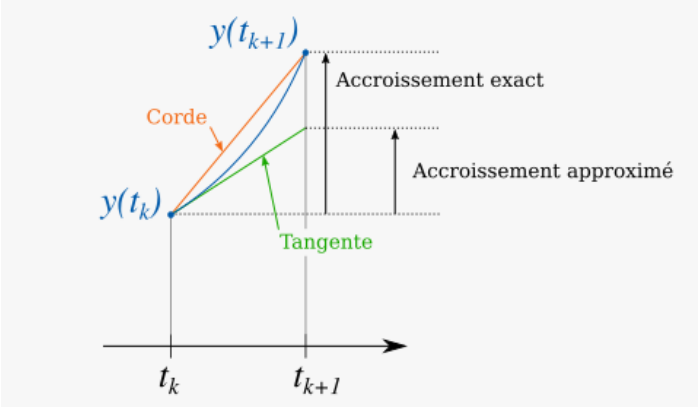
1. Résolution par la méthode d'Euler

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

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

On casse l'intervalle $I = [a, b]$ en n intervalles égaux, de largeur $h = \frac{(b-a)}{n}$ appelée pas.

La méthode d'Euler consiste à calculer successivement des valeurs approchées des réels $y(t_k)$, avec $t_k = a + kh$ où k = 1, ...,n, en approchant la courbe sur l'intervalle $[t_k, t_{k+1}]$ par sa tangente en t_k. La pente de la tangent en t_k est $y'(t_k) = f(y(t_k), t_k)$.



Récurrence de la méthode d'Euler : $y(t_{k+1}) \approx y(t_k) + h\, y'(t_k)$, où $y'(t_k) = f(y(t_k), t_k)$

L'approximation sera d'autant meilleure que le pas h est petit ou que le nombre d'intervalles n est grand.

2. Exemple d'implémentation :

On veut faire une résolution approchée de $y'(t) = f(y(t), t)$ sur l'intervalle $[a, b]$ cassé en n intervalles.

Principe :

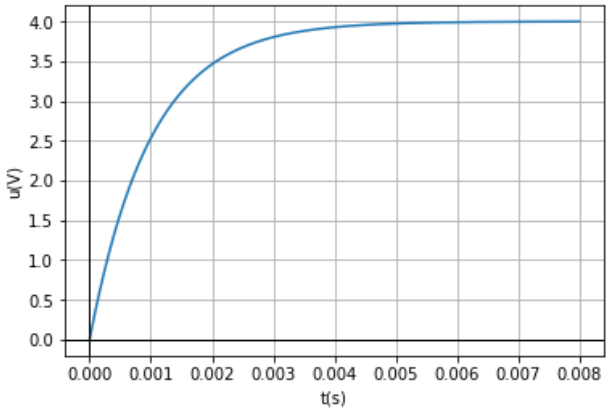
```
t=[a]                          Initialisation de la liste t
y= [y0]                        Initialisation de la liste y : y(a) =  condition initiale
n= valeur de n                 Nécessaire pour calculer h est le pas de la méthode
pour k de 0 à n exclu faire :
        Rajouter (t[k] +  h) à la liste t
        Rajouter (y[k] + h * f(y[k],t[k])) à la liste y      On tape directement la fonction f(y,t)
```

Programme Python :

```
# Exemple : charge d'un condensateur initialement déchargé
y'+ 1/τ y = 1/τ 4, étude pendant 8 ms avec y(0)= 0, τ = 1 ms et n = 1000

t = [0]
y = [0]
n = 1000
h = (0.008-0)/n
tau = 0.001
for k in range(n):
    t.append(t[k]+h)
    y.append(y[k]+h*((4-y[k])/tau))

#Pour tracer la solution :
plt.plot(t,y)
plt.xlabel('t(s)')
plt.ylabel('u(V)')
plt.axhline(lw=1,color='k')
plt.axvline(lw=1,color='k')
plt.grid()
```



II. Résolution d'une équation différentielle du 2nd ordre par la méthode d'Euler

Les algorithmes d'intégration numérique dont on dispose ne permettent d'intégrer que des équations différentielles du premier ordre.

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

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

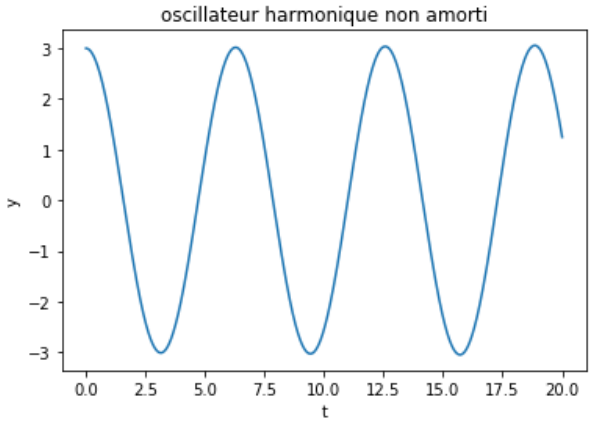
Pour intégrer une équation différentielle du deuxième ordre, il faut la convertir en un système de deux équations différentielles du premier ordre.

On introduit la variable Y telle que $y'(t) = Y(t)$. Comme $y''(t) = Y'(t)$, l'équation différentielle devient : $Y'(t) = f(y, Y, t)$, où la fonction $f(y, Y, t)$ dépend de l'équation différentielle.

On peut donc résoudre par la méthode d'Euler n'importe quelle équation différentielle du 2nd ordre à condition de travailler avec 2 variables : $y(t)$ et $Y(t) = y'(t)$.

$$y(t_{k+1}) \approx y(t_k) + h\, y'(t_k)$$

$$y'(t_{k+1}) \approx y'(t_k) + h\, y''(t_k), \text{ où } y''(t_k) = f(y(t_k), y'(t_k), t_k)$$

Exemple d'implémentation :

```
# Exemple : OHNA tel que y''= - y sur l'intervalle [0,20] avec
y(0)=3,y'(0)=0, n = 1000

t = [0]
y = [3]
Y=[0]
n = 1000
h = (20-0)/n
for k in range(n):
    t.append(t[k]+h)
    Y.append(Y[k]+h*(-y[k]))
    y.append(y[k]+h*Y[k])

#Pour tracer la solution :
plt.plot(t,y)
plt.ylabel('y')
plt.xlabel('t')
plt.title('oscillateur harmonique non amorti')
```