Équations différentielles et méthode d'Euler

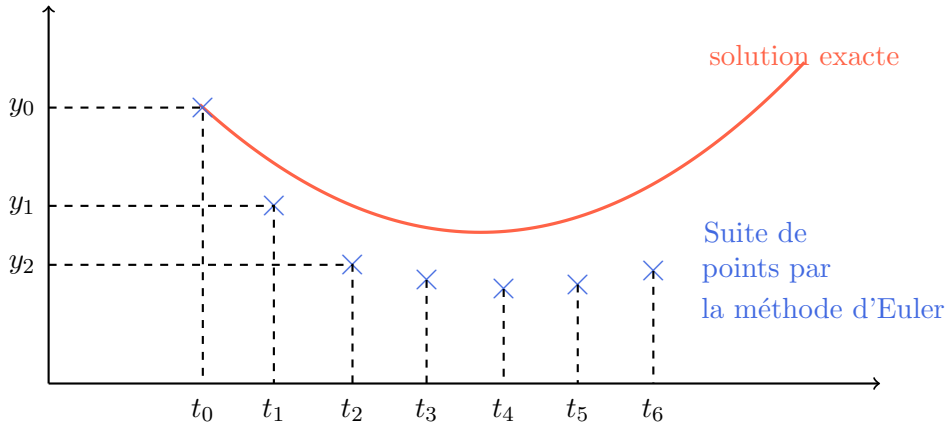
Méthode d'Euler

On souhaite résoudre des équations différentielles du type $y'(t) = y(t)$ avec $y(t_0) = y_0$ en utilisant la méthode d'Euler sur un intervalle de la forme $[t_0\,;\,t_0+T]$.

On partage l'intervalle $[t_0\,;\,t_0+T]$ en n intervalle de largeur $h = \dfrac{T}{n}$.

On définit ensuite une suite de points de la façon suivante :

- Le premier point $\mathrm{A}_0(t_0\,;\,y_0)$.
- $\mathrm{A}_1(t_1\,;\,y_1)$ avec $t_1 = t_0 + h$ et $y_1 = y_0 + hy'(t_0)$.
  L'équation différentielle nous permet ainsi d'écrire : $y_1 = y_0 + hf(t_0)$.
  Sur l'intervalle $[t_0\,;\,t_1]$, on approche la courbe solution (qui est inconnue) par le segment $[\mathrm{A}_0\mathrm{A}_1]$ de coefficient directeur $y'(t_0)$, c'est à dire par la tangente à la courbe solution au point A_0.
- $\mathrm{A}_2(t_2\,;\,y_2)$ avec $t_2 = t_1 + h$ et $y_2 = hy'(t_1)$ ou encore $y_2 = y_1 + hf(t_1)$.
  Sur l'intervalle $[t_1\,;\,t_2]$, on approche la courbe solution par le segment $[\mathrm{A}_1\mathrm{A}_2]$ qui est parallèle à la tangente à la courbe solution au point d'abscisse t_1.
- On continue jusqu'au dernier point : $\mathrm{A}_n(t_n\,;\,y_n)$ avec $t_n = t_{n-1} + h$ et $y_n = y_{n-1} + hf(t_{n-1})$.



On retiendra la formule d'approximation suivante :

$$f(x+h) \approx f(x) + hf'(x)$$

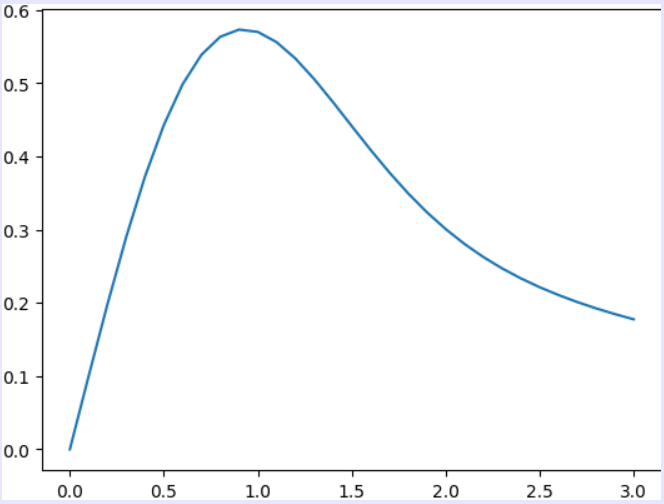
Voir un exemple au dos.

Exemple On considère l'équation différentielle $\begin{cases} y' + 2xy &= 1 \\ y(0) &= 0 \end{cases}$.

La première ligne se ramène à $y' = 1 - 2xy$.

On obtient ainsi $y(x+h) = y(x) + hy'(x) = y(x) + h(1 - 2xy(x)) = y(x)(1 - 2hx) + h$.

Pour le premier point, cela donne $y(0+h) = y(0)(1-0) + h = h$. $A_0(h\,;\,h)$.
Pour le deuxième point : $y(h+h) = y(h)(1-2h^2) + h = 2h - 2h^3$. $A_1(2h\,;\,2h-2h^3)$.
Et ainsi de suite.

Il ne reste plus qu'à programmer cela sous Python. Il faudra préciser la valeur du pas h souhaité et les bornes de l'intervalle $[a\,;\,b]$ sur lequel faire l'approximation. Nous choisirons $a = 0$ et $b = 3$ pour un pas de 0,1.

```
import matplotlib.pyplot as plt
def courbe(a,h):
    x=0
    y=0
    X=[x]
    Y=[y]
    borne=int(a/h)
    for i in range(borne):
        y=y*(1-2*h*x)+h
        x=x+h
        X.append(x)
        Y.append(y)
    plt.plot(X,Y)
    plt.show()
    return X,Y
courbe(3,0.1)
```