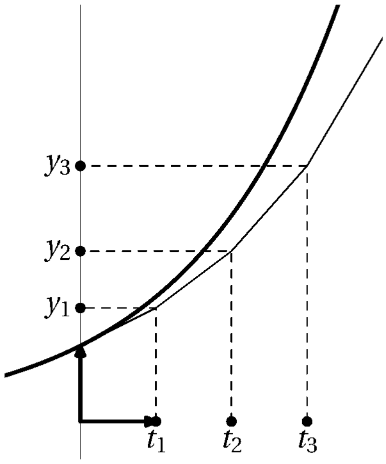
20

Méthode d'Euler



◇ On considère une équation différentielle d'ordre 1 avec condition initiale (*problème de Cauchy*) :

$$\begin{cases} y' = F(y,t) \\ y(t_0) = y_0 \end{cases}$$

◇ Le but de la résolution approchée :

- Pour des instants donnés $t_0, t_1, \dots, t_{N-1}$, déterminer approximativement les valeurs prises par la fonction y, notées $y_0, y_1, \dots, y_{N-1}$;
- La valeur y_0 est donnée et les autres sont à calculer ;
- Pour pouvoir mener le calcul, il faut que t_{i+1} soit « proche » de t_i, on aura en général $t_i = t_0 + i\Delta_t$ avec Δ_t fixé « assez petit » (c'est le *pas* de la simulation).

⚠ Bien comprendre les différentes notations :

- $y(t)$ est la valeur en t de la fonction y solution du problème de Cauchy de départ ;
- $t_0, t_1, \dots, t_{N-1}$ sont les instants où l'on va calculer numériquement (et approximativement) les valeur de y ;
- $t[i]$ représente, en PYTHON, ce que l'on note mathématiquement t_i ;
- $y[i]$ représente, en PYTHON, y_i (ou encore $y(t_i)$).

	t	y
0	t_0	y_0
1	t_1	?
2	t_2	?
$\vdots$	$\vdots$	$\vdots$
$N-2$	t_{N-2}	?
$N-1$	t_{N-1}	?

◇ Avec PYTHON, t et y sont des listes $t = [t_0, \dots, t_{N-1}]$ et $y = [y_0, \dots, y_{N-1}]$ (il peut également s'agir de vecteurs NUMPY).

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

$$\frac{y(t_{i+1}) - y(t_i)}{t_{i+1} - t_i} \simeq y'(t_i) = F(y(t_i), t_i)$$

En utilisant les notations y_i et y_{i+1} :

$$\frac{y_{i+1} - y_i}{t_{i+1} - t_i} \simeq F(y_i, t_i)$$

On *définira* donc $y[i+1]$ en posant :

$$\forall i \in [\![0, N-2]\!],\ y[i+1] = y[i] + (t[i+1] - t[i]) \times F(y[i], t[i])$$

◇ L'algorithme correspondant :
- Données : F, y_0, t ;
- N est le nombre d'éléments de t ;
- On définit y de même taille que t ;
- Résultat : y.

$y[0] \leftarrow y_0$
pour i **allant de** 0 **à** $N-2$
$\quad y[i+1] \leftarrow y[i] + (t[i+1] - t[i]) \times F(y[i], t[i])$
fin pour

◇ Traduction en PYTHON :

```
def euler(F,y0,t):
    N=len(t)
    y=[0]*N
    y[0]=y0
    for i in range(N-1):
        y[i+1]=y[i]+(t[i+1]-t[i])*F(y[i],t[i])
    return y
```

Exemple Python. On considère l'équation $y' = y$ avec la condition initiale $y(0) = 1$. La solution est la fonction exponentielle, ce qui permet de comparer les résultats obtenus. On va travailler sur l'intervalle $[0,1]$ en considérant tout d'abord un pas de 0.25 puis un pas de 0.1. La fonction exponentielle est représentée en pointillés pour comparer.

```
import numpy as np
import matplotlib.pyplot as plt
def F(y,t):
    return y
```

```
t=np.linspace(0,1,5)
y=euler(F,1,t)
plt.plot(t,y,'b-')
plt.plot(t,np.exp(t),'r--')
```

```
t=np.linspace(0,1,11)
y=euler(F,1,t)
plt.plot(t,y,'b-')
plt.plot(t,np.exp(t),'r--')
```