

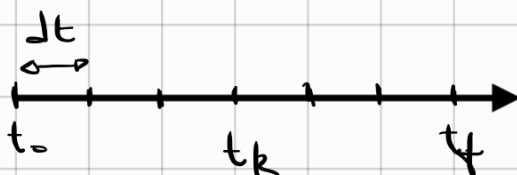
Résolution d'équations différentielles

• Premier exemple :

$$\begin{cases} y'(t) = -2 \cdot t \cdot y(t) + \cos(t) \\ \quad \quad \quad = F(y, t) \\ y(0) = 1 \end{cases}$$

à résoudre sur $I = [0, 10]$

Notons $t_k = k \cdot dt + \underbrace{t_0}_0$



Et on fait l'hypothèse que:

$$y'(t) \approx \frac{y(t+dt) - y(t)}{dt}$$

Et donc $y'(t_k) \approx \frac{y(t_{k+1}) - y(t_k)}{dt}$

On approxime ensuite $y(t_k) \approx y_k$ où les y_k seront calculés de manière itérative.

On a donc: $\frac{y_{k+1} - y_k}{dt} = -2 \cdot t_k \cdot y_k + \cos(t_k)$

Et donc $y_{k+1} = y_k + dt \cdot (-2 \cdot t_k \cdot y_k + \cos(t_k))$, ce

qui pourrait aussi s'écrire:

$$\begin{cases} y_{k+1} = y_k + dt \cdot F(y_k, t_k) \\ \text{et } y_0 = 1 \end{cases}$$

→ Sur Python, on peut résoudre "à la main":

```
import numpy as np
```

```
Y = [y0]
```

```
T = [0]
```

```
dt = 0.1
```

```
n = int((10 - 0) / dt)
```

```
for i in range(0, n + 1):
```

```
    ysuiv = Y[-1] + dt * (-2 * T[-1] * Y[-1]  
                        + np.cos(T[-1]))
```

```
    Y.append(ysuiv)
```

```
    T.append(T[-1] + dt)
```

→ Ou utiliser la fonction odeint où

```
Y = odeint(F, Y0, T)
```

Fonction $F(Y, t)$

condition initiale

liste des temps

```
def F(Y, t):
```

```
    return -2 * t * Y + np.cos(t)
```

```
T = np.linspace(0, 10, n + 1)
```

t_{initial}

t_{final}

nombre de points

```
Y = odeint(F, 1, T)
```

condition initiale

```
import matplotlib.pyplot as plt
```

```
plt.plot(T, Y)
```

• Exercice n° 1 :

$$\begin{cases} u'(t) = \frac{1}{u(t)} - 5 \cdot u(t) \cdot \sin(u(t)) \\ u(0) = 2 \end{cases}$$

```
def F(x, t):
    return 1/x - 5 * x * np.sin(x)
```

n = 100

t0 = 0.

tf = 5.

LT = np.linspace(t0, tf, n)

x0 = 2

LX = odeint(F, x0, LT)

• Exercice n° 2 :

$$\begin{cases} u'(t) = -\cos(u(t)) - \cos(t) \\ v'(t) = -\sin(u(t)) - \sin(t) \\ u(0) = 2 \\ v(0) = 4 \end{cases}$$

On pose ici : $Y = \begin{pmatrix} u \\ v \end{pmatrix}$ ici $Y[0]$ donne u
 $Y[1]$ " v

et donc $Y'(t) = \begin{pmatrix} u'(t) \\ v'(t) \end{pmatrix}$

$$\begin{matrix} \text{A} \swarrow \\ \text{B} \nwarrow \end{matrix} = \begin{pmatrix} -\cos(u(t)) - \cos(t) \\ -\sin(u(t)) - \sin(t) \end{pmatrix}$$

$F(Y(t), t)$

On a donc :

```
def F(y, t):
```

$$A = -np.\cos(y[1]) - np.\cos(t)$$

$$B = -np.\sin(y[0]) - np.\sin(t)$$

```
return [A, B]
```

$$t0 = 0$$

$$tf = 1$$

$$n = 100$$

$$LT = np.linspace(t0, tf, n)$$

$$y0 = [2, 4]$$

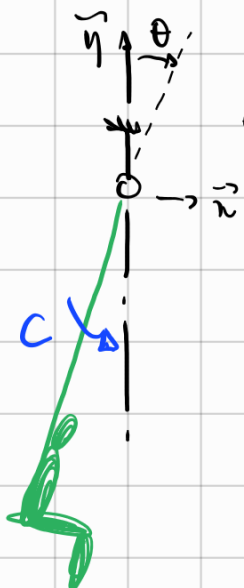
$$LY = \text{odeint}(F, y0, LT)$$

$$\text{plt.plot}(LY[:, 0], LY[:, 1])$$

$\downarrow$ solution

$\downarrow$ solution

• **Exercice n° 3:** on pose : $y = \begin{pmatrix} \theta \\ \dot{\theta} \end{pmatrix}$



et donc $\dot{y} = \begin{pmatrix} \dot{\theta} \\ \ddot{\theta} \end{pmatrix} = \begin{pmatrix} \dot{\theta} \\ -\frac{m \cdot g \cdot L}{J} \cdot \sin \theta + C \end{pmatrix}$

avec f° de θ et $\dot{\theta}$

```
def F(y, t):
```

$$A = y[1]$$

```
if y[0] <= 0 and y[1] >= 0:
```

$$B = (-m * g * L * np.\sin(y[0]) + C) / J$$

```
else:
```

$$B = -m * g * L * np.\sin(y[0]) / J$$

```
return [A, B]
```

$y_0 = [0.1, 0]$

$t_0 = 0$

$t_f = 60$ # s

$n = 100$

$LT = \text{np.linspace}(t_0, t_f, n)$

$LY = \text{odeint}(F, y_0, LT)$

$\text{plt.plot}(LT, LY[:, 0])$ # pour tracer θ en f° du tps