1 Présentation de la méthode d'Euler

Définition 1

On appelle **équation différentielle ordinaire**[1] toute équation différentielle de la forme :

$$y'(t) = F(t, y(t)).$$

Le but de la **méthode d'Euler** est de résoudre de manière approchée, sur un intervalle $[a, b]$, une équation différentielle ordinaire (avec condition initiale) de la forme :

$$\begin{cases} y(a) = y_0 \\ \forall t \in [a, b],\ y'(t) = F(t, y(t)). \end{cases}$$

Pour cela, on veut construit une suite finie de $n + 1$ points $(y_k)_{0 \leqslant k \leqslant n}$ approchant les valeurs exactes $(y(t_k))_{0 \leqslant k \leqslant n}$ où :

$$\forall k \in [\![0, n]\!],\ t_k = a + kh \quad \text{en notant} \quad h = \frac{b-a}{n}.$$

La construction des valeurs approchées se base l'approximation $y'(t) \approx \dfrac{y(t+h) - y(t)}{h}$ lorsque le **pas** h est suffisamment petit. Ainsi, on obtient :

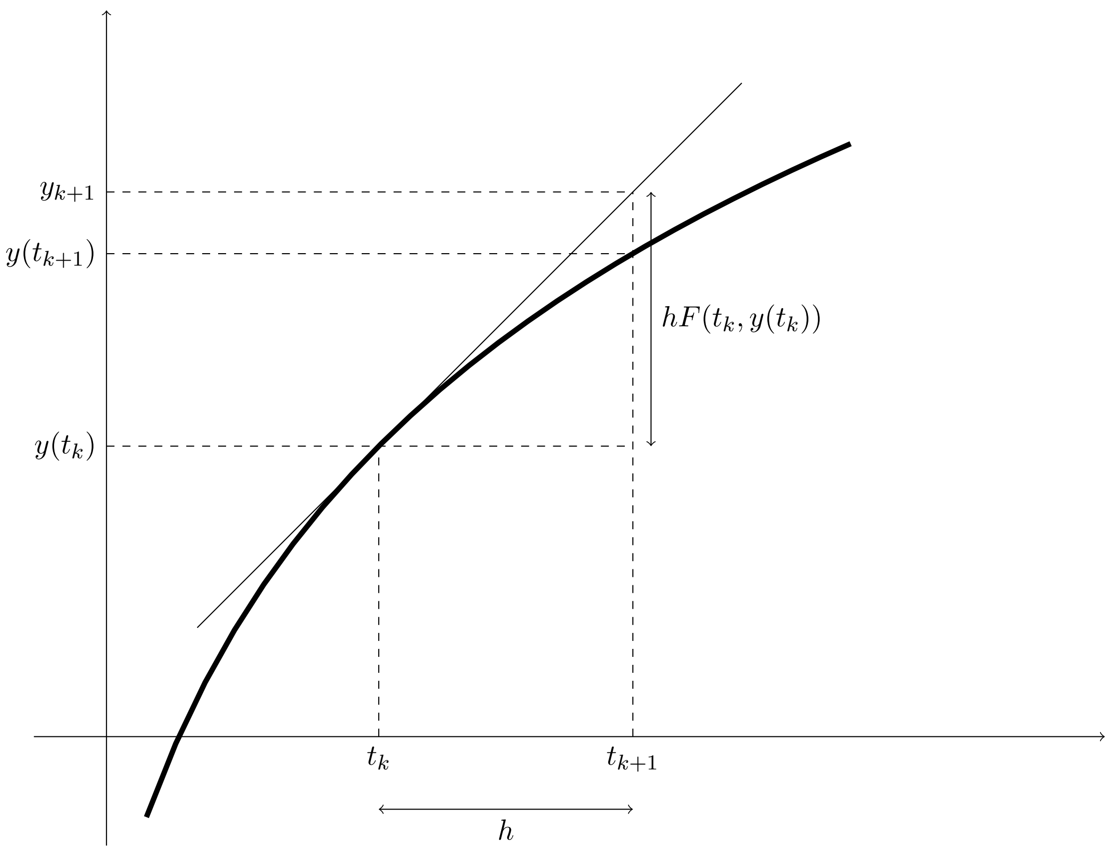
$$y(t + h) \approx y(t) + hy'(t) = y(t) + hF(t, y(t)).$$

On peut donc prédire la valeur de la solution à l'instant $t + h$ à partir de sa valeur à l'instant t.

La suite des valeurs approchées calculées par la méthode d'Euler est définie par la relation de récurrence ci-dessous, appelée, **schéma numérique de la méthode d'Euler** :

$$\boxed{\begin{cases} y_0 = y(t_0) \quad \textit{condition initiale (seule valeur exacte connue)} \\ \forall k \in [\![0, n-1]\!],\ y_{k+1} = y_k + hF(t_k, y_k). \end{cases}}$$

Graphiquement, cette méthode consiste à approcher la courbe solution passant par le point de coordonnées $(t_k, y(t_k))$ par sa tangente à chaque instant, comme l'illustre la figure suivante :



2 Premiers exemples

Question 1. Recopier et compléter le script ci-dessous afin de résoudre de manière approchée par la méthode d'Euler l'équation différentielle $y'(t) = \sin(t)\sin(y(t))$ sur l'intervalle $[0, 50]$ avec la condition initiale $y(0) = 1$. On affichera le graphe solution.

```
import matplotlib.pyplot as ...
import numpy as np
a, b = ..., ...
y_0, n = 1, 255
t, y  = [a], [y_0]
h = ...
for k in range(...):
    y.append(y[k] + ...)
    t.append(...)
plt.plot(..., ...)
plt.show()
```

Question 2. Écrire une fonction `euler` prenant en argument une fonction `F`, trois flottants `a`, `b` et `y_0`, un entier naturel `n`, et qui résout de manière approchée l'équation différentielle ordinaire $y'(t) = F(t, y(t))$ sur l'intervalle $[a, b]$ par la méthode d'Euler, avec pour condition initiale $y(a) =$ `y_0`. La fonction devra renvoyer un couple de deux listes `t` et `y` de longueur `n+1`.

Question 3. On cherche à représenter graphiquement la solution exacte et les solutions approchées par la méthode d'Euler de l'équation différentielle $y' = y$ avec pour condition initiale $y(0) = 1$ pour différentes valeurs du nombre $n + 1$ de points. Recopier et compléter le code ci-dessous et commenter le résultat.

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

a, b, y_0 = ..., ..., ...
for ... in [10,100,1000]:
    t, y = ...
    legende = "n = " + str(...)
    plt.plot(..., ..., label = legende)
plt.plot(..., np.exp(t), label= "exp")
plt.legend(loc = "best")
plt.show()
```

La méthode d'Euler n'est pas la seule méthode numérique de résolution d'équation différentielle, c'est juste la plus simple. On présente ci-dessous deux autres méthodes par leur schéma numérique.

- La **méthode de Heun** est définie par :

$$\boxed{\forall k \in [\![0, n-1]\!],\ y_{k+1} = y_k + h\left(\frac{p_1 + p_2}{2}\right) \text{ où } \begin{cases} p_1 = F(t_k, y_k) \\ p_2 = F(t_k + h, y_k + hp_1). \end{cases}}$$

- La **méthode RK$_4$ de Runge-Kutta** est définie par :

$$\boxed{\forall k \in [\![0, n-1]\!],\ y_{k+1} = y_k + h\left(\frac{p_1 + 2p_2 + 2p_3 + p_4}{6}\right) \text{ où } \begin{cases} p_1 = F(t_k, y_k) \\ p_2 = F\left(t_k + \dfrac{h}{2}, y_k + \dfrac{h}{2}p_1\right) \\ p_3 = F\left(t_k + \dfrac{h}{2}, y_k + \dfrac{h}{2}p_2\right) \\ p_4 = F(t_k + h, y_k + hp_3). \end{cases}}$$

Question 4. En s'inspirant du code écrit à la question 2, écrire deux fonctions `heun` et `rk4` implémentant les méthodes de Heun et RK_4.

Question 5. Afficher sur le même graphique la solution exacte et les solutions approchées à l'équation différentielle de la question 3 par les méthodes d'Euler, de Heun et RK_4 pour $n = 10$ (on résoudra toujours sur l'intervalle $[0, 1]$).

3 Modèle proie-prédateur de Lotka-Volterra

Le modèle de proie-prédateur permet de décrire la dynamique de populations d'un système biologique où interagissent un prédateur et sa proie. Il a été proposé de manière indépendante par Alfred James Lotka (1925) et Vito Volterra (1926).
En notant $x(t)$ et $y(t)$ les effectifs respectifs des proies et des prédateurs à l'instant t, le modèle s'écrit sous la forme du système d'équations différentielles suivantes :

$$\begin{cases} \dfrac{\mathrm{d}x(t)}{\mathrm{d}t} = x(t)\left(\alpha - \beta y(t)\right) \\ \dfrac{\mathrm{d}y(t)}{\mathrm{d}t} = y(t)\left(\delta x(t) - \gamma\right) \end{cases}$$

où :

- α est le taux de reproduction intrinsèque des proies ;
- β est le taux de mortalité des proies due aux prédateurs ;
- δ est le taux de reproduction des prédateurs en fonction des proies mangées ;
- γ est le taux de mortalité intrinsèque des prédateurs.

Question 6. Comment interpréter la première équation lorsque $\beta = 0$? et la seconde lorsque $\delta = 0$? Comment évoluent les populations concernées ?

Question 7. Interpréter les quantités $\beta x(t)y(t)$ et $\delta x(t)y(t)$ dans les deux équations du modèle.

Question 8. En notant toujours x_k et y_k des approximations par la méthode d'Euler de $x(t_k)$ et $y(t_k)$ à l'instant t_k, déterminer les relations de récurrence que vérifient les suites $(x_k)_{0 \leqslant k \leqslant n}$ et $(y_k)_{0 \leqslant k \leqslant n}$.
En déduire une fonction `lotkaVolterra` qui prend en argument les effectifs initiaux x_0 et y_0 des proies et prédateurs, deux flottants a et b (correspondant aux bornes l'intervalle de résolution), un entier naturel n, et quatre flottants représentant les paramètres α, β, δ et γ et qui renvoie la liste des instants $(t_k)_{0 \leqslant k \leqslant n}$ ainsi que les listes des approximations $(x_k)_{0 \leqslant k \leqslant n}$ et $(y_k)_{0 \leqslant k \leqslant n}$.

Question 9. Représenter graphiquement les effectifs $x(t)$ et $y(t)$ des populations de proies et de prédateurs dans le cas où $\alpha = \frac{2}{3}$, $\beta = \dfrac{4}{3}$, $\delta = \gamma = 1$ lorsque $t \in [0, 50]$ et $(x_0, y_0) \in [1, 2]^2$.
On pourra utiliser la fonction `subplot` du module `matplotlib.pyplot` pour afficher et séparer les effectifs des proies et des prédateurs en fonction du temps.

Question 10. Avec les mêmes paramètres qu'à la question précédente, représenter graphiquement l'effectif des prédateurs en fonction de celui des proies. Commenter.
On pourra faire varier les conditions initiales et afficher les résultats sur le même graphique.