

Équations différentielles

L'objectif de ce TP est de mettre en œuvre la méthode d'Euler et de la comparer à d'autres schémas numériques (Heun et RK4) pour la résolution approchée d'équations différentielles ordinaires.

Les questions de ce TP sont à traiter dans le fichier .py associé, dont chaque cellule est numérotée ; le sujet indique simplement, pour chaque question, le numéro de la cellule concernée.

Présentation de la méthode d'Euler

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 :

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

Étape 1 : on subdivise l'intervalle. On ne cherche pas à calculer $y(t)$ pour tout $t \in [a, b]$, mais seulement en un nombre fini de $n + 1$ instants régulièrement espacés :

$$\forall k \in \llbracket 0, n \rrbracket, t_k = a + kh \quad \text{où} \quad h = \frac{b - a}{n}.$$

Le réel h s'appelle le **pas** de la subdivision. On a bien $t_0 = a$ et $t_n = b$.

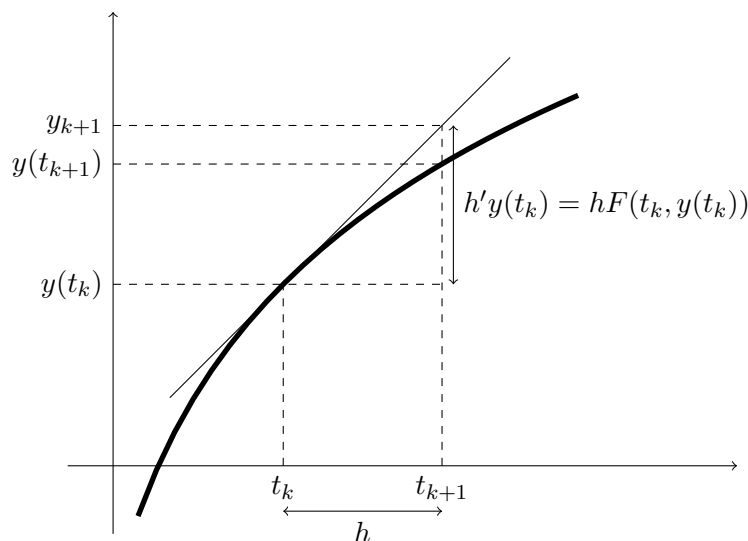
Étape 2 : on décide d'une notation pour une suite approchée. On note $(y(t_k))_{0 \leq k \leq n}$ les valeurs exactes et $(y_k)_{0 \leq k \leq n}$ les valeurs approchées, avec $y_0 = y(a)$ exact.

Étape 3 : relation de récurrence. Par définition du nombre dérivé, pour h petit, $y'(t) \approx \frac{y(t+h) - y(t)}{h}$, donc :

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

On obtient le schéma numérique d'Euler explicite :

$$\begin{cases} y_0 = y(a) \\ \forall k \in \llbracket 0, n-1 \rrbracket, y_{k+1} = y_k + hF(t_k, y_k). \end{cases}$$



La courbe pleine est la solution exacte ; le segment est la tangente en t_k suivie par la méthode d'Euler. On approche donc $y(t_{k+1})$ par y_{k+1} , obtenu en suivant la tangente.

1. Le terme *ordinaire* est utilisé par opposition au terme *équation aux dérivées partielles*.

1 Premier exemple : l'équation logistique

On cherche à résoudre de manière approchée par la méthode d'Euler l'équation logistique sur $[0, 10]$ avec $y(0) = 1$, $r = 0,5$ et $K = 10$:

$$y'(t) = r y(t) \left(1 - \frac{y(t)}{K}\right) \quad (1)$$

On décide de subdiviser l'intervalle $[0, 10]$ en 1000 intervalles.

Question 1. Que valent ici a , b , y_0 et n ? Quelle relation de récurrence vérifie la suite $(y_k)_{0 \leq k \leq n}$? Remplir et exécuter la cellule **Q1** du fichier à compléter.

Question 2. Compléter le code de la cellule **Q2** de manière à définir une fonction `euler_logistique` qui renvoie le couple $(\mathbf{t}, \mathbf{y})$ (dans cet ordre) où $\mathbf{t}$ désigne la liste des $(t_k)_{0 \leq k \leq n}$ et $\mathbf{y}$ désigne la liste des $(y_k)_{0 \leq k \leq n}$ obtenus par résolution par la méthode d'Euler de l'équation logistique sur $[0, 10]$.

Pour construire la liste `ord` des images par une fonction `f` à partir d'une liste `abscisses`, on peut utiliser la définition de liste par compréhension :

```
ord = [f(x) for x in abscisses]
```

Si la fonction `f` n'est définie qu'à l'aide des opérateurs mathématiques de base $(+, -, *, /, **)$ et de fonctions du module `numpy`, alors on peut écrire :

```
ord = f(abscisses) # attention : ord n'est pas une liste, mais de type numpy.array
```

On rappelle aussi que la syntaxe pour représenter graphiquement un nuage de points définis les listes `x` et `y` de leurs abscisses et ordonnées respectives

```
import matplotlib.pyplot as plt
plt.plot(x,y,label="legende") # on peut changer le style des points avec marker="+"
plt.legend(loc = "best") # optionnel
plt.show()
```

Question 3. On admet que la fonction

$$f : t \mapsto \frac{K y_0 e^{rt}}{K + y_0 (e^{rt} - 1)}$$

est solution de l'équation logistique (1) avec la condition initiale fixée. Compléter le code de la cellule **Q3** afin représenter graphiquement la fonction f ainsi que la solution obtenue par la méthode d'Euler à la question précédente (en fixant $r = 0,5$, $K = 10$).

Question 4. Exécuter le code de la cellule **Q4** et observer le comportement de la solution approchée par la méthode d'Euler lorsqu'on modifie la valeur de n .

Expliquer géométriquement comment est construit le point de coordonnées (t_1, y_1) par la méthode d'Euler.

2 Deuxième exemple : vers une résolution généralisée

Question 5. 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 $[a, b]$ avec condition initiale $y(a) = y_0$. La fonction devra renvoyer un couple (t, y) de deux listes `t` et `y` de longueur `n+1`.

Dans la suite de ce paragraphe, on cherche à représenter graphiquement la solution exacte et les solutions approchées par la méthode d'Euler de l'équation différentielle avec condition initiale :

$$\begin{cases} \forall t \in [0, 5], y'(t) + y(t) = -5 \sin(5t)e^{-t} \\ y(0) = 1. \end{cases} \quad (2)$$

Question 6. Identifier une fonction $F : (t, y) \mapsto F(t, y)$ permettant d'écrire l'équation différentielle ci-dessus sous la forme $y'(t) = F(t, y(t))$.

$F : (t, y) \mapsto$

Compléter alors le code de la cellule **Q6** définissant la fonction `F`.

Question 7. Compléter le script de la cellule **Q7** afin de résoudre de manière approchée l'équation différentielle (2) pour différentes valeurs de n (10, 30 et 100), à l'aide de la fonction `euler`, puis d'afficher ces solutions approchées.

Question 8. On admet que la fonction $f : t \mapsto \cos(5t)e^{-t}$ est l'unique solution de (2). Ajouter à la représentation graphique précédente le graphe de la fonction f .

3 Comparaison avec d'autres méthodes

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 :

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

- La **méthode RK₄ de Runge-Kutta** est définie par :

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

Question 9. En s'inspirant du code écrit à la question 5, écrire deux fonctions `heun` et `rk4` implémentant les méthodes de Heun et RK_4 . On conservera les mêmes arguments, dans le même ordre, que ceux de la fonction `euler`.

Question 10. Représenter graphiquement la solution exacte de l'équation différentielle (2) ainsi que les solutions approchées obtenues par la méthode d'Euler, de Heun et RK_4 de Runge-Kutta pour $n = 50$.

Question 11. Exécuter la cellule **Q11** et comparer les différentes méthodes en jouant sur le nombre de points des solutions approchées.