

TP6

Méthode d'Euler

1 Équations différentielles d'ordre 1

Considérons l'équation différentielle $\begin{cases} y(a) = y_0 \\ y'(t) = f(t, y(t)) \end{cases}$ sur un intervalle $I = [a; b]$.

Grâce à la méthode d'Euler, on approxime la solution en des valeurs t_k ayant pour image $y(t_k)$.

La méthode consiste à approximer les valeurs $y(t_k)$ par les valeurs y_k de la suite définie par récurrence par $\begin{cases} t_{k+1} = t_k + h \\ y_{k+1} = y_k + h \times f(t_k, y_k) \end{cases}$.

Exercice 1.1. On considère l'équation différentielle $\begin{cases} y(1) = 1 \\ ty'(t) - 2y(t) = 0 \end{cases}$ sur $[1; 2]$.

1. Quelle est la solution exacte ?
2. Rédiger une fonction sous Python qui :
 - Reçoit a, b et n en paramètres d'entrée.
 - Stocke les t_k dans une liste T .
 - Stocke les y_k dans une liste Y
 - Trace la courbe approximant la solution exacte dans un repère (utiliser l'annexe)
3. Modifier la fonction précédente pour tracer la solution exacte dans le même repère.
4. Tracer plusieurs courbes en augmentant la valeur de n . Que constate-t-on ?

Exercice 1.2. On considère l'équation différentielle

$$\begin{cases} y(1) = 0 \\ y' - \frac{3y}{x} - x = 0 \end{cases} \text{ sur } [1; 2].$$

1. Quelle est la solution exacte ?
2. Quel est le pourcentage d'erreur pour l'image de 2 si $n = 20$? si $n = 100$?
3. Quelles conclusions en tirer ?

Exercice 1.3. On considère l'équation différentielle

$$\begin{cases} y(1) = 0 \\ y' = \frac{1}{x} \end{cases} \text{ sur } [1; 2].$$

1. Quelle est la solution exacte ?
2. Appliquer la méthode d'Euler pour différentes valeurs de n .
3. Que dire du pourcentage d'erreur pour l'image de 2 ?

Exercice 1.4. Rédiger une fonction sous Python permettant d'approximer la primitive qui s'annule en a d'une fonction continue f sur un intervalle $[a; b]$.

Par exemple, comment approximer sur $[0; 2\pi]$ la primitive de $x \mapsto \frac{1}{1+x^2}$?

2 Équations différentielles d'ordre 2

Exercice 2.1. Application au second ordre

On souhaite résoudre l'équation différentielle du second ordre

$$(E) : \begin{cases} y(0) = 0 \text{ et } y'(0) = 1 \\ y'' + y = 0 \end{cases} \text{ sur } [0; 2].$$

Si l'on pose $Y = (y, y')$, alors :

$$\begin{aligned} y'' + y = 0 &\Leftrightarrow y'' = -y \Leftrightarrow Y' = (y', y'') = (y', -y) \\ &\Leftrightarrow Y' = F(Y) \text{ si } F(y_1, y_2) = (y_2, -y_1) \end{aligned}$$

$$\text{Alors } (E) \Leftrightarrow \begin{cases} Y(0) = (0, 1) \\ Y' = F(Y) \end{cases}$$

La méthode d'Euler peut encore s'appliquer ici. On reprend la méthode d'Euler connue avec quelques modifications puisque Y est un couple et F une application de deux variables.

1. La liste des valeurs approchées n'est plus une liste de réels mais une suite de couples de réels. Nous la noterons **liste_y**.

(a) Rédiger une fonction **F(Y)** recevant en paramètre une liste de deux éléments et renvoyant la liste de deux éléments voulue.

(b) Traduire l'égalité $Y_{k+1} = Y_k + h \times F(Y_k)$ sous forme d'égalité de deux 2-uplets.

(c) Rédiger la fonction **euler(a, b, y0, y0p, n, F)** correspondant à l'équa-

$$\text{tion } (E) : \begin{cases} Y(a) = (y0, y0p) \\ Y' = F(Y) \end{cases} \text{ sur } [a; b] \text{ renvoyant :}$$

- la liste des abscisses
- La liste **liste_y**.

2. Construire avec **Matplotlib** la représentation graphique de la solution approchée et celle de la solution exacte.

Remarque : Les ordonnées pour la solution approchée sont les premiers éléments de chaque couple dans la liste des y renvoyée par la fonction *euler*

ANNEXE

A l'aide de la bibliothèque Matplotlib, et de son instruction plot(), il est possible de tracer des courbes qui relient des points dont les abscisses et ordonnées sont fournies dans des listes. Pour cela, on utilise la suite d'instructions suivantes :

```
import matplotlib.pyplot as plt
plt.clf()           # Efface le graphique précédent
plt.plot(list_x,list_y) # Trace la courbe
plt.ylim(ymin,ymax) # Définit la plage des ordonnées
plt.show()         # Affiche la courbe
```

Par exemple pour :

```
list_x=[0,1,2,3]
list_y=[4,2,8,5]
ymin = 0
ymax = 10
```

On obtient la courbe ci-contre :

