Chapitre OPP3 : Méthode d'Euler

Plan du cours

I. Principe de la méthode 1
- I. 1 - Discrétisation du temps . . . 1
- I. 2 - Approximation de la dérivée . . . 2
- I. 3 - Schéma d'Euler . . . 2

II. Implémentation en Python 4
- II. 1 - Programme . . . 4
- II. 2 - Influence du pas de temps . . . 5

III. Équations différentielles d'ordre 2 6

Introduction

▷ En physique, l'évolution temporelle d'un système est très souvent décrite par une **équation différentielle** : charge d'un condensateur, chute d'un objet avec frottements, oscillations d'un ressort... On sait résoudre analytiquement quelques cas simples (équations linéaires à coefficients constants), mais dès que l'équation se complique (excitation de forme quelconque, frottements non linéaires, etc.), on doit recourir à une **résolution numérique**.

Dans tout ce chapitre, la fonction inconnue y dépend du temps t, et on s'intéresse d'abord aux équations différentielles d'ordre 1 de la forme

$$y'(t) + a\,y(t) = b \qquad \text{avec la condition initiale} \qquad y(t_0) = y_0.$$

Problématique :
Comment calculer, à l'aide d'un ordinateur, une solution approchée d'une équation différentielle ?

I. Principe de la méthode

Définition n°1 : Méthode d'Euler
La méthode d'Euler (explicite) est une méthode de **résolution numérique** des équations différentielles d'ordre 1, c'est-à-dire de la forme

$$y'(t) = f\big(t, y(t)\big) \qquad \text{avec} \qquad y(t_0) = y_0.$$

Pour l'équation $y' + ay = b$, on a simplement $f(t, y) = b - a\,y(t)$
On cherche à obtenir une **approximation** des valeurs de y à différents instants.

▷ La méthode d'Euler repose sur deux idées : la **discrétisation du temps** et l'**approximation de la dérivée**.

I. 1 - Discrétisation du temps

▷ L'ordinateur ne peut pas stocker la valeur de y à **chaque** instant (il y en a une infinité) : on est contraint de **discrétiser** le temps.

Définition n°2 : Discrétisation

On note h le pas de temps (ou de résolution) choisi par l'utilisateur.
Les instants de calculs sont : $t_i = t_0 + hi$ avec $i \in [\![0; N]\!]$.

On note y_i la valeur approchée de $y(t_i)$ par cette méthode.

Remarque

Pour étudier le système entre t_0 et t_f, il faut $N = \dfrac{t_f - t_0}{h}$ pas : plus h est petit, plus on calcule de points.

I. 2 - Approximation de la dérivée

▷ Par définition, la dérivée de y à l'instant t s'écrit $\dfrac{dy}{dt}(t) = y'(t) = \lim\limits_{h \to 0} \dfrac{y(t+h) - y(t)}{h}$.

▷ L'idée d'Euler est de **ne pas** faire tendre h vers 0, mais de garder un pas h petit mais fini.

Propriété n°1 : Approximation de la dérivée

Pour h "petit", on a : $y'(t) \simeq \dfrac{y(t+h) - y(t)}{h}$

On $t_{i+1} = t_i + h$ d'où $y'(t_i) = \dfrac{y(t_i + h) - y(t_i)}{t_{i+1} - t_i} \Leftrightarrow \boxed{y'(t_i) = \dfrac{y(t_{i+1}) - y(t_i)}{t_{i+1} - t_i}}$

L'approximation est d'autant meilleure que h est petit.

I. 3 - Schéma d'Euler

▷ On isole $y(t_{i+1})$ dans l'approximation précédente : $y(t_{i+1}) - y(t_i) = y'(t_i)(t_{i+1} - t_i)$ ($t_{i+1} - t_i = h$)

$$\Leftrightarrow \boxed{y(t_{i+1}) = y(t_i) + h\,y'(t_i)}$$

et la dérivée $y'(t_i)$ est donnée par l'équation différentielle elle-même : $y'(t_i) = f(t_i, y(t_i))$. On ne connaît pas les valeurs exactes $y(t_i)$: on les remplace par les valeurs approchées y_i calculées de proche en proche.

Propriété n°2 : Relation de récurrence d'Euler

Les valeurs approchées de y_i sont définies par la suite récurrente :

$$\begin{cases} y_0 \\ y_{i+1} = y_i + h\, f(t_i, y_i) \end{cases} \quad \text{avec} \quad f(t_i, y_i) = y'(t_i)$$

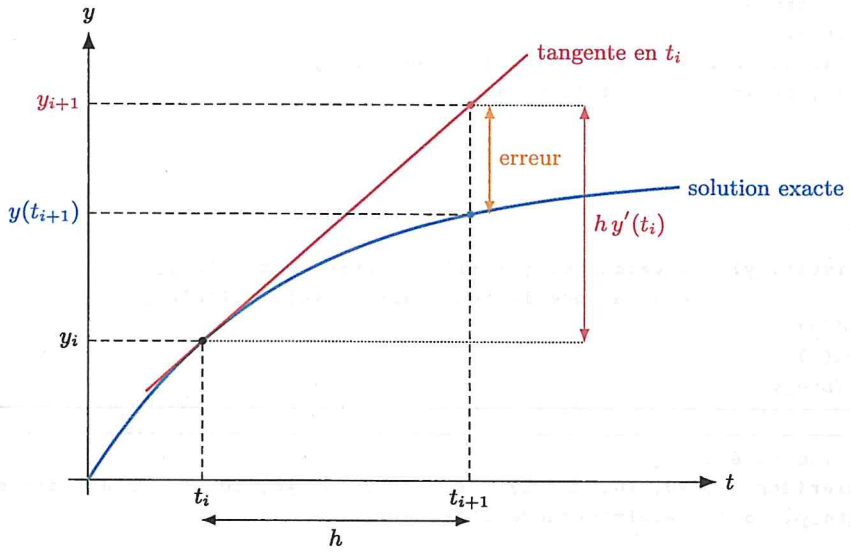
Pour l'équation $y' + ay = b$ alors $f = b - ay$ d'où :

$$y_{i+1} = y_i + h(b - ay_i)$$

$$\Leftrightarrow \underline{y_{i+1} = (1 - ah)y_i + hb}$$

Pour déterminer y_{i+1}, il faut connaître y_i et h.

▷ Interprétation graphique : à chaque pas, on avance le long de la **tangente** à la courbe au point courant, au lieu de suivre la courbe elle-même.



Application n°1 : Charge d'un condensateur

Un condensateur, initialement déchargé, se charge à travers une résistance sous une tension E constante. La tension u à ses bornes vérifie

$$\frac{\mathrm{d}u}{\mathrm{d}t} + \frac{u}{\tau} = \frac{E}{\tau} \qquad \text{avec} \qquad u(0) = 0.$$

Q 1. Identifier a et b par rapport à la forme générale $y' + ay = b$.

Q 2. Écrire la relation de récurrence d'Euler donnant u_{i+1} en fonction de u_i.

Q 3. On choisit $h = \tau/2$. Calculer u_1 et u_2, puis comparer à la solution exacte $u(t) = E\left(1 - \mathrm{e}^{-t/\tau}\right)$ aux mêmes instants.

Remarque

L'erreur commise à chaque pas vient du fait que la courbe n'est pas une droite : elle est d'autant plus faible que h est petit. On montre que l'erreur globale de la méthode d'Euler est **proportionnelle à** h : diviser le pas par 10 divise (à peu près) l'erreur par 10. On dit que la méthode est d'**ordre 1**.

II. Implémentation en Python

II. 1 - Programme

▷ On propose ci-dessous une implémentation en très peu de fonctions. Les capacités numériques du programme peuvent être découpées en davantage d'étapes.

```python
import numpy as np
import matplotlib.pyplot as plt

## Définition des paramètres et de la dérivée
a = ...
b = ...

def derive(t, y):
    """Renvoie y'(t), calculée à l'aide de l'équation différentielle."""
    return b - a*y

## Implémentation de la méthode d'Euler
def euler(derive, y0, t0, tf, h):
    # derive : fonction dérivée définie auparavant
    # y0 : condition initiale
    # t0 : instant initial
    # tf : instant final jusqu'auquel on veut les valeurs de y
    # h  : pas de temps, choisi par l'utilisateur
    t = t0
    y = y0
    liste_t = [t0]
    liste_y = [y0]
    while t < tf:
        y = y + h*derive(t, y)    # calcul de y_(i+1) à partir de t_i et y_i
        t = t + h                 # on avance le temps après avoir calculé y
        liste_y.append(y)
        liste_t.append(t)
    return liste_t, liste_y
```

```python
## Tracé de la résolution numérique
liste_t, liste_y = euler(derive, y0, t0, tf, h)   # valeurs de y0, t0, tf, h à indiquer
plt.plot(liste_t, liste_y, 'o-', label='méthode d\'Euler')

## Tracé de la solution théorique (si elle est connue)
def fonction_theorique(t):
    return ...    # par exemple b/a + (y0 - b/a)*np.exp(-a*t)

t = np.array(liste_t)    # conversion en tableau numpy pour calculer sur tous les points
plt.plot(t, fonction_theorique(t), label='solution exacte')

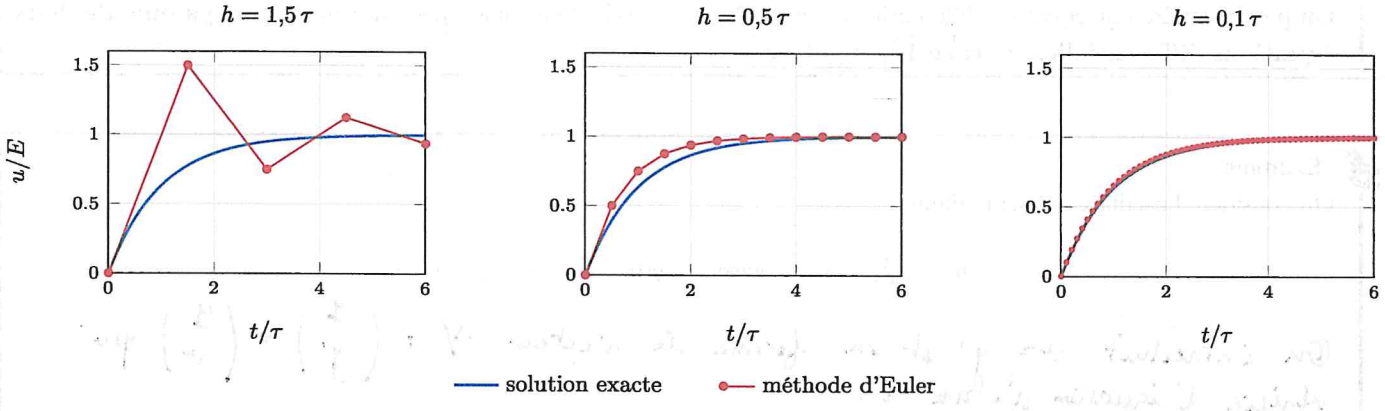
plt.xlabel('t (s)')
plt.ylabel('y')
plt.legend()
plt.grid()
plt.show()
```

Remarque

L'ordre des instructions dans la boucle est important : pour calculer y_{i+1}, on doit utiliser t_i et y_i. On met donc à jour y **avant** de mettre à jour t.

II. 2 - Influence du pas de temps

On reprend la charge du condensateur ($y = u/E$, t en unités de τ) pour trois pas de temps différents.



Propriété n°3 : Influence du pas de temps

Plus le pas h est petit (donc plus le nombre de points est grand), plus la résolution numérique est **précise**.
En contrepartie, le **temps de calcul** augmente, et pour un pas trop grand la méthode peut même donner des résultats aberrants (oscillations non physiques, divergence).

Application n°2 : Stabilité de la méthode

On reprend l'équation $y' + ay = b$ avec $a > 0$, dont la solution exacte tend vers $y_\infty = b/a$.

Q 1. Montrer que la suite $z_i = y_i - \dfrac{b}{a}$ vérifie $z_{i+1} = (1 - ah)\, z_i$. En déduire y_i en fonction de i, y_0, a, b et h.

Q 2. À quelle condition sur h la suite (y_i) converge-t-elle vers b/a ? À quelle condition le fait-elle sans osciller ?

Q 3. Interpréter les trois graphes précédents (pour la charge du condensateur, $a = 1/\tau$). Que se passerait-il pour $h = 2{,}5\,\tau$?

Remarque

En pratique, on choisit un pas h **très petit devant le temps caractéristique** du système (ici τ), par exemple $h \approx \tau/100$, puis on vérifie que diminuer encore h ne modifie plus le résultat de manière visible.

III. Équations différentielles d'ordre 2

Propriété n°4 : Équation d'ordre 2
On peut résoudre une équation différentielle d'ordre 2 par la méthode d'Euler en se ramenant à un **système de deux équations différentielles d'ordre 1**, portant sur y et sur $v = y'$.

Exemple
On considère l'oscillateur harmonique

$$y'' + \omega_0^2 y = 0 \qquad \text{avec} \qquad y(0) = y_0 \quad \text{et} \quad y'(0) = 0.$$

On introduit $v = y'$ et on forme le vecteur $Y = \begin{pmatrix} y \\ y' \end{pmatrix} = \begin{pmatrix} y \\ v \end{pmatrix}$ qui vérifie l'équation d'ordre 1 :

$$\begin{pmatrix} y \\ v \end{pmatrix}' = \begin{pmatrix} v \\ -\omega_0^2 y \end{pmatrix} \quad \text{avec} \quad \begin{pmatrix} y(0) \\ y'(0) \end{pmatrix} = \begin{pmatrix} y_0 \\ 0 \end{pmatrix}$$

On applique le même principe aux deux composantes simultanément :

$$\begin{cases} y_{i+1} = y_i + h\, v_i \\ v_{i+1} = v_i + h\left(-\omega_0^2 y_i\right) \end{cases} \Leftrightarrow \begin{cases} y_{i+1} = y_i + h v_i \\ v_{i+1} = v_i - h\omega_0^2 y_i \end{cases}$$

Remarque
Pour l'oscillateur harmonique, la méthode d'Euler explicite ne conserve pas l'énergie : l'amplitude calculée est multipliée par $\sqrt{1+\omega_0^2 h^2}$ à chaque pas et augmente donc lentement au cours du temps. Cet effet est d'autant plus faible que h est petit devant la période $2\pi/\omega_0$. L'implémentation en Python sera vue en TP.

Corrigé des applications du chapitre OPP3

Application n° 1 : Charge d'un condensateur

Q1) $y' + ay = b$ avec $a = \frac{1}{\tau}$ et $b = \frac{E}{\tau}$

Q2) Approximation de la dérivée : $u'(t_i) \simeq \frac{u(t_{i+1}) - u(t_i)}{t_{i+1} - t_{i-1}}$

D'où $u'(t_i)(t_{i+1} - t_i) = u(t_{i+1}) - u(t_i)$

$\Leftrightarrow u_{i+1} = u_i + f(t_i, u_i)\,h$ avec $u'(t_i) = f(t_i, u_i)$

Or $u' + au = b$ d'où $u'(t_i) = b - a u(t_i)$

Donc $u_{i+1} = u_i + (b - a u(t_i))h$

$\Leftrightarrow u_{i+1} = (1 - ah)u_i + bh$

Q3) $u_1 = \left(1 - \frac{a\tau}{2}\right)u_0 + \frac{b\tau}{2} \Leftrightarrow u_1 = \frac{E}{2}$

$u_2 = \left(1 - \frac{a\tau}{2}\right)u_1 + \frac{b\tau}{2} \Leftrightarrow u_2 = \left(1 - \frac{1}{2}\right)u_1 + \frac{E}{2}$

$\Leftrightarrow u_2 = \frac{E}{4} + \frac{E}{2}$

$\Leftrightarrow u_2 = \frac{3E}{4}$

Par ailleurs, $u(t) = E(1 - e^{-t/\tau})$ (solution analytique)

d'où $u(\tau/2) = E(1 - e^{-1/2}) \simeq 0{,}39\,E \neq 0{,}50\,E$

$u(\tau) = E(1 - e^{-1}) \simeq 0{,}63\,E \neq 0{,}75\,E$.

Le pas $h = \frac{\tau}{2}$ est beaucoup trop élevé, peu précis.

Application n° 2 : Stabilité de la méthode

Q1) $y' + ay = b$, $a > 0$ et $y_\infty = \frac{b}{a}$.

Considérons $(z_i)_{i \in \mathbb{N}}$ tq $z_i = y_i - y_\infty$

Alors $z_{i+1} = y_{i+1} - y_\infty$.

Or $y_i' = \frac{y_{i+1} - y_i}{h} \Leftrightarrow y_{i+1} = h y_i' + y_i$ et $y_i' = b - a y_i$

Donc $y_{i+1} = y_i + h(b - a y_i)$

Finalement : $z_{i+1} = y_i + hb - ah y_i - y_\infty$

$= z_i - ah\left(y_i - \frac{b}{a}\right)$

$= z_i - ah z_i$

Donc $\boxed{z_{i+1} = (1 - ah) z_i}$

Q2) $(z_i)_{i \in \mathbb{N}}$ est une suite géométrique de raison $q = 1 - ah$.

D'où (z_i) converge ssi $|q| < 1 \Leftrightarrow |1 - ah| < 1$

$\Leftrightarrow -1 < 1 - ah < 1$

D'où $ah < 2 \Leftrightarrow \boxed{h < \frac{2}{a}}$.

$z_i = z_0 (1 - ah)^i = \left(y_0 - \frac{b}{a}\right)(1 - ah)^i$

Donc $y_i = \frac{b}{a} + \left(y_0 - \frac{b}{a}\right)(1 - ah)^i \xrightarrow[i \to +\infty]{} \frac{b}{a}$ OK.

Pas d'oscillations ssi $q \geq 0 \Leftrightarrow 1 - ah \geq 0 \Leftrightarrow ah \leq 1 \Leftrightarrow \underline{h \leq \frac{1}{a}}$

Q3) Pour $a = \frac{1}{\tau}$, alors $\frac{2}{a} = 2\tau$ et $\frac{1}{a} = \tau$.

- $h = 1{,}5\,\tau$: converge mais oscille.
- $h = 0{,}5\,\tau$: converge sans osciller.
- $h = 0{,}1\,\tau$: idem