

TP N°22 : Résolution des équations différentielles du premier ordre

1 Introduction

Après les équations numériques, les *équations différentielles*. Dans ce cours, nous étudions les *équations différentielles linéaires du premier ordre (EDL1)*. Les équations différentielles ont un lien évident avec l'intégration numérique. C'est donc sans surprise que l'on va utiliser dans cette section des méthodes proches de celles de la précédente : découpage de l'intervalle de résolution en n morceaux, puis approximation de la solution sur chacun de ces intervalles.

C'est le principe de base de la *méthode d'Euler*, qui est la principale méthode de résolution numérique d'équations différentielles, sur laquelle nous allons concentrer tous nos efforts.

Dans la suite, nous allons étudier la méthode d'Euler, méthode numérique permettant d'obtenir une approximation de la solution d'un tel *problème de Cauchy*. Et on termine par des exemples d'utilisation de la fonction de résolution approchée d'équations différentielles *odeint*.

2 Définitions

Définition 1. Une **équation différentielle** est une équation qui relie une fonction f à une ou plusieurs de ses dérivées. Exemple : $f(x) = \alpha f'(x) + \beta f''(x)$.

Remarque 1. Pour simplifier l'écriture des équations différentielles, on remplace couramment $f(x)$ par y et $f'(x)$ par y' . Exemple : $f(x) = \alpha f'(x) + \beta$ devient $y = \alpha y' + \beta$.

Définition 2. Une **équation différentielle** est une équation dont l'inconnue est une fonction y (réelle ou complexe, même si nous traiterons surtout le cas réel dans ce chapitre), et faisant intervenir les dérivées successives $y', y'', \dots, y^{(n)}$ de la fonction y . L'équation est dite **d'ordre n** si la dérivée d'ordre le plus élevé de y apparaissant dans l'équation est $y^{(n)}$.

Définition 3. Une équation différentielle est **linéaire** si elle s'écrit sous la forme : $a_n y^{(n)} + a_{n-1} y^{(n-1)} + \dots + a_1 y' + a_0 y = b$ où $a_0, a_1, \dots, a_n$ et b sont des fonctions.

Définition 4. La fonction b est appelé le **second membre**. Si b est *nulle*, alors l'équation est dite **homogène** ou **sans second membre**. Si $a_0, a_1, \dots, a_n$ sont des *constantes*, on parle d'équation différentielle linéaire à **coefficients constants**.

Définition 5. Une **courbe intégrale** associée à une équation différentielle est une courbe représentative de la fonction solution y de l'équation différentielle.

Remarque 2. Les équations différentielles considérées doivent être écrites sous la forme : $y' = F(y, t)$ où F est un fonction à deux variables y et t .

Par exemple, pour l'équation différentielle $y' + 2ty = 1$ on définit $F(y, t) = 1 - 2ty$ ce qui s'écrit avec *Python* :

```
def F(y, t):
    return 1-2*t*y
```

Remarque 3. Même si, comme c'est parfois le cas, on écrit l'équation sous la forme : $y'(t) + 2ty(t) = 1$ il ne faut surtout pas définir la fonction F en écrivant :

```
def F(y, t):
    return 1-2*t*y(t) #INCORRECT
```

Remarque 4. En mathématiques et en physique, l'habitude est plutôt de considérer les équations différentielles sous la forme $y' = F(t, y)$ et non $y' = F(y, t)$. Mais en Python, la fonction *odeint*, que nous verrons par la suite, utilise l'écriture $y' = F(y, t)$. Par souci de simplicité, nous nous conformerons à cette écriture dans ce cours.

Dans le cadre de notre cours on cherche à résoudre un problème numérique simple dit *problème de Cauchy*.

3 Cadre du problème de Cauchy

Un *problème de Cauchy* est un problème constitué d'une équation différentielle dont on recherche une solution vérifiant une certaine condition initiale. $\{ \begin{matrix} y' = F(t, y) \\ y(t_0) = y_0 \end{matrix} \}$ où :

- y (est l'inconnue) est une fonction dérivable définie sur un intervalle I de $\mathbb{R}$;
- $t_0 \in I$;

On possède donc une équation différentielle et une condition initiale. Le théorème de Cauchy Lipschitz assure l'existence et l'unicité d'une solution à ce problème.

4 Méthode d'Euler

La méthode d'Euler est une procédure numérique pour résoudre par approximation des équations différentielles du premier ordre avec une condition initiale. C'est la plus simple des méthodes de résolution numérique des équations différentielles.

On considère une équation différentielle d'ordre 1 avec condition initiale (problème de Cauchy) : $\{ \begin{matrix} y' = F(y, t) \\ y(t_0) = y_0 \end{matrix} \}$

On cherche à résoudre le problème de Cauchy sur un intervalle $I = [a, b]$.

Soit $n \in \mathbb{N}^*$, on subdivise l'intervalle $[a, b]$ en n petits segments de même longueur $h = \frac{b-a}{n}$.

On pose alors pour $0 \leq k \leq n$: $t_0 = a$ $t_1 = t_0 + h$ $t_2 = t_0 + 2h$... $t_k = a + kh$... $t_n = t_0 + nh = b$

On part de t_0 . On approche le petit morceau de courbe entre t_0 et t_1 par la tangente à la courbe au point d'abscisse t_0 .

La méthode d'Euler sert à déterminer approximativement les valeurs prises par y aux instants $t_0, t_1, \dots, t_{n-1}$.

Nous utiliserons les notations suivantes :

- $y(t_0), y(t_1), \dots, y(t_{n-1})$ sont les valeurs exactes (on connaît seulement $y(t_0)$) ;
- $y_0, y_1, \dots, y_{n-1}$ sont les valeurs approchées fournies par la méthode d'Euler.

On a alors :

$$\begin{aligned} \frac{y(t_1) - y(t_0)}{h} &\approx y'(t_0) \\ &\approx F(y(t_0), t_0) \\ y(t_1) &\approx y_0 + hF(y(t_0), t_0) \end{aligned}$$

On pose : $y_1 = y_0 + hF(y_0, t_0)$. Le nombre y_1 est une approximation de la valeur exacte $y(t_1)$.

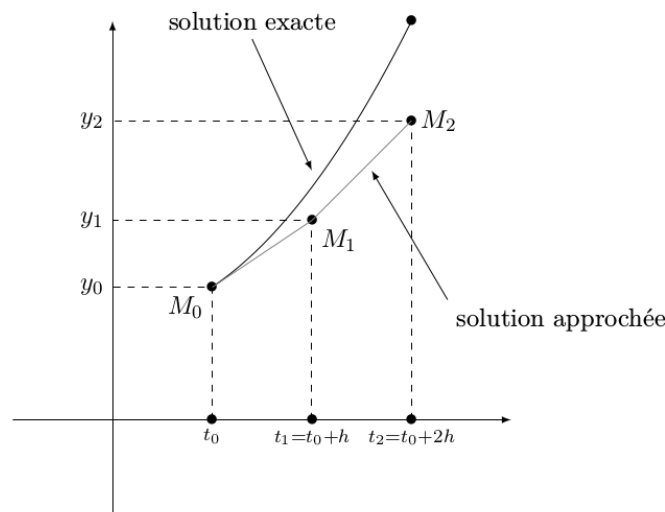
On recommence le procédé à partir du point $M_1(t_1, y_1)$...

De manière générale, on pose :

$$\forall k \in 0, \dots, n-1, y_{k+1} = y_k + hF(y_k, t_k)$$

Ce qui amène à construire successivement les points $M_k(t_k, y_k)$.

La ligne polygonale reliant ces points est alors une approximation de la courbe représentative de la solution.



Cette méthode de résolution approchée, la seule à notre programme, s'appelle la méthode d'Euler. Il existe de nombreuses autres méthodes qui peuvent être proposées dans le cadre d'un problème.

4.1 Comment programmer la méthode d'Euler?

Dans ce programme, on donne la valeur du pas h et on vérifie à chaque itération que $a + t_k h \leq b$.

```

1 def euler(F, a, b, y0, h=1e-2):
2     """Fournit une solution approchée à l'équation y'=F(y,t), y(a)=y0
3     sur l'intervalle [a,b] par la méthode d'Euler, le pas utilisé est h, par défaut 0.01
4     La solution renvoyée est une liste de valeurs pour les_t et une liste de valeurs pour les_y"""
5     y = y0
6     t = a
7     les_y = [y0] # la liste des valeurs renvoyées
8     les_t = [a]
9     while t+h <= b:
10         #les_y = [y0,...,y_k], les_t = [t0,...,t_k]
11         y += h * F(y, t)
12         les_y.append(y)
13         t += h
14         les_t.append(t)
15         # à la fin de la boucle, t contient a + nh ≈ b
16     return les_t, les_y

```

Remarque 5. Si le nombre de subdivisions n nous est donné à la place de h , on pourra préférer écrire la fonction avec une boucle while, en utilisant la fonction linspace du module numpy.

4.2 Exemple Python

On considère l'équation $y' = y$ avec la condition initiale $y(0) = 1$. La solution est la fonction exponentielle, ce qui permet de comparer les résultats obtenus.

On va travailler sur l'intervalle $[0,1]$ en considérant tout d'abord un pas de 0.25 puis un pas de 0.1. La fonction exponentielle est représentée en pointillés pour comparer. F doit être définie comme une fonction de deux variables même si ici elle ne dépend que de y .

```

import numpy as np
import matplotlib.pyplot as plt
def F(y,t):
    return y

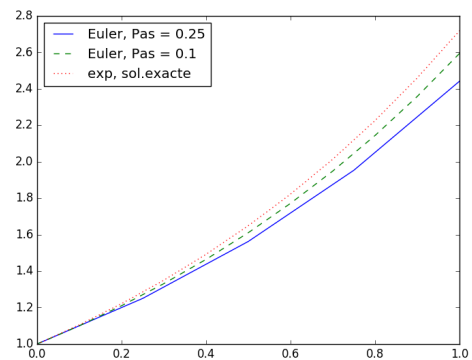
t,y = euler(F,0,1,1,0.25)
#ligne bleue continue
plt.plot(t,y,'b-',label='Euler, Pas = 0.25')

t,y = euler(F,0,1,1,0.1)
#ligne verte discontinue
plt.plot(t,y,'g--',label='Euler, Pas = 0.1')

t = np.linspace(0,1,11)
#ligne rouge pointillée
plt.plot(t,np.exp(t),'r:',label='exp,sol.exacte')

plt.legend(loc='upper left')
plt.show()

```



Exercice 1

On considère le problème de Cauchy : $\begin{cases} \frac{dy}{dt} = -ty + \cos(t) \\ y(0) = 1 \end{cases}$

Représenter graphiquement la solution de ce problème sur l'intervalle $[0,5]$.

5 La fonction odeint du package scipy.integrate

La bibliothèque `scipy` fournit une fonction de résolution approchée et automatique des équations différentielles ordinaires. Il s'agit de la fonction `odeint`, acronyme de `ordinary differential equations integrating`. Pour pouvoir l'utiliser, il faut d'abord l'importer :

```
from scipy.integrate import odeint
```

On appelle `odeint` sous la forme `odeint(F, y0, t)` où F est la fonction qui intervient dans l'équation différentielle, y_0 est la valeur initiale et t est le vecteur contenant les valeurs $[t_0 = a, t_1, \dots, t_n = b]$.

Après avoir défini les objets de l'équation que l'on veut résoudre, on demande la résolution de l'équation différentielle au moyen de la commande suivante : `y = odeint (F , y0 , t )`

Le résultat y est un tableau des approximations de $y(t)$ correspondant aux différents instants du tableau t .

6 Exemples de résolution approchée d'ED1O.

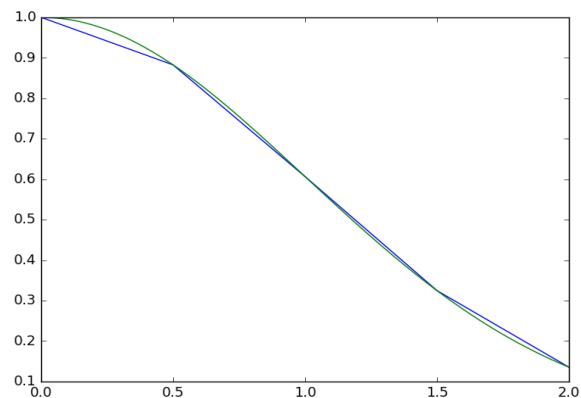
Exemple 1. On considère l'équation différentielle d'ordre 1 avec condition initiale : $\begin{cases} \frac{dy}{dt} = -ty \\ y(0) = 1 \end{cases}$

On veut calculer des valeurs approchées de y entre 0 et 2 pour 5 valeurs régulièrement espacées.

Remarque : Le tracé est superposé avec celui de la fonction $t \mapsto \exp(-t^2/2)$.

```
1  #ED
2  def F(y,t):
3      return -t*y
4  t=np.linspace(0,2,5)
5  y=odeint(F,1,t)

6  #exp
7  exp_t = np.linspace(0,t[4],1000)
8  exp_y = np.exp(-exp_t**2/2)
9  plt.plot(t,y,exp_t,exp_y)
10 plt.show()
```



Exemple 2. On veut résoudre sur l'intervalle $[0, 5]$ le problème de Cauchy : $\begin{cases} y' + 2ty = 1 \\ y(0) = 1 \end{cases}$

```
from math import *
from scipy.integrate import odeint
import numpy as np
import matplotlib.pyplot as plt
def F(y, t): # La fonction F associée à l'équation différentielle
    return 1-2*t*y
t = np.linspace(0,5,100) # Le vecteur t qui représente l'intervalle de temps considéré
y = odeint(F,1,t) # Le calcul du vecteur y avec la commande odeint
# Représentation graphique
plt.clf();plt.plot(t,y, 'b-');plt.xlabel("x"); plt.ylabel("y"); plt.show()
```

Exercice 2: Problème de cinétique chimique

On s'intéresse à une réaction chimique faisant disparaître un réactif A d'une solution. On suppose que la vitesse de disparition de A est proportionnelle à sa concentration (réaction d'ordre 1) : $\frac{d[A]}{dt} = -\alpha[A]$ où α est une constante s'exprimant en s^{-1} (la valeur $\frac{\ln 2}{\alpha}$ est un temps appelé temps de demi-réaction).

L'équation est bien de la forme $[A]' = F(t, [A])$ où $F : (t, y) \mapsto -\alpha y$

On suppose $\alpha = 1s^{-1}$ et au temps $t = 0$, $[A] = 1mol/l$.

Tracer la courbe représentative de l'évolution de la vitesse de disparition de A jusqu'à $t = 6s$.