Chapitre 13

Résolution approchée d'équations différentielles du 1er ordre

13.1 Equation différentielle du 1^{er} ordre

13.1.1 Formulation du problème

Soit ϕ une application réelle définie sur une partie de $\mathbb{R}^2$. Résoudre, ou intégrer, une équation différentielle d'ordre 1 de la forme :

$$y' = \phi(x, y) \qquad (E)$$

C'est déterminer toutes les application y définies et dérivables sur une réunion d'intervalles ouverts I de $\mathbb{R}$ et vérifiant :

$$\forall\, x \in I, \quad y'(x) = \phi(x, y(x))$$

• Exemples :

1. Primitivation : $y'(x) = a(x)$; ici $\phi(x,y) = a(x)$. Lorsque $x \mapsto a(x)$ est continue, les solutions sont toutes les primitives de a : $y(x) = \int a(x)dx$.
2. équation différentielle linéaire homogène du 1er ordre : $y'(x) = a(x).y(x)$; ici $\phi(x,y) = a(x).y$. Lorsque $x \mapsto a(x)$ est continue, les solutions sont toutes les applications : $x \mapsto k \times exp(\int a(x)dx)$; $k \in \mathbb{R}$.
3. équation différentielle linéaire du 1er ordre : $y'(x) = a(x).y(x) + b(x)$; ici $\phi(x,y) = a(x).y + b(x)$.
4. équations différentielles non-linéaires du 1er ordre. Par exemple : $y' = x\sin(y)$. On a souvent besoin ici d'une méthode de calcul approchée.

13.1.2 Existence et Unicité de solutions : Théorème de Cauchy-Lipschitz

• Une solution est <u>maximale</u>, si elle ne peut pas se prolonger en une solution.

Exemple : $x^3 y' = 2y$ a pour solution l'application $y : x \mapsto exp\left(-\frac{1}{x^2}\right)$ définie sur $\mathbb{R}^*$. C'est une solution non maximale : en effet $y : x \mapsto exp\left(-\frac{1}{x^2}\right)$ se prolonge sur $\mathbb{R}$ en une application dérivable en posant $y(0) = y'(0) = 0$. C'est une solution maximale.

• Résoudre une équation différentielle $y' = \phi(x, y)$ <u>avec condition initiale</u> $y_0 = y(x_0)$ qui peut s'écrire aussi (x_0, y_0) (avec (x_0, y_0) dans le domaine D de ϕ) consiste à déterminer une solution y à l'équation différentielle vérifiant $y(x_0) = y_0$. On parle aussi de <u>*Problème de Cauchy*</u>.

Définition. $\phi : D \subset \mathbb{R}^2 \longrightarrow \mathbb{R}$ est de classe C^1 si ses dérivées partielles :

$$(x,y) \longmapsto \frac{\partial \phi}{\partial x}(x,y) \qquad (x,y) \longmapsto \frac{\partial \phi}{\partial y}(x,y)$$

existent et sont continues.

• Théorème de Cauchy-Lipschitz (version faible) : Si ϕ est de classe C^1 sur un domaine ouvert $D =]x_a, x_b[\times]y_a, y_b[$ de $\mathbb{R}^2$, l'équation différentielle $y' = \phi(x, y)$ avec la condition initiale $y_0 = y(x_0)$ (où $(x_0, y_0) \in D$) admet une unique solution maximale. Toute autre solution est une restriction de la solution maximale.

13.2 Méthodes d'Euler-Cauchy

13.2.1 Principe de la méthode d'Euler-Cauchy

Soit l'équation différentielle $f'(x) = \phi(x, f(x))$ où ϕ est définie sur $I \times \mathbb{R}$ avec I un intervalle ouvert non-vide de $\mathbb{R}$.

Soit un segment $[a, b]$ inclus dans I et la condition initiale $(x_0, y_0) = (a, f(a))$. La méthode d'Euler-Cauchy consiste à considérer une subdivision régulière de

$[a,b]$ en n segments, soit $n+1$ points $x_k = a + k\frac{b-a}{n}$ et à calculer de proche en proche une valeur approchée de $y_k = f(x_k)$ en approchant la fonction f par sa tangente au point d'abscisse x_k.

La suite $(y_k)_{0\leqslant k\leqslant n}$ est ce que l'on entend par une résolution approchée du problème de Cauchy.

C'est à dire, une fois y_k déterminé, on trouve y_{k+1} par :

$$y_{k+1} = y_k + \int_{x_k}^{x_{k+1}} f'(x)dx = y_k + \int_{x_k}^{x_{k+1}} \phi(x, f(x))dx$$
$$\simeq y_k + f'(x_k)(x_{k+1} - x_k) \simeq \boxed{y_k + \phi(x_k, y_k).\frac{b-a}{n} \simeq y_{k+1}}$$

13.2.2 Schéma d'Euler-Cauchy explicite à l'ordre 1

• La méthode d'Euler-Cauchy consiste à remplacer dans l'équation différentielle : $\boxed{f'(x) = \phi(x, f(x))}$ le nombre dérivée $f'(x)$ en x, par :

$$\boxed{f'(x) \approx \frac{f(x+dx) - f(x)}{dx}}$$

Pour obtenir le **schéma d'Euler explicite à l'ordre 1 :**

$$\boxed{f(x+dx) \approx f(x) + f'(x).dx}$$

• Donnée une subdivision régulière $(x_k)_{0\leqslant k\leqslant n}$ de l'intervalle $[a,b]$ d'intégration, dont les valeurs sont contenues dans le tableau :

```
X = np.linspace(a,b,n+1)
```

La solution approchée consistera en les valeurs prises par f aux points de (x_k),
contenues dans un tableau de longueur $n+1$:

```
Y = np.empty(n+1)
```

`Y[0] = y0`	Condition intiale
`Y[k+1] = Y[k] + (b-a)/n * Phi(X[k], Y[k])`	Pour tout $k \in [[0,n]]$

13.2.3 Code d'une fonction implémentant la méthode d'Euler-Cauchy

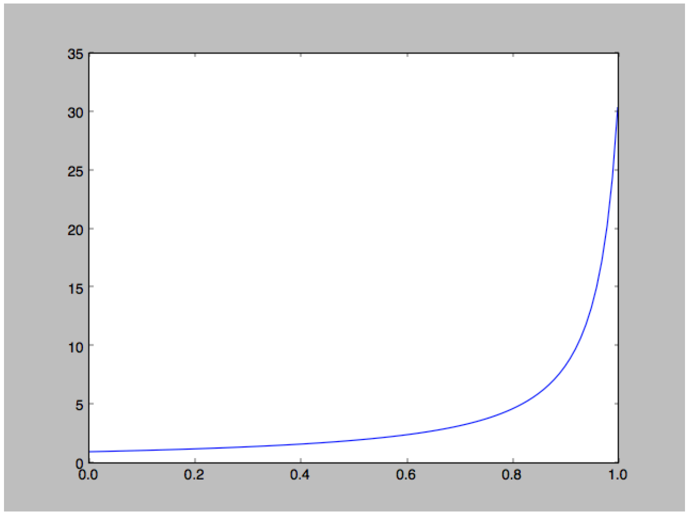
```
import numpy as np
import matplotlib.pyplot as plt

def euler(a,b,Phi,y0,n):
    """Méthode de Cauchy Euler d'intégration approchée de
    y'= Phi(x,f(x)) sur [a,b] avec condition initiale y0 = f(a)
    avec n+1 points"""
    x = np.linspace(a,b,n+1)
    y = np.empty(n+1)          # tableau vide de n+1 elts
    y[0] = y0
    pas = (b-a)/n
    for k in range(n):
        y[k+1] = y[k] + pas * Phi(x[k],y[k])
    return (x,y)
```

13.2.4 Exemple

• **Exemple :** Résolution numérique de $y' = y^2$ avec $y(0) = 1$ sur $[0,1]$.

```
Phi = lambda x,y: y**2
x,y = euler(0,1,Phi,1,100)
plt.plot(x,y) ; plt.show()
```



Résolution numérique sur l'intervalle $[0,1]$ de $y' = y^2$ avec condition initiale

$(0,1)$.

Un calcul simple montre que la solution exacte sur $]-\infty;1[$ est : $x \mapsto \dfrac{1}{1-x}$; en réalité elle n'est pas définie en $x=1$!...

13.3 Résolution sous `scipy`

13.3.1 Intégration d'équations différentielle avec `odeint()`

• La méthode `odeint()` du sous-module `integrate` de `scipy` permet l'intégration d'équations différentielles avec condition initiale :

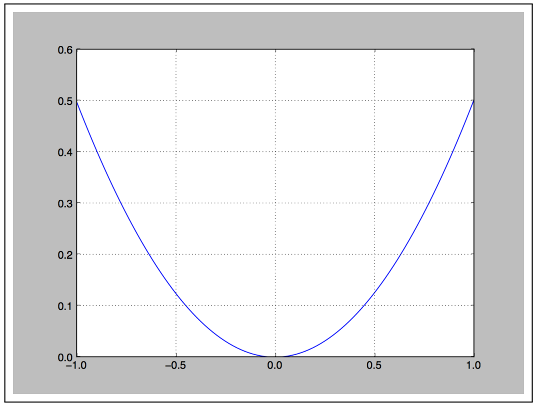
O.D.E. pour 'Ordinary Differential Equation', c.à.d. 'Equation différentielle ordinaire', on dirait plutôt du premier ordre : $y' = \phi(y,t)$ avec condition initiale (t_0, y_0).

• Utilisation : Prenons l'exemple de l'intégration sur $[-1,1]$ de l'E.D. :

$$\boxed{y' = t \quad \text{avec } y(-1) = \frac{1}{2}}$$

dont l'unique solution est $y(t) = \dfrac{t^2}{2}$. (C'est en fait ici une recherche de primitive.)

```
import numpy as np
t = np.linspace(-1,1,100)    # temps de l'échantillonnage
phi = lambda y,t: t    # fonction du second membre
from scipy import integrate
y0 = 1/2    # condition initiale au temps initial t[0]=-1
y = integrate.odeint(phi, y0, t)    # Résolution
import matplotlib.pyplot as plt    # tracé
plt.plot(t,y)
plt.grid()
plt.show()
```



Voici le tracé obtenu. Il correspond bien au résultat attendu : $y(t) = \dfrac{t^2}{2}$ sur l'intervalle $[-1,1]$.

• Autre exemple :

$$\boxed{y' = -y \quad \text{avec } y(0) = 1}$$

a pour unique solution $y(t) = e^{-t}$.

```
phi = lambda y,t: -y    # fonction second membre
t = np.linspace(0,4,100)    # 100 échantillonnages entre 0 et 4
y0 = 1                      # valeur initiale au temps t[0] = 0
y = integrate.odeint(phi,y0,t)    # intégration
plt.plot(t,y)           # tracé
plt.grid()
plt.show()
```