TP Informatique n° 7
Equations différentielles

1 Rappels sur les équations différentielles

On appelle équation différentielle scalaire d'ordre 1 une équation de la forme

$$y'(t) = F(t, y(t))$$

où l'inconnue y est une fonction de classe $\mathcal{C}^1$ définie sur un intervalle de $\mathbb{R}$, et F une fonction de classe $\mathcal{C}^1$. De nombreux phénomènes sont modélisés à l'aide d'équations différentielles, et lorsqu'on impose une valeur y_0 en t_0, un théorème (dit de Cauchy-Lipschitz) assure qu'il existe une unique solution vérifiant l'équation différentielle et la condition initiale voulue (on appelle cela un problème de Cauchy). Malheureusement, il est souvent impossible de trouver une expression pour cette solution, c'est pourquoi on recourt aux méthodes d'approximation numériques.

Voici quelques exemples d'équations différentielles que l'on souhaite résoudre.

■ **Exemple 1** La fonction exponentielle est l'unique solution du problème de Cauchy :

$$\left\{\begin{array}{rcl} y' & = & y \\ y(0) & = & 1 \end{array}\right.$$

■

■ **Exemple 2** Le modèle logistique (ou modèle de Verhulst) :

$$\left\{\begin{array}{rcl} y' & = & ry(1-\frac{y}{K}) \\ y(0) & = & y_0 \end{array}\right.$$

avec r et $K > 0$. C'est un modèle non-linéaire, mais dont on connaît encore l'expression de la solution. ■

■ **Exemple 3** ⚠ L'équation différentielle

$$\left\{\begin{array}{rcl} y' & = & \sin(ty) \\ y(0) & = & y_0 \end{array}\right.$$

n'est pas autonome : contrairement aux précédents exemples, $f(t, y)$ dépend de t ! ■

■ **Exemple 4** L'équation différentielle de l'oscillateur harmonique, utilisé en électricité :

$$\left\{\begin{array}{rcl} y''+\omega^2 y & = & 0 \\ y(0) & = & y_0 \\ y'(0) & = & \dot{y}_0 \end{array}\right.$$

C'est une équation différentielle du second ordre à coefficients constants, que l'on sait résoudre à l'aide des fonctions trigonométriques. ■

■ **Exemple 5** L'équation différentielle du pendule :

$$\begin{cases} \theta'' + \omega^2 \sin(\theta) &= 0 \\ \theta(0) &= \theta_0 \\ \theta'(0) &= \dot{\theta}_0 \end{cases}$$

C'est une équation différentielle non-linéaire du second ordre. Elle est nettement plus difficile à résoudre explicitement. ■

■ **Exemple 6** Le modèle proie-prédateur de Lotka et Volterra :

$$\begin{cases} x'(t) &= rx(t) - px(t)y(t) \\ y'(t) &= -dy(t) + qx(t)y(y) \\ x(0) &= x_0 \\ y(0) &= y_0 \end{cases}$$

■

Commenter ce dernier exemple, et l'écrire sous la forme $Z'(t) = F(t, Z(t))$ avec $Z(t) = \binom{x(t)}{y(t)}$.

2 Méthode d'Euler

On souhaite résoudre numériquement (de façon approchée) une équation différentielle du premier ordre, sur l'intervalle $[0, T]$, en se donnant un *pas* $h = \frac{T}{N}$ (avec $N \in \mathbb{N}^*$), et une subdivision $t_k = k\frac{T}{N}$, ($k \in [\![0, N]\!]$). Connaissant y_0, on cherche à calculer des valeurs approchées y_k de $y(t_k)$, où y est la solution de l'équation différentielle véfifiant $y(0) = y_0$.

En considérant la méthode des rectangles (déjà vue), on remarque que :

$$y(t_{k+1}) - y(t_k) = \int_{t_k}^{t_{k+1}} y'(u)du = \int_{t_k}^{t_{k+1}} f(u, y(u))du \approx hf(t_k, y(t_k))$$

On obtient donc la méthode d'Euler : pour $k \in [\![0, N-1]\!]$,

$$y_{k+1} = y_k + hf(t_k, y_k)$$

■ **Exercice 1** Ecrire deux fonctions `euler` et `eulerbis` implémentant la méthode d'Euler, l'une utilisant une boucle while et l'autre une boucle for. ■

■ **Exercice 2** En l'absence de prédateurs et avec une nourriture abondante, l'évolution de la famille Grolapin suit un modèle de Malthus : $y' = ry$. Représenter graphiquement la solution exacte pour différentes valeurs de r ainsi que les solutions approchées pour dfférentes valeurs de N. ■

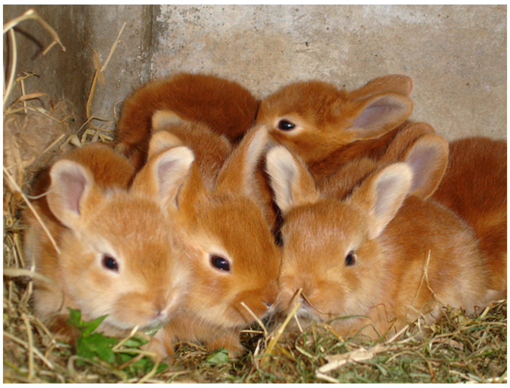

FIGURE 1 – La famille Grolapin au complet

■ **Exercice 3** Mettre en oeuvre la méthode d'Euler pour résoudre l'équation différentielle de l'exemple 3. ■

3 Utilisation de la bibliothèque `Scipy`

Afin de résoudre le système différentiel issu du modèle de Lotka-Volterra, on va utiliser une fonction de la bibliothèque `scipy` plus performante (plus précise avec un nombre similaire d'opérations) pour la résolution d'équations différentielles.

Listing 1 – Modèle proie-prédateur de Lotka-Volterra

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint
```

```python
r = float(input("entrer le coeffcient r"))
p = float(input("entrer le coeffcient p"))
d = float(input("entrer le coeffcient d"))
q = float(input("entrer le coeffcient q"))
proie0 = float(input("nombre de proies initial"))
predat0 = float(input("nombre de predateurs initial"))
T=float(input("temps maximal"))

def F(y,t):
    return(np.array([r*y[0]-p*y[0]*y[1],-d*y[1]+q*y[0]*y[1]]))


y0=np.array([proie0,predat0])
temps=np.linspace(0,T,501)
res=odeint(F,y0,temps)

plt.figure()
plt.grid()
plt.xlabel("Temps")
plt.ylabel("Effectif")
plt.plot(temps,res[:,0],label="Proie")
plt.plot(temps,res[:,1],label="Predateur")
plt.legend(loc="best")
plt.title("Evolution des populations")

plt.figure()
plt.plot(res[:,0],res[:,1])
plt.title("Portrait de phase")
plt.show()
```

Commenter le programme et les sorties graphiques.

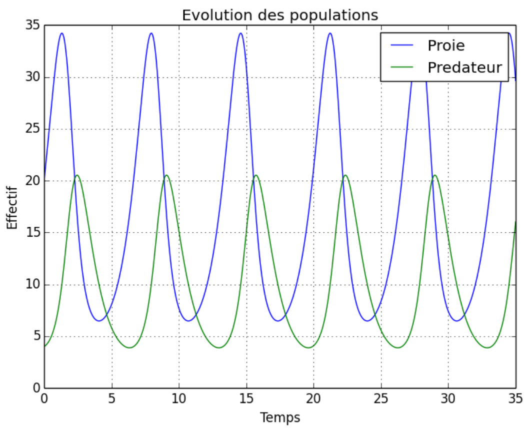

FIGURE 2 – Résolution numérique du modèle de Lotka-Volterra

FIGURE 3 – Un terme de prédation a été ajouté au modèle

4 Equations différentielles linéaires d'ordre 2

Dans cette partie, on se propose de résoudre numériquement l'équation différentielle $y'' + y = 0$ avec les conditions initiales $y(0) = 0$ et $y'(0) = 1$, dont on sait que la solution est la fonction sinus. Pour ela, on va utiliser les opérations sur les tableaux de la bibliothèque `Numpy`.

■ **Exemple 7** Commenter le résultat des commandes suivantes :

```
>>> (1,2) + (4,5)

>>> [1,2] + [4,5]

>>> 2*np.array([0,1]) + np.array([3,0])
```

■

■ **Exercice 4** Adapter la méthode d'Euler pour résoudre cette équation différentielle. Commenter le graphique obtenu. ■

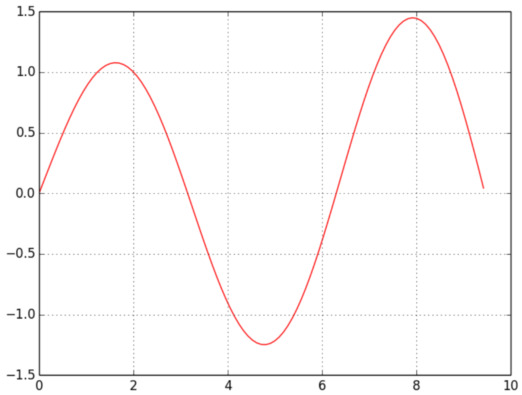

FIGURE 4 – Solution numérique de l'équation différentielle $y'' + y = 0$