Université Lyon 1 Année 2018-2019
Master Mathématiques Générales 1ère année
Logiciels Scientifiques

TP3 – Python

1 Étude d'EDO

Pour tout le TP, on pourra entrer les commandes

```
import numpy as np
import scipy.integrate as spi
import matplotlib.pyplot as plt
```

Exercice 1 - Schémas numériques

On considère l'EDO suivante

$$\begin{cases} y'(t) = ay(t) \\ y(0) = y_0 \end{cases}$$

dont la solution est $y(t) = y_0\, e^{at}$.

On se propose d'étudier les solutions numériques de cette équation. On considère une discrétisation uniforme de l'intervalle en temps $[0, T]$ et on note $t_i = ih$, $i \in [\![1, n]\!]$, avec $h = T/n$. Il s'agit de construire une approximation $\bar{y}_i \approx y(t_i)$ pour tout $i \in [\![1, n]\!]$.

Dans la suite, on considèrera les valeurs numériques $a = -1$, $y_0 = 1$, $T = 1$, $n = 10$.

1. **Méthode explicite à un pas -** L'approximaton $\bar{y}_i$ est définie comme suit :

$$\begin{cases} \bar{y}_0 = y_0 \\ \bar{y}_{i+1} = \bar{y}_i + h\, a\, \bar{y}_i. \end{cases}$$

   Écrire une fonction `EulerEx(a,y_0,h,n)` donnant la liste des $\bar{y}_i$ correspondant à la méthode.

2. **Méthode implicite à un pas -** L'approximaton $\bar{y}_i$ est définie comme suit :

$$\begin{cases} \bar{y}_0 = y_0 \\ \bar{y}_{i+1} = \bar{y}_i + h\, a\, \bar{y}_{i+1}. \end{cases}$$

   Écrire une fonction `EulerIm(a,y_0,h,n)` donnant la liste des $\bar{y}_i$ correspondant à la méthode.

3. **Méthode de Runge-Kutta d'ordre 2 -** L'approximaton $\bar{y}_i$ est définie comme suit :

$$\begin{cases} \bar{y}_0 = y_0 \\ k_1 = a\, \bar{y}_i \\ k_2 = a\,(\bar{y}_i + \dfrac{h}{2}\, k_1) \\ \bar{y}_{i+1} = \bar{y}_i + h\, k_2. \end{cases}$$

   Écrire une fonction `RK2(a,y_0,h,n)` donnant la liste des $\bar{y}_i$ correspondant à la méthode

4. **Méthode de Runge-Kutta d'ordre 4 -** L'approximaton $\bar{y}_i$ est définie comme suit :

$$\begin{cases} \bar{y}_0 = y_0 \\ k_1 = a\, \bar{y}_i \\ k_2 = a\,(\bar{y}_i + \dfrac{h}{2}\, k_1) \\ k_3 = a\,(\bar{y}_i + \dfrac{h}{2}\, k_2) \\ k_4 = a\,(\bar{y}_i + h\, k_3) \\ \bar{y}_{i+1} = \bar{y}_i + \dfrac{h}{6}\,(k1 + 2k_2 + 2k_3 + k_4)\,. \end{cases}$$

   Écrire une fonction `RK4(a,y_0,h,n)` donnant la liste des $\bar{y}_i$ correspondant à la méthode.

5. On considère $a = -1$, $y_0 = 1$, $T = 1$, $n = 10$. Tracer la solution exacte et les solutions approchées obtenues par les quatre méthodes sur une même figure en mettant un titre et une légende.

6. Calculer l'erreur commise pour chacune des méthodes en $t = T$.

Exercice 2 - Système proie-prédateur

Soit $l(t)$ le nombre de lapins et $c(t)$ le nombre de coyotes présents à un temps t donné. On considère les données suivantes :

- pour les lapins : taux de natalité=1/2, taux de mortalité=c/50 correspondant aux lapins dévorés par les coyotes
- pour les coyotes : taux de natalité=l/10000 proportionnel à la nourriture disponible, taux de mortalité=1/10).

Le système d'équations différentielles correspondant à l'évolution des deux populations pour $t > 0$ est alors le suivant :

$$\begin{cases} \dfrac{\mathrm{d}l}{\mathrm{d}t} = l\left(\dfrac{1}{2} - \dfrac{c}{50}\right) \\ \dfrac{\mathrm{d}c}{\mathrm{d}t} = c\left(\dfrac{l}{10000} - \dfrac{1}{10}\right) \end{cases}$$

1. Déterminer les points d'équilibre théoriques du système.
2. Écrire le problème sous la forme $\dfrac{\mathrm{d}x}{\mathrm{d}t} = f(x,t)$.
3. Écrire une fonction `f(x,t)` correspondant au problème.
4. Résoudre numériquement le système d'équations différentielles sur $[0,50]$ pour différentes données initiales.
   ↪ *Utiliser la commande* `spi.odeint`.
5. Tracer l'évolution du système $(l(t),c(t))$ pour les différentes données initiales testées, ainsi que les points d'équilibre du système.
6. Le champ de vecteurs correspondant au système est le suivant :

$$(x,y) \mapsto \left[x\Big(\frac{1}{2} - \frac{y}{50}\Big), y(\frac{x}{10000} - \frac{1}{10}\Big)\right]$$

   Définir deux vecteurs X et Y discrétisant respectivement $[0,2600]$ et $[0,50]$ avec $N = 50$ points, puis construire les tableaux U et V tels que $(U[i,j], V[i,j])$ contienne le champ de vecteur au point $(X[i], Y[j])$.
7. Tracer le portrait de phase du système sur la même figure, en règlant la longueur des flèches. et matérialiser les points d'équilibre par des croix rouges.
   ↪ *Utiliser la commande* `plt.quiver` *avec l'option* `angles='xy'`.
