TP Informatique commune 1A : Équations différentielles

On importera les bibliothèques suivantes:

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

1. Premier exemple

Nous allons illustrer la résolution numérique d'équations différentielles.

On considère le problème de Cauchy : $\begin{cases} y' = -2xy \\ y(0) = 1 \end{cases}$, dont la solution exacte est $\phi(x) = e^{-x^2}$.

Rappelons tout d'abord les deux méthodes que nous utiliserons.

Méthode d'Euler

Étant donnés deux réels a et b et un système de Cauchy de type $\begin{cases} y' = F(y,x) \\ y(0) = y_0 \end{cases}$, on souhaite déterminer une solution approchée du système sur $[a,b]$. On choisit pour cela un entier $n > 0$ et on pose:

- $h = \frac{b-a}{n}$ (le pas)
- Pour $0 \leqslant i \leqslant n$, $x_i = a + ih$
- Pour $0 \leqslant i \leqslant n-1$, $y_{i+1} = y_i + hF(y_i, x_i)$

Méthode `odeint` du module `scipy`

La fonction `F` étant définie, et `X` étant le vecteur $[x_0, \ldots, x_n]$ défini plus haut, on peut récupérer un vecteur `Y` approchant la solution exacte aux points x_i avec la commande `odeint(F,y0,X)` en ayant au préalable importer la fonction `odeint` comme indiqué en début de TP.

Résolution approchée de l'exemple

1. Écrire les fonctions `F` : $(y,x) \mapsto -2xy$ et `phi` : $x \mapsto e^{-x^2}$ (on rappelle que la fonction `exp` se trouve dans le module `numpy`).

   On pose $a = 0$, $b = 5$, $n = 20$.

2. Construire le vecteur `X` $= [x_0, \ldots, x_n]$ (on pourra utiliser la commande `linspace` du module `numpy`).

3. Calculer le vecteur `Y_E` $= [y_0, \ldots, y_n]$ de la méthode d'Euler (on pourra d'abord construire une liste puis la convertir en vecteur).

4. Calculer le vecteur `Y_V` $= [\phi(x_0), \ldots, \phi(x_n)]$ (cela peut être fait en une seule commande !)

5. Calculer le vecteur `Y_SP` obtenu à l'aide de la méthode `odeint`. Remarque : il sera renvoyé sous forme d'un vecteur colonne, on peut le transformer en vecteur ligne avec la commande `Y_SP = Y_SP[:,0]` (plus généralement, `M[:,i]` permet d'extraire la `i`-ème colonne d'une matrice, cela sera utile plus tard).

6. Déterminer $\max(|Y\_E[i] - Y\_V[i]|)$ et $\max(|Y\_SP[i] - Y\_V[i]|)$ (là aussi, une seule commande suffit ! On rappelle que le module `numpy` dispose d'une fonction `max`).

7. Tracer sur un même dessin les courbes représentant `Y_V, Y_E, Y_SP` en fonction de `X`. Commenter.

8. Recommencer en augmentant la valeur de n.

2. Système proie-prédateur de Lotka-Volterra

On s'intéresse maintenant au système différentiel :

$$\begin{cases} x'(t) = x(t)(1 - y(t)) \\ y'(t) = y(t)(x(t) - \frac{1}{2}) \\ x(0) = \frac{1}{2}, y(0) = \frac{1}{2} \end{cases}$$

On vectorialise ce système en posant $X(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}$ (on le représentera en Python par un vecteur ligne). L'équation devient $X'(t) = G(X, t)$ à laquelle on peut appliquer directement les méthodes précédentes. En effet, si `X` est un tableau `numpy`, la formule de la méthode d'Euler a un sens.

1. Ecrire la fonction `G(X,t)` correspondante. Le paramètre `X` ainsi que le résultat seront des tableaux `numpy`.

2. Appliquer la méthode d'Euler à cette équation en prenant $a = 0$, $b = 10$ et $n = 100$. On pourra construire une liste de tableaux `numpy` que l'on convertira ensuite en matrice appelée `X_E`.

3. Extraire de `X_E` ses vecteurs colonnes `x_E` et `y_E`.

4. Faire le même travail avec la fonction `odeint`. Les résultats obtenus seront nommés `X_S`, `x_S` et `y_S`.

5. Tracer sur un même dessin les courbes représentant `y_E` en fonction de `x_E` et `y_S` en fonction de `y_S`. Sachant que l'on peut montrer mathématiquement que les solutions du système de Lotka-Volterra sont périodiques, qu'en déduisez-vous sur la qualité des méthodes d'Euler et `odeint` ?

3. Équation du pendule

On souhaite résoudre l'équation du pendule (d'ordre 2) : $\begin{cases} \theta''(t) = -\sin(\theta(t)) \\ \theta(0) = 0.1, \theta'(0) = 0 \end{cases}$.

En posant $X(t) = \begin{pmatrix} \theta(t) \\ \theta'(t) \end{pmatrix}$, se ramener à une équation vectorielle d'ordre 1 en X puis la résoudre à l'aide des méthodes précédentes (on prendra $a = 0$, $b = 10$, $n = 100$). Tracer les courbes de $\theta(t)$ en fonction de t ainsi que $\theta'(t)$ en fonction de $\theta(t)$ (portrait de phase).