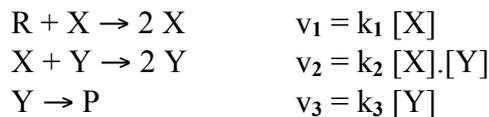


## TD INFORMATIQUE : Modélisation et résolutions numériques d'équations différentielles (2)

But : utiliser des méthodes d'approximations numériques de solutions d'équations différentielles. On utilisera notamment la méthode d'Euler « explicite » (méthode au programme), la méthode d'Euler « implicite » (dans la partie IV) et la fonction *odeint* du sous-module *scipy.integrate*.

### III. Modèle prédateur-proie de Lotka-Volterra

Certaines réactions chimiques comme la réaction de Belousov-Zhabotinskii montrent des oscillations temporelles de concentrations d'espèces chimiques. Les systèmes prédateur-proie (requin-sardine, lynx-lièvre, etc.) présentent également des oscillations temporelles. Le modèle cinétique de Lotka-Volterra est le suivant :



On prendra les valeurs numériques suivantes :

$$k_1 = 1,0 \text{ s}^{-1}; \quad k_2 = 1,0 \text{ L.mol}^{-1}.\text{s}^{-1}; \quad k_3 = 0,5 \text{ s}^{-1}.$$

$$[X]_0 = 1,0 \text{ mol.L}^{-1}; \quad [Y]_0 = 1,5 \text{ mol.L}^{-1}.$$

Dans le cadre de ce modèle, on peut montrer que :

$$\begin{aligned} d[X]/dt &= k_1 [X] - k_2 [X].[Y] \\ d[Y]/dt &= k_2 [X].[Y] - k_3 [Y] \end{aligned}$$

1) Utiliser la méthode d'Euler pour obtenir une approximation numérique du système prédateur-proie de Lotka-Volterra. On pourra écrire une fonction qui, appliquée à  $f$ ,  $g$ ,  $x_0$ ,  $y_0$ ,  $t_0$ ,  $T$  et  $p$  ( $p$  représente, au choix, le nombre de points calculés ou le pas numérique) renvoie les listes des concentrations calculées pour les différents temps  $t_n$  par la méthode d'Euler.

Représenter dans un sous-graphe les approximations des concentrations  $[X]$  et  $[Y]$  en fonction du temps (0 à 30 s). Dans l'autre sous-graphe, représenter le portrait de phase  $[Y]$  en fonction de  $[X]$ . Commenter les résultats.

2) Si le temps le permet, faire de même avec la fonction *odeint*.

### IV. Étude d'un oscillateur harmonique

Un point matériel de masse  $m$  est attaché à un ressort vertical de raideur  $k$  et de longueur  $l_0$  au repos. On note  $Ox$  l'axe vertical ascendant et on pose  $x=l-l_{\text{eq}}$  où  $l_{\text{eq}}$  désigne la position à l'équilibre. En absence de frottements, la relation fondamentale de la dynamique appliquée au point matériel dans le référentiel terrestre supposé galiléen s'écrit :

$$\frac{d^2x}{dt^2} + \omega^2 x = 0 \quad \text{avec} \quad \omega^2 = \frac{k}{m}$$

On prendra :  $k = 20 \text{ N.m}^{-1}$  et  $m = 5 \text{ kg}$

1) Utiliser la méthode d'Euler (« explicite ») pour représenter dans un sous-graphe la position  $x$ , la vitesse  $v$  en fonction du temps. Dans l'autre sous-graphe, représenter le portrait de phase vitesse en fonction de la position.

2) Conclure : la méthode d'Euler explicite est-elle précise ? est-elle stable ?

3) Mêmes questions en utilisant la méthode d'Euler « implicite » (donnée dans l'annexe 1 ci-après).

4) Si le temps le permet : mêmes questions en utilisant la méthode de Runge-Kutta4 « explicite » (donnée dans l'annexe 2 ci-après). Sinon utiliser la fonction *odeint* de Scipy.

### Annexe 1 : Oscillateur harmonique par méthode d'Euler implicite

On note  $x_n, v_n$  et  $a_n$  les approximations numériques de la position  $x$ , de la vitesse  $v$  et de l'accélération  $a$  au temps  $t_n$ , ainsi que  $dt$  le pas pour la résolution numérique du système d'équation différentielle.

– La méthode d'Euler explicite pour l'oscillateur harmonique sans frottement revient à écrire :

$$x_{n+1} = x_n + v_n dt \quad \text{et} \quad v_{n+1} = v_n + a_n dt = v_n - \omega^2 \cdot x_n dt$$

– La méthode d'Euler implicite pour ce système est légèrement différente et consiste à écrire :

$$x_{n+1} = x_n + v_{n+1} dt \quad \text{et} \quad v_{n+1} = v_n + a_{n+1} dt = v_n - \omega^2 \cdot x_{n+1} dt$$

on peut déduire de ce système à deux équations que :

$$x_{n+1} (1 + \omega^2 \cdot dt^2) = x_n + v_n dt \quad \text{et} \quad v_{n+1} (1 + \omega^2 \cdot dt^2) = v_n - \omega^2 \cdot x_n dt$$

**Annexe 2 :** On trouve sur internet l'information suivante concernant la méthode de Runge Kutta d'ordre 4 ([http://fr.wikipedia.org/wiki/Méthodes\\_de\\_Runge-Kutta](http://fr.wikipedia.org/wiki/Méthodes_de_Runge-Kutta)) :

Considérons le problème suivant :

$$y' = f(t, y), \quad y(t_0) = y_0$$

La méthode RK4 est donnée par l'équation :

$$y_{n+1} = y_n + h/6 \cdot (k_1 + 2k_2 + 2k_3 + k_4)$$

$$\begin{aligned} \text{où : } \quad k_1 &= f(t_n, y_n) \\ k_2 &= f(t_n + h/2, y_n + k_1 \cdot h/2) \\ k_3 &= f(t_n + h/2, y_n + k_2 \cdot h/2) \\ k_4 &= f(t_n + h, y_n + k_3 \cdot h) \end{aligned}$$

L'idée est que la valeur suivante ( $y_{n+1}$ ) est approchée par la somme de la valeur actuelle ( $y_n$ ) et du produit de la taille de l'intervalle (pas numérique noté  $h$ ) par la pente estimée. La pente est obtenue par une moyenne pondérée de pentes :

- $k_1$  est la pente au début de l'intervalle ;
- $k_2$  est la pente au milieu de l'intervalle, en utilisant la pente  $k_1$  pour calculer la valeur de  $y$  au point  $t_n + h/2$  par le biais de la méthode d'Euler ;
- $k_3$  est de nouveau la pente au milieu de l'intervalle, mais obtenue cette fois en utilisant la pente  $k_2$  pour calculer  $y$ ;
- $k_4$  est la pente à la fin de l'intervalle, avec la valeur de  $y$  calculée en utilisant  $k_3$ .

Dans la moyenne des quatre pentes, un poids plus grand est donné aux pentes au point milieu :

$$\text{pente} = (k_1 + 2k_2 + 2k_3 + k_4)/6$$

La méthode RK4 est une méthode d'ordre 4, ce qui signifie que l'erreur commise à chaque étape est de l'ordre de  $h^5$ , alors que l'erreur totale accumulée est de l'ordre de  $h^4$ .

Ces formules sont aussi valables pour des fonctions à valeurs vectorielles.