# TP 12 : Résolution d'équations différentielles

On rappelle que la formulation générale d'une équation différentielle d'inconnue $x$ est de la forme :

$$\begin{cases} x'(t) &= f(x(t),t) \\ x(t_0) &= x_0 \end{cases}$$

où la fonction inconnue cherchée est à valeurs dans $\mathbb{R}^k$ (ainsi $f$ est définie sur un sous ensemble de $\mathbb{R}^k \times \mathbb{R}$ et $x_0 \in \mathbb{R}^k$).

## 1 Équations différentielles en dimension 1

Télécharger l'annexe sur le site web. On reprend comme exemple le circuit RC brièvement aperçu dans le TP 3.

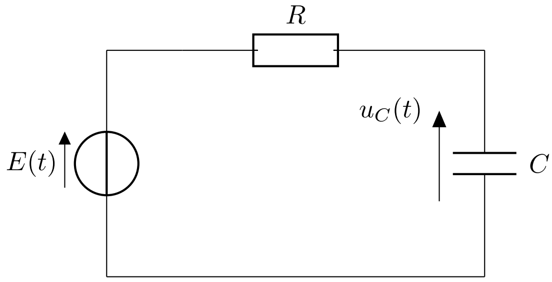


Figure 1: Circuit RC - Charge d'un condensateur

**Question 1.** *Mise sous forme précédente.* On rappelle que l'équation vérifiée par la tension $u$ aux bornes du condensateur satisfait l'équation

$$\mathrm{RC}\frac{\mathrm{d}\,u}{\mathrm{d}t}(t) + u(t) = E(t)$$

où $E$ est la force électromotrice du générateur.

1. Mettre cette équation sous la forme « canonique » $u'(t) = f(u(t), t)$. Pas de code ici, c'est à faire à la main !
2. Dans l'annexe, on a pose $R = 500\Omega$, $C = 10^{-6}$F et on définit plusieurs tensions $E_1$, $E_2$ et $E_3$ (tension constante, tension en créneaux, tension en triangles). Définir informatiquement trois fonctions `fRC1`, `fRC2`, `fRC3` égales à la fonction $f$ dans chacun des cas.

On rappelle la méthode d'Euler explicite : pour $h$ un petit pas de temps, on fait l'approximation $x(t + h) \simeq x(t) + hx'(t)$. Cette méthode peut-être vue comme une application de la méthode des rectangles à gauche pour le calcul d'intégrale, ou encore la troncature du développement de Taylor de $x$ en $t$ au premier ordre. On peut donc calculer itérativement des approximations de $x$ à partir de $x_0$, en suivant le *schéma numérique* :

$$x_0 \text{ donné}; \qquad x_i = x_{i-1} + (t_i - t_{i-1})f(x_{i-1}, t_{i-1}) \text{ pour } i \geq 1$$

**Question 2.** *Réécriture de la méthode d'Euler.* Écrire une fonction `euler_explicite(f,x0,T)` qui prend comme arguments une fonction `f`, une valeur intiale `x0`, et un tableau de temps $(t_i)_{0\leq i<n}$, et renvoyant une liste d'approximations $(x_i)_{0\leq i<n}$ obtenue par la méthode d'Euler explicite à l'aide du schéma précédent. Évidemment $x_i \simeq x(t_i)$ est calculé pour $i \geq 1$ à partir de $x_{i-1}$.

Dans la suite, on aura $t_0 = 0$, et $x_0 = 0$ : on ferme l'interrupteur au temps $t = 0$ et le condensateur est initialement déchargé.

**Question 3.** *Comparaison avec la solution exacte.* Dans le cas de la tension $E_1$ constante, la solution exacte de l'équation différentielle est : $u(t) = E_1(1 - \exp^{-t/\tau})$, avec $\tau = \mathrm{RC}$. Tracer sur un même graphique :

— la solution exacte pour une f.e.m $E_1$ constante, sur l'intervalle $[0, 5\tau]$ (on pourra définir un tableau Numpy de temps relativement proches, par exemple 1000 points dans l'intervalle, via `np.linspace`.

— l'approximation donnée par la méthode d'Euler explicite, pour un pas de temps variable (on pourra utiliser `np.linspace` également, en prenant par exemple 10, 20 ou 100 points).

**Question 4.** *Comparaison avec* ***`odeint`*** *.* Reprendre la question précédente avec `fRC2` et `fRC3`, en remplaçant la solution exacte par la solution fournie par `odeint` (qui s'utilise comme notre fonction `euler_explicite`), sur l'intervalle $[0, 30\tau]$. Vérifier qu'avec 10 ou 20 points, la méthode d'Euler explicite est assez mauvaise. Vérifier tout de même qu'avec un grand nombre de points (1000 par exemple), la solution obtenue avec la méthode d'Euler converge vers la solution obtenue par `odeint` (qu'on peut considérer exacte).

On rappelle maintenant le schéma de la méthode d'Euler *implicite*. L'idée est de procéder par intégration via la méthode des rectangles à droite :

$$x(t+h) = x(t) + \int_t^{t+h} f(x(u), u)\,\mathrm{d}u \simeq x(t+h) + hf(x(t+h), t+h)$$

Pour obtenir $x(t+h)$ à partir de $x(t)$, il faut donc résoudre une équation numérique de la forme $g(y) = 0$. La fonction `sco.fsolve(g,y0)` peut être utilisée ici, pour chercher un zéro de l'équation $g(y) = 0$ au voisinage de `y0`. On en déduit donc le schéma suivant

$$x_0 \text{ donné}\,; \qquad x_i \text{ solution de } y - x_{i-1} - (t_i - t_{i-1})f(y, t_i) \text{ pour } i \geq 1$$

**Question 5.** Écrire une fonction `euler_implicite(f,x0,T)` qui prend comme arguments une fonction `f`, une valeur intiale `x0`, et un tableau de temps $(t_i)_{0\leq i<n}$, et renvoyant une liste d'approximations $(x_i)_{0\leq i<n}$ obtenue par la méthode d'Euler implicite à l'aide du schéma précédent. Pour l'utilisation de `sco.fsolve` dans le calcul de $x_i$, on prendra comme point de départ $x_{i-1}$.

**Question 6.** *Comparaison explicite/implicite*. Sur l'intervalle $[0, 30\tau]$, comparer pour une tension en créneaux (on utilisera `fRC2`) :

— la solution donnée par `odeint` avec 1000 points sur l'intervalle ;

— les solutions données par les méthode d'Euler explicite et implcite, avec peu de points (20 par exemple, voir figure 2).


FIGURE 2: Comparaison des méthodes explicite et implicite (20 points)

*Remarque :* on voit que la méthode implicite est *plus stable* que la méthode explicite !

## 2 Un système différentiel : équations de Lokta-Volterra [1]

On considère un écosystème où vivent des proies (par exemple des sardines), et des prédateurs (par exemple des requins), dépendant du temps. On note $r(t)$ la densité de prédateurs au temps $t$, et $s(t)$ la densité de proies dans

1. Encore un TIPE cadeau !

le milieu. On considère le système différentiel suivant, qui modélise l'évolution de ces densités, où interviennent 4 constantes $A, B, C$ et $D$ toutes strictement positives.

$$\left\{\begin{array}{lll} r'(t) & = & Ar(t)s(t) - Br(t) \\ s'(t) & = & -Cr(t)s(t) + Ds(t) \end{array}\right.$$

L'idée derière ces équations est naturelle : la population des requins augmente en proportion du nombre de sardines, et décroît en l'absence de proies. Inversement, les sardines se multiplient en l'absence de prédateurs, mais diminuent en proportion du nombre de requins !

On se donne également deux réels $r_0$ et $s_0$ donnant les densités (qui sont donc des réels positifs) au temps $t = 0$. On admettra par la suite que le système admet une unique solution $t \mapsto (r(t), s(t))$ vérifiant $x(0) = r_0$ et $y(0) = s_0$ définie sur $\mathbb{R}_+$.

Puisque le système est non-linéaire, on va le résoudre de manière approchée par une méthode numérique.

## 2.1 Résolution avec les méthodes d'Euler

**Question 7.** *Reformulation du système.* On pose dans la suite $x(t)$ le vecteur $(r(t), s(t))$. Donner la fonction $f$ telle que $x'(t) = f(x(t), t)$ (remarque : la fonction $f$ ne dépend pas de son deuxième argument ici !). Définir informatiquement une fonction `fLokta(x,t)` prenant en paramètre un tableau Numpy de taille 2 et un réel (non utilisé) et renvoyant un tableau Numpy de taille 2 représentant $x'(t)$. Pour créer un tableau Numpy de taille 2 de composantes $a$ et $b$, utiliser simplement `np.array([a,b])`. Les constantes $A, B, C$ et $D$ sont données dans l'annexe.

Normalement, les fonctions `euler_explicite` et `euler_implicite` précédemment écrites fonctionnent également dans ce cadre.

**Question 8.** *Graphique.* Tracer le diagramme $r$ en abscisse, $s$ en ordonnée sur l'intervalle de temps $[0, 30]$, pour les trois méthodes de résolution :

— euler explicite ;

— euler implicite ;

— odeint.

On prendra comme condition initiale $(r_0, s_0) = (0.8, 0.7)$, et pour la résolution environ 5000 points dans l'intervalle, pour débuter (voir figure 3).

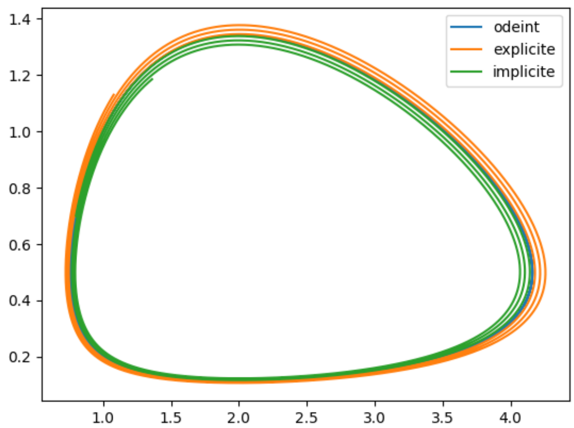


FIGURE 3: Résolution de l'équation de Lokta-Volterra (5000 points sur $[0, 30]$)

**Question 9.** Il est facile de voir que si $(r(t), s(t))$ est une solution de l'équation, alors la fonction $t \mapsto F(r(t), s(t))$ est constante, avec $F(r, s) = Cr + As - B\ln(s) - D\ln(r)$ (on dit que $F$ est une intégrale première du mouvement). Ainsi, les points $(r(t), s(t))$ forment une courbe fermée. Faire varier le nombre de points / l'intervalle de temps, et vérifier que la méthode d'Euler explicite a toujours tendance à « exploser ».

## 2.2 D'autres schémas numériques

On résout ici le système précédent en introduisant deux autres schémas numériques évoqués en cours.

**Méthode Runge-Kutta 2.** Cette méthode repose sur la méthode d'intégration du point milieu :

$$\int_{t_i}^{t_{i+1}} f(x(u),u)\,\mathrm{d}u \simeq (t_{i+1}-t_i)f(x(t_{i+1/2}),t_{i+1/2}) \qquad \text{avec} \quad t_{i+1/2} = \frac{t_i+t_{i+1}}{2}$$

Bien entendu, on ne connaît pas d'approximation de $x(t_{i+1/2})$, on va donc l'approcher pour pouvoir utiliser l'approximation précédente. Pour ce faire, on utilise la même technique que dans la méthode d'Euler explicite : $x(t_{i+1/2}) \simeq x(t_i) + \frac{t_{i+1}-t_i}{2} f(x_i,t_i)$. Ainsi, le schéma obtenu est le suivant :

$$x_{i+1} = x_i + hf\left(x_i + \frac{h}{2}f(x_i,t_i), t_i + \frac{h}{2}\right) \qquad \text{avec} \quad h = t_{i+1}-t_i$$

**Question 10.** Implémenter la méthode Runge-Kutta 2. Reprendre la question 8, en rajoutant la méthode RK2. Vérifier que la solution obtenue avec RK2 est beaucoup plus précise (voir figure 4).

**Méthode Runge-Kutta 4.** Cette méthode est encore plus précise que la méthode RK2, et repose sur la méthode de Simpson pour l'approximation d'intégrales. Voici le schéma :

$$\int_{t_i}^{t_{i+1}} f(x(u),u)\,\mathrm{d}u \simeq \frac{t_{i+1}-t_i}{6}\left(f(x(t_i),t_i) + 4f(x(t_{i+1/2}),t_{i+1/2}) + f(x(t_{i+1}),t_{i+1})\right)$$

Il faut donc estimer les 3 termes de la somme, voici comment on procède :

- le premier terme $f(x(t_i),t_i)$ s'approche simplement avec $k_1 = f(x_i,t_i)$ ;
- on peut faire une première estimation de $x(t_{i+1/2})$ et donc de $f(x_{i+1/2},t_{i+1/2})$ à l'aide de l'approximation des rectangles à gauche (la même que dans la méthode RK2) : on note $k_2 = f(x_i + \frac{h}{2}k_1, t_{i+1/2})$ ;
- à l'aide de l'approximation $x(t_{i+1/2}) \simeq x(t_i) + \frac{h}{2}k_2$, on en déduit une autre approximation de $f(x_{i+1/2},t_{i+1/2})$, qu'on note $k_3 = f(x_i + \frac{h}{2}k_2, t_{i+1/2})$ :
- L'approximation $x(t_{i+1}) \simeq x(t_i) + hk_3$ nous permet enfin d'obtenir une approximation de $f(x(t_{i+1}),t_{i+1})$, qu'on note $k_4 = f(x_i + hk_3, t_{i+1})$.

On peut donc calculer une estimation de l'intégrale, et donc en déduire un schéma permettant d'obtenir une estimation de $x(t_{i+1})$ à partir de $x(t_i)$. Comme on a deux estimations ($k_2$ et $k_3$) pour le terme central $f(x(t_{i+1/2}),t_{i+1/2})$, on prend la moyenne des deux. En résumé, on approche l'intégrale ci-dessus par $\frac{t_{i+1}-t_i}{6}(k_1+2k_2+2k_3+k_4)$.

**Question 11.** Implémenter la méthode Runge-Kutta 4. Tracer sur un même graphique les solutions données par `odeint`, `RK2` et `RK4`. Vérifier que la solution obtenue avec RK4 est très proche de celle calculée par `odeint`.

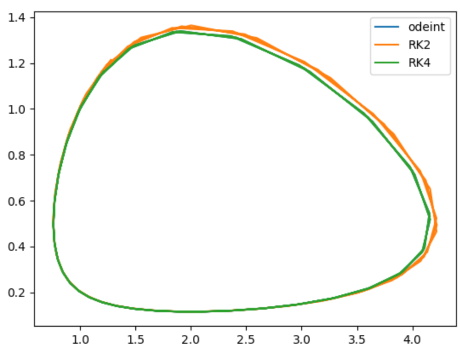


FIGURE 4: Résolution de l'équation de Lokta-Volterra avec RK2 et RK4. On a pris seulement 100 points dans l'intervalle $[0,30]$.