TP3
Equations différentielles

3 janvier 2018

MILHA El Mehdi & ERDEMIR Eren

Dans ce TP, on présente quelques méthodes pour intégrer numériquement sur un intervalle $t \in [0, T]$, une équation différentielle ordinaire - dite EDO - c'est-à-dire de la forme

$$u'(t) = f(t, u(t)), \quad u(0) = u_0 \tag{1}$$

La fonction f, qui prend en argument un couple de réels et qui renvoie un réel, ainsi que la valeur initiale u_0 sont donnés ; la fonction u, qui prend un réel en argument et qui renvoie un réel, est inconnue ; c'est la solution cherchée.

1. Exemple : considérons l'EDO très simple

$$u'(t) = -u(t), \quad u(0) = 1.0$$

   Le problème est ici de déterminer la solution u.
   (a) La fonction f vaut $f(t, u(t)) = u'(t) = -u(t))$.
   (b) on a calculé à la main la solution u de l'EDO ci-dessus,c'est u(t)=-u'(t)=exp(-x) et on l'a dessiné sur papier ; sur un intervalle $t \in [0, 2]$ (voir feuille).
   (c) Pour représenter graphiquement u , on a utilisé le module `matplotlib` ; on l'a défini dans notre fichier tp3 en utilisant la fonction np.exp(-x) ,malgré quelques difficultés qu'on a rencontré au départ avec le module math,on a réussi grâce à l'aide du professeur. On a sauvegardé la figure au format `png` sans problème.
2. Calcul approché de u au moyen de la **méthode d'Euler**.

(a) On a calculé la suite u_k, définie par la méthode d'Euler, avec $T = 2.0$ et $n = 10$. On a donc réalisé une subdivision sur [0 ;2] en 10 parties égales,on a posé $h = T/n$,d'où l'on obtient une subdivision $t_k = 2 * k$. On définit ensuite une suite récursive $u_k + 1 = u_k + h * f$ que l'on a enregistré dans une liste. On obtient ainsi par récurrence une suite d'approximation $u(k + 1)$ de la valeur exacte $u_{tk} + 1$.

(b) On a représenté sur le même graphique la solution exacte u et les points (t_k, u_k) en rouge. Remarque :les approximations sont assez précises.

On a écrit une fonction python `euler` qui prend en arguments une foncion f de deux variables scalaires t, u, une valeur initiale u_0, un réel T, un entier n, et qui renvoie deux `numpy arrays`, tt contenant les valeurs t_k et uu contenant les valeurs u_k, obtenues par la méthode d'Euler.

(a) On a appliqué la fonction `euler` à l'exemple précédent (on aura besoin d'écrire la fonction f de l'EDO - fonction de deux variables - dans une fonction python) et retrouvé les résultats précédents.

(b) On a examiné les EDOs suivantes dont certaines équations peuvent être facilement intégrées à la main, d'autres non.

$$
\begin{aligned}
u'(t) &= -u(t) + t, & u(0) &= 1.0\\
u'(t) &= u^2(t), & u(0) &= 1.0\\
u'(t) &= u^2(t) - t, & u(0) &= 1.0
\end{aligned}
$$

Nous montrons maintenant comment intégrer une EDO d'ordre supérieur. Prenons l'exemple de l'oscillateur harmonique (modélisation du ressort) qui est une EDO d'ordre 2

$$u''(t) + \omega^2 u(t) = 0, \quad u(0) = u_0, \quad u'(0) = v_0 \tag{2}$$

où ω, u_0, v_0 sont des scalaires donnés ; ω s'appelle la pulsation, u_0 la position initiale, v_0 la vitesse initiale. Posons ensuite $v = u'$ et écrivons l'EDO sous la forme

$$\begin{cases} u'(t) = v(t) \\ v'(t) = -\omega^2 u(t) \end{cases} \tag{3}$$

On pose $U = [u, v]$ et $U_0 = [u_0, v_0]$ de sorte que l'EDO s'écrit maintenant

$$U'(t) = F(t, U(t)), \quad U(0) = U_0 \tag{4}$$

avec $F(t, [u, v]) = [v, -\omega^2 u]$; F est donc une fonction de $\mathbb{R} \times \mathbb{R}^2$ dans $\mathbb{R}^2$.

On a ainsi réécrit l'EDO d'ordre 2 en l'EDO) d'ordre 1. Attention ! dans le premier cas, la fonction f va de $\mathbb{R} \times \mathbb{R}$ dans $\mathbb{R}$; dans le deuxième cas, la fonction F va de $\mathbb{R} \times \mathbb{R}^2$ dans $\mathbb{R}^2$. Il est maintenant facile de résoudre cette EDO numériquement, il suffit de lui appliquer la méthode d'Euler sous la forme $U_{k+1} = U_k + hf(t_k, U_k)$, avec U_0 qui est donnée. C'est une récurrence qui fournit la suite des points U_k de $\mathbb{R}^2$, approximations des valeurs $U(t_k)$, où U est la solution exacte de l'EDO.

(a) On a calculé, à la main, la solution exacte de l'EDO .

(b) On a modifié la fonction `euler`, pour qu'elle prenne en arguments F, une foncion de deux variables $t \in \mathbb{R}, U \in \mathbb{R}^2$, $U_0 \in \mathbb{R}^2$, un réel T, un entier n, et qui renvoie deux `numpy arrays`, tt contenant les valeurs t_k et UU contenant les valeurs U_k, obtenues par la méthode d'Euler.

(c) On a résolu numériquement l'EDO , avec $\omega = 1.0, T = 4\pi$ et les conditions initiales $u(0) = 1.0, \quad u'(0) = 0.0$; on écrira la fonction F dans une fonction python ; on prendra différentes valeurs de n.

(d) Faire trois graphiques :

position u en fonction du temps t vitesse v en fonction du temps t vitesse v en fonction de position u

Sur chaque graphique, représenter la solution exacte et la solution approchée donnée par la méthode d'Euler.