M66, Modélisation et analyse numérique

TP4 : Équations Différentielles Ordinaires

Vous êtes invités à télécharger le fichier `tp4_fonction.sci` à partir de moodle, puis de le compléter au fur et à mesure avec les nouvelles procédures. Pour chaque exercices il va falloir créer un fichier, `tp4_exo1.sce`, ..., `tp4_exo4.sce`, comportant les lignes de code correspondant à la résolution de l'exercice et incluant au début le fichier `tp4_fonction.sci` et les autres initialisations habituelles (`clear`,`clc`,...).

Rappel méthodes à un pas

L'objectif des méthodes numériques pour les EDO est de calculer une valeur approchée de la solution du problème de Cauchy :

$$\begin{cases} u'(t) = f(t, u(t)), \quad t > 0 \\ u(0) = u_0 \end{cases} \tag{$\star$}$$

avec $f : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^d$, $d \geq 1$.

Dans ce TP, pour simplifier, on se limitera au cas autonome,
f ne dépendra pas de t, et aux méthodes à pas constant.

Afin de calculer une solution approchée de $(\star)$ sur un intervalle $[0, T]$, on se donne une subdivision régulière de $[0, T]$: $(t_n)_{0 \leq n \leq N}$ avec $t_n = nh$ où $h = T/N$ est le pas de la subdivision. On veut calculer une suite $(u_n)_{0 \leq n \leq N}$ telle que u_n soit une « bonne » approximation de $u(t_n)$ (u solution exacte).

Les méthodes utilisées dans ce TP sont :

Méthode d'Euler explicite : (à programmer)
$u_{n+1} = u_n + hf(u_n)$, avec u_0 pour condition initiale.

Méthode d'Euler implicite : (disponible dans `tp4_fonction.sci`)
$u_{n+1} = u_n + hf(u_{n+1})$, avec u_0 pour condition initiale.

Méthode de Crank-Nicolson : (disponible dans `tp4_fonction.sci`)

$$u_{n+1} = u_n + \frac{h}{2}(f(u_n) + f(u_{n+1})), \text{ avec } u_0 \text{ pour condition initiale.}$$

Méthodes de Runge-Kutta (RK2). : (à programmer)

Euler amélioré (avec $\alpha = \frac{1}{2}$) et Heun (avec $\alpha = 1$) :

$$\begin{cases} k_{n,1} = f(u_n) \\ k_{n,2} = f(u_n + \alpha h k_{n,1}) \\ u_{n+1} = u_n + h\Big(\big(1 - \frac{1}{2\alpha}\big)k_{n,1} + \frac{1}{2\alpha}k_{n,2}\Big), \end{cases}$$

avec u_0 pour condition initiale.

Méthode de Runge-Kutta (RK4) : (disponible dans `tp4_fonction.sci`)

$$\begin{cases} k_{n,1} = f(u_n) \\ k_{n,2} = f(u_n + \frac{h}{2}k_{n,1}) \\ k_{n,3} = f(u_n + \frac{h}{2}k_{n,2}) \\ k_{n,4} = f(u_n + hk_{n,3}) \\ u_{n+1} = u_n + \frac{h}{6}\Big(k_{n,1} + 2k_{n,2} + 2k_{n,3} + k_{n,4}\Big), \end{cases}$$

avec u_0 pour condition initiale.

La méthode d'Euler explicite

Exercice 1 (La méthode d'Euler explicite)

a) Programmer la méthode d'Euler explicite `[U]=EulerExplicite(f,T,N,U0)` qui étant donnés une fonction `f`, la borne supérieur `T` de l'intervalle $[0, T]$, le nombre de subdivisions `N` de cette intervalle et la condition initiale `U0` (qui est un vecteur colonne), retourne une matrice `U` dont les $N + 1$ colonnes représente les valeurs approchées (en particulier `U[1]` $=$ `U0`).

Cette méthode est à mettre dans `tp4_fonction.sci` et sera utilisé à plusieurs reprises dans ce TP.

b) On considère le problème de Cauchy suivant :

$$\begin{cases} y'(t) = -10y(t), \quad t \in [0, 1], \\ y(0) = 2. \end{cases}$$

Définissez une fonction `solex` qui est la solution exacte de ce problème.

c) Tracer sur une même figure la solution exacte et les solutions approchées obtenues par la méthode d'Euler explicite pour $N = 2^k$ avec $k = 4, \dots, 10$.

d) Comme la solution exacte y du problème est $\mathcal{C}^2$, on rappelle que la méthode d'Euler explicite vérifie :

$$\max_{0 \le n \le N} |y_n - y(t_n)| \le Ch \sup_{0 \le t \le T} |y^{(2)}(t)|.$$

Dans une autre figure, tracer la norme infinie de l'erreur commise en échelle « log-log » en fonction du pas $h = T/N$, pour les même $N = 2^k$ avec $k = 4, \dots, 10$, ainsi qu'une droite de pente 1. Que constatez-vous ?

e) On considère les deux problèmes de Cauchy suivants :

$$\begin{cases} y'(t) = 3y(t), \quad t \in [0, 20], \\ y(0) = 1/3. \end{cases} \quad , \quad \begin{cases} y'(t) = -3y(t), \quad t \in [0, 20], \\ y(0) = 1/3. \end{cases}$$

Quelle sont les solutions exactes de ces deux problèmes ? Mettre en évidence pour chacune d'entre elles la sensibilité de la solution numérique par rapport à une petite perturbation aléatoire dans la condition initiale. Pour cela effectuer une résolution par la méthode d'Euler explicite avec $y(0) = 1/3 + \varepsilon$, où $\varepsilon \le 10^{-7}$ est une perturbation aléatoire. Représenter les résultats et commenter les.

Comparaison de quelques schémas numériques

Exercice 2 (Ordres de convergence)

On se propose dans cette exercice de comparer les ordres de convergence des différentes méthodes décrites dans l'introduction de cette feuille.

On rappelle que, en supposant que les solutions y de l'équation différentielles soient de classe $\mathcal{C}^{p+1}$, une méthode numérique est d'ordre de convergence p si :

$$\max_{0 \le n \le N} |y_n - y(t_n)| \le Ch^p \sup_{0 \le t \le T} |y^{(p+1)}(t)|.$$

Trois des méthodes dont on aura besoin, Euler implicite, Crank-Nicolson et Runge-Kutta 4, sont fournies dans le fichier `tp4_fonction.sci`. Les deux méthodes implicites exige comme paramètre, en plus de la fonction `f`, sa dérivée `Df` pour pouvoir résoudre l'équation de récurrence grâce à la méthode de Newton.

a) En plus de la méthode d'Euler explicite déjà implémentée, programmer les méthodes d'Euler amélioré `[U]=EulerAmeliore(f,T,N,U0)` et de Heun `[U]=Heun(f,T,N,U0)`.

b) On considère l'équation différentielle linéaire :

$$\begin{cases} y'(t) = -y(t), \; t \in [0, 5], \\ y(0) = 2. \end{cases}$$

Comme dans 1.d), on se propose de résoudre, puis évaluer les erreurs commises en norme infinie pour tous les méthodes, les trois que vous avez programmées et les trois fournies. Pour cela, faire les calculs pour $N = 2^k$ avec $k = 4, \dots, 10$. Puis mettre en évidence les ordres de convergence en traçant, sur un même graphique, la norme infinie de l'erreur commise en échelle « log-log » pour chacune de ces méthodes, ainsi que 3 droites respectivement de pente 1, 2 et 4.
Que constatez-vous ?

Systèmes différentiels à coefficients constants

Exercice 3 (Un problème raide)

On considère le système :

$$\begin{cases} y_1'(t) = y_2(t), \\ y_2'(t) = -\lambda_1\,\lambda_2 y_1(t) + (\lambda_1 + \lambda_2) y_2(t), \end{cases}$$

avec λ_1 et λ_2 deux réels négatifs distincts tels que $|\lambda_1| >> |\lambda_2|$. On cherche à calculer la solution sur $[0, 20]$.

On fixe pour cet exercice $\lambda_1 = -100$, $\lambda_2 = -1$ et la condition initiale $y_1(0) = y_2(0) = 1$.

a) En utilisant la fonction `expm` donner la solution exacte du problème `y=solex(t)`.

b) Résoudre le problème par les méthodes d'Euler explicite, d'Euler implicite et de Runge-Kuta pour les nombres de subdivisions $N = 966$, $N = 1030$ et $N = 2000$. Pour chaque N tracer sur un graphique la norme de Y_n et sur un autre graphique la trajectoire de Y_n. Que constatez-vous ?

Indication : Pour les méthodes itératives sur $\mathbb{R}^n$, la « dérivée » `Df` *est la matrice jacobienne de* `f`*.*

Exercice 4 Modèle de Lotka-Volterra

On veut appliquer les méthodes d'Euler explicite, d'Euler implicite, d'Euler amélioré, de Heun, de Crank-Nicolson et de Runge-Kutta 4 à la résolution du modèle de Lotka-Volterra :

$$\begin{cases} x' = ax - bxy \\ y' = -cy + dxy \end{cases}$$

Pour cet exercices on fixe les constantes et la condition initiale comme suit :

$$a = 3,\ \ b = 1,\ \ c = 2,\ \ d = 1,\ \ x(0) = 1,\ \ y(0) = 2.$$

a) Calculer la solution sur l'intervalle $[0, T]$, `T=10`, avec `N = 200` subdivisions. Représenter la solution numérique de chaque méthode dans le plan de phase (x, y) (marquer le point initiale). Commenter les résultats.

b) On veut comparer les méthodes d'Euler amélioré, de Heun, de Crank-Nicolson et de Runge-Kutta 4 sur un intervalle plus longue $[0, T]$, `T=100`. Pour faire cela, traçer sur un même graphique l'évolution en temps de l'intégrale première

$$H(x, y) = dx - c \ln x + by - a \ln y,$$

avec le même pas de discrétisation constant qua dans la question précédente . Discuter la stabilité de ces méthodes quand le temps T devient grand.

Rappel : L'intégrale premier $H(x, y)$ est constant le long des solutions du système.