ANALYSE NUMÉRIQUE - TPs

Fiche de Mathématiques 5 - Schémas d'Euler et de Runge-Kutta.

1 Rappels sur les équations différentielles

1.1 Définition, existence et unicité d'une solution

Une équation différentielle est une équation reliant une fonction inconnue et une ou plusieurs de ses dérivées. Une équation est différentielle ordinaire (EDO) si la fonction inconnue ne dépend que d'une variable ; sinon on parle d'équation aux dérivées partielles (EDP). Ici, on va s'intéresser aux équations différentielles ordinaires, qu'on peut présenter rapidement à travers les deux exemples suivants :

Exemple 1.1 Trouver la fonction u définie sur $[0,T]$, où T est le temps final, à valeurs dans $\mathbb{R}$ telle que

$$(P_1)\left\{\begin{array}{l} u'(t)=au(t)\\ u(0)=u0\end{array}\right. .$$

On remarque que la première équation de (P_1) est une équation qui relie la fonction inconnue u à sa dérivée u', c'est bien une EDO. La deuxième équation de (P_1) est quant à elle une condition initiale ; elle fixe la valeur de la fonction u au temps 0. La première équation de (P_1) est une équation de croissance si $a>0$ et d'extinction si $a<0$. Le problème (P_1) a pour solution

$$u(t)=u_0\exp(at),$$

ce qu'on vérifie facilement.

Exemple 1.2 *Le problème du pendule.*
Trouver la fonction φ définie sur $[0,T]$ à valeurs dans $\mathbb{R}$ telle que

$$(P_2)\left\{\begin{array}{l} \varphi''(t)+\dfrac{g}{L}\sin\varphi(t)=0\\ \varphi(0)=\varphi_0\\ \varphi'(0)=\psi_0\end{array}\right.$$

où $g>0$ est l'accélération de la pesanteur et L la longueur du pendule. On notera que la première équation de (P_2) relie la fonction inconnue φ à sa dérivée φ''. La fonction φ représente l'angle du pendule ; les deux dernières équations de (P_2) fixent l'angle initial et la vitesse relative initiale. Ce problème du second ordre peut être écrit comme un système différentiel autonome du premier ordre sur $\mathbb{R}^2$: trouver la fonction u définie sur $[0,T]$ à valeurs dans $\mathbb{R}^2$ telle que

$$u'(t)=f(t,u(t)),$$

en introduisant la fonction de $[0,T]\times\mathbb{R}^2$ dans $\mathbb{R}^2$:

$$f(t,x)=\begin{pmatrix} x_2\\ -\dfrac{g}{L}\sin(x_1)\end{pmatrix},\text{ où } x=\begin{pmatrix}x_1\\ x_2\end{pmatrix}.$$

De manière générale, on s'intéressera ici aux équations différentielles avec une condition initiale, encore appelées

Définition 1.1 *Problème de Cauchy.*
Pour une fonction $f:[0,T]\times\mathbb{R}^N\mapsto\mathbb{R}^N$, trouver u définie sur $[0,T]$ à valeurs dans $\mathbb{R}^N$ telle que

$$(P)\left\{\begin{array}{l} u'(t)=f(t,u(t))\text{ pour } t\in[0,T]\\ u(0)=u_0\in\mathbb{R}^N\end{array}\right. .$$

On se restreint au cas où la fonction $f(t,u)$ est lipschitzienne par rapport à la variable u, c'est-à-dire vérifiant la définition suivante :

Définition 1.2 *Continuité lipschitzienne uniforme.*
Soit $f(t,x)$ une fonction définie continue de $[0,T]\times\mathbb{R}^N$ dans $\mathbb{R}^N$. On dit que f est lipschitzienne par rapport à x uniformément en t s'il existe une constante L telle que pour tout $x, x^\star \in \mathbb{R}^N$ et tout $t\in[0,T]$:

$$|f(t,x)-f(t,x^\star)| \le L|x-x^\star|.$$

La constante L est appelée la constante de Lipschitz.

Dans ce cas on a un résultat d'existence et d'unicité pour la solution du problème de Cauchy (P) :

Théorème 1.1 *Théorème de Cauchy-Lipschitz.*
Soit $T>0$ donné, u_0 dans $\mathbb{R}^N$ et $f(t,x)$ une fonction continue de $[0,T]\times\mathbb{R}^N$ dans $\mathbb{R}^N$, lipschitzienne par rapport à x uniformément en t. Alors il existe une unique fonction $u(t)$ de $[0,T]$ dans $\mathbb{R}^N$, solution du problème différentiel (P).

1.2 Résolution numérique d'une équation différentielle

On étudie dans ce TP les méthodes d'approximation numérique du problème (P) explicites à un pas. Soit n le nombre des points d'approximation. Posons $h=T/n$ le pas de maillage, et $t_i = ih$, $i=0,1,\dots,n$, les points du maillage. Les méthodes considérées ici consistent à trouver les approximations y_i des valeurs $u(t_i)$ de la solution exacte u aux points du maillage t_i, $i=0,1,\dots,n$, telles que

$$\begin{cases} y_0 = u_0 \\ \\ y_{i+1} = y_i + h\phi(t_i,y_i,h) \text{ pour } i=0,1,\dots,n-1 \end{cases}$$

où $\phi(t,y,h)$ est une fonction continue des trois variables sur $[0,T]\times\mathbb{R}^N\times[0,h^\star]$.

Définition 1.3 *On dit qu'une méthode est consistante si*

$$\lim_{h\to 0} \sum_{i=0,\dots,n-1} |u(t_{i+1})-u(t_i)-h\phi(t_i,u(t_i),h)| = 0.$$

Définition 1.4 *On définit l'erreur de consistance*

$$E(t_i,y,h) = \frac{\varphi(t_{i+1})-y}{h} - \phi(t_i,y,h),$$

où $\varphi(t)$ est la solution de $\varphi'(t)=f(t,\varphi)$ avec la condition initiale $\varphi(t_i)=y$. On dit que la méthode est d'ordre p si, pour tout $i=0,\dots,n-1$,

$$|E(t_i,y,h)| \le Kh^p,$$

avec K indépendant de t_i et y pour y dans un borné.

Définition 1.5 *On dira que la méthode est convergente si*

$$\lim_{h\to 0}\max_{i=0,\dots,n} |u(t_i)-y_i| = 0.$$

On définit la stabilité à partir du schéma perturbé :

$$\begin{aligned} y_{i+1} &= y_i + h\phi(t_i,y_i,h) \text{ avec } y_0 \text{ donné,} \\ z_{i+1} &= z_i + h\phi(t_i,z_i,h) + \varepsilon_i \text{ avec } z_0 \text{ donné.} \end{aligned}$$

Définition 1.6 *Le schéma est dit stable s'il existe une constante M ne dépendant que de ϕ telle que*

$$\max_{i=0,\dots,n} |y_i - z_i| \le M\left(|y_0-z_0| + \sum_{i=0}^{n}|\varepsilon_i|\right).$$

On a deux résultats fondamentaux :

Une méthode à un pas consistante et stable est convergente. Si $f\in\mathcal{C}^p([0,T]\times\mathbb{R}^N)$ et que le schéma est d'ordre p, alors l'erreur globale de discrétisation est en $\bigcirc(h^p)$.

Si $\phi(t,y,h) : [0,T]\times\mathbb{R}^N\times[0,h^\star] \mapsto \mathbb{R}^N$ est lipschitzienne en y uniformément par rapport à t et h, alors le schéma est stable.

1.3 Exemples de schémas explicites

- Euler explicite : $\begin{cases} y_0 = u_0 \\ y_{i+1} = y_i + hf(t_i, y_i) \text{ pour } i = 0, 1, \ldots, n-1 \end{cases}$

- Runge-Kutta, deuxième ordre : $\begin{cases} y_0 = u_0 \\ k_1 = f(t_i, y_i) \\ k_2 = f(t_i + h/2, y_i + hk_1/2) \\ y_{i+1} = y_i + hk_2 \text{ pour } i = 0, 1, \ldots, n-1 \end{cases}$

- Runge-Kutta, troisième ordre : $\begin{cases} y_0 = u_0 \\ k_1 = f(t_i, y_i) \\ k_2 = f(t_i + h/2, y_i + hk_1/2) \\ k_3 = f(t_i + h, y_i + hk_2) \\ k_4 = f(t_i + h, y_i + hk_3) \\ y_{i+1} = y_i + h(k_1 + 4k_2 + k_4)/6 \text{ pour } i = 0, 1, \ldots, n-1 \end{cases}$

- Runge-Kutta, quatrième ordre : $\begin{cases} y_0 = u_0 \\ k_1 = f(t_i, y_i) \\ k_2 = f(t_i + h/2, y_i + hk_1/2) \\ k_3 = f(t_i + h/2, y_i + hk_2/2) \\ k_4 = f(t_i + h, y_i + hk_3) \\ y_{i+1} = y_i + h(k_1 + 2k_2 + 2k_3 + k_4)/6 \text{ pour } i = 0, 1, \ldots, n-1 \end{cases}$

1.4 Utilisation du solveur de SCILAB

SCILAB propose une panoplie de schémas numériques déjà tout programmés rassemblés dans la fontion `ode`. Pour résoudre un système différentiel du premier ordre écrit sous la forme classique

$$Y'(t) = F(t, Y(t)) \tag{1}$$

$$Y(0) = Y0 \tag{2}$$

on programme la fonction $F(t, y)$ dans une fonction SCILAB de syntaxe d'appel

```
function YP=F(t,Y)
YP=...
endfunction
```

On définit la valeur initiale dans la variable `Y0`, les temps initiaux et finaux `t0` et `tn`, et on laisse SCILAB choisir la discrétisation et le schéma adapté pour calculer la solution `Y` au temps final `tn` :

```
Y=ode(Y0,t0,tn,F) ;
```

Si on veut la solution aux temps intermédiaires définis dans un vecteur, on écrit

```
t=linspace(t0,tn,n+1) ;
Y=ode(Y0,t0,t,F) ;
```

2 Exercices

Exercice 1

1. Pour le premier exemple d'équation différentielle, le problème (P_1), programmer la méthode d'Euler explicite. Considérer l'intervalle $[0, T]$ avec $T = 1$, $a = -1$ et $y_0 = 1$. Pour un entier n donné (par exemple $n = 10$), considérer une grille $t_i = ih$, $i = 0, \ldots, n$, de l'intervalle $[0, T]$, avec $h = T/n$. Comparer graphiquement la solution numérique avec la solution exacte.

2. Même question pour le schéma de Runge-Kutta d'ordre 4.
3. Pour les deux schémas, faire varier le temps final T ($T = 10$, $T = 100$). Faire aussi varier le nombre de points dans le maillage n ($n = 100$, $n = 1000$). Qu'observe t-on ?
4. Même question en utilisant le solveur de SCILAB.
5. Vérifier numériquement l'ordre de convergence des schémas d'Euler et de Runge-Kutta d'ordre 4. Pour cela, il faut faire une série d'essais numériques avec le paramètre h qui varie ; considérer par exemple une boucle pour $j = 1, \dots, 6$ et définir $n = 10 \times 2^{j-1}$ et $h = T/n$. Visualiser graphiquement la dépendance de l'erreur sur h, ainsi que les graphes de h^p, où p est l'ordre de convergence approprié. Faire les graphes dans l'échelle logarithmique.

Exercice 2 On considére le système différentiel du pendule, soit le problème (P_2).

1. Programmer le schéma d'Euler et la méthode de Runge-Kutta d'ordre 2. Considérer $T = 50$, $g = 10$, $L = 1$, $n = 1000$ ou $n = 10000$, $h = T/n$ et la condition initiale $\varphi_0 = 1$ et $\phi_0 = 0$.
2. Représenter graphiquement l'angle en fonction du temps. Comparer les solutions obtenues avec et les deux schémas.
3. Qu'observe t-on en allant de $n = 1000$ à $n = 10000$? Tous les résultats sont-ils fiables ? Ou y a-t-il des résultats qui semblent suspects ? Quelle méthode semble la plus précise ?

Exercice 3

1. Résoudre à la main l'équation différentielle
$$\begin{cases} x'(t) = tx(t) + x(t)^2 t^3 \\ x(0) = 1. \end{cases}$$
C'est une équation de type Bernoulli
$$x'(t) + P(t)x(t) = Q(t)x(t)^q$$
qu'on peut résoudre en posant $z(t) = x(t)^{1-q}$.
2. Écrire un script SCILAB pour calculer numériquement la solution avec le schéma d'Euler et comparer avec la solution exacte (prendre par exemple $n = 1000$). Représenter les deux solutions sur l'intervalle $[0, 1.2]$ sur le même graphe.
3. Programmer le schéma étudié dans l'exercice 2 :
$$y_{n+1} = y_n + h\left[f(t_n, y_n) + h\left(\frac{1}{2} - a\right) f^{[1]}(t_n, y_n) + haf^{[1]}\left(t_n + \frac{h}{6a}, y_n + \frac{h}{6a} f(t_n, y_n)\right)\right]$$
avec
$$f^{[1]}(t, y) = \frac{\partial f}{\partial t}(t, x) + \frac{\partial f}{\partial x}(t, x) f(t, x).$$
Le schéma dépend d'un paramètre a, qu'on pourra faire varier (par exemple $a = 1, 5, 50$). Discuter l'influence de la variation du paramètre a. Comparer avec la solution exacte.

Exercice 4 On étudie dans cet exercice un système différentiel dit oscillateur de FitzHugh-Nagumo, très utilisé en biologie. Il peut servir par exemple à modéliser la réponse d'une membrane cellulaire à une excitation électrique.
$$\begin{cases} x'(t) = f(x(t)) - y(t) + I_a \\ y'(t) = bx(t) - \gamma y(t) \\ x(0) = x_0 \\ y(0) = y_0 \end{cases}$$
ajusté pour les valeurs :
$$f(x) = x(a - x)(x - 1),$$
avec $0 < a < 1$, $b > 0$, $\gamma > 0$. On prendra par exemple $a = 0.25$, $b = \gamma = 0.002$. On pourra partir d'un système au repos $x_0 = y_0 = 0$. Ici $x(t)$ joue le rôle du potentiel membranaire et $y(t)$ est une variable générale d'excitabilité. La constante I_a joue le rôle d'un courant externe de stimulation, à faire varier pour passer d'une solution stationnaire à des solutions périodiques ($I_a = 0.25$ ou $I_a = 0.2$ par exemple).

1. Programmer en SCILAB la fonction $f(x)$.

Répondre aux questions suivantes pour différentes valeurs du paramètre I_a.

2. On note $M(t) = (x(t), y(t))$. Calculer les solutions stationnaires éventuelles du système (c'est-à-dire telles que $M(t) = Cte$). On pourra utiliser la fonction SCILAB `fsolve`.
3. Tracer sur la même figure les courbes $y = bx/\gamma$ et $y = f(x)+I_a$. Vérifier les résultats de la question précédente.
4. Écrire la fonction second membre $F(t, P) : \mathbb{R} \times \mathbb{R}^2 \mapsto \mathbb{R}^2$ telle que le système différentiel s'écrive sous la forme $M'(t) = F(t, M(t))$. Programmer $F(t, P)$ en SCILAB.
5. Résoudre le système différentiel avec un schéma de Runge-Kutta d'ordre 4 (prendre par exemple le temps final $T = 1000$ et $n = 500$). Tracer dans une figure les graphes $(t, x(t))$ et $(t, y(t))$, et dans la figure de la question 3) la courbe $(x(t), y(t))$. Changer la condition initiale (x_0, y_0) et superposer les deux graphes (x, y) obtenus.
6. Comparer les résultats obtenus avec différents schémas, par exemple Euler et Runge-Kutta d'ordre 4.

Références

[1] M. BUFFAT. *Résolution d'équations différentielles avec « SCILAB »*, DESS Modélisation Numérique en Mécanique et utilisation de codes industriels, Université Claude Bernard Lyon I.

[2] T. DUDOK DE WIT. *Analyse numérique*, Licence 3 de Physique, Université d'Orléans (2010). http ://lpce.cnrs-orleans.fr/∼ddwit/enseignement/cours_analnum.pdf

[3] M. PIERRE. *Compléments de cours pour l'épreuve de modélisation (2010)*, Agrégation Externe de Mathématiques, Université de Poitiers (2010). http ://www-math.univ-poitiers.fr/_pierre/AGREG/AGR/coursagreg2010.pdf

[4] M. VOHRALÍK. *Travaux Dirigés - Bases des méthodes numériques*, Master de Sciences & Technologies, Mention Mathématiques & Applications, Université Pierre et Marie-Curie (2010-2011) http ://www.ann.jussieu.fr/∼vohralik/Enseig/BasMethNum/TP1.pdf