

Analyse numérique des équations différentielles pour l'oral de modélisation, TD1

1 Introduction

L'oral de modélisation n'est pas une épreuve de programmation et une utilisation raisonnable des boîtes noires de MATLAB pour les illustrations numériques est tout aussi appréciée par le jury, l'essentiel de la discussion portant sur la pertinence et la précision des résultats obtenus (les catalogues d'algorithmes sont évidemment à proscrire). A ce titre, il faudra connaître les principales caractéristiques des schémas numériques utilisés. Les schémas numériques les plus connus pour l'intégration des équations différentielles ordinaires sont les schémas d'Euler explicite, de Heun, de Runge Kutta d'ordre 4 (RK4), d'Euler implicite, de Crank Nicolson (ou schéma du point milieu), Adams (c'est une méthode multipas, peu présentée par les candidats mais que le jury aimerait voir un peu plus souvent...). Pour discuter de la validité des résultats, on pensera aux points suivants: la méthode converge-t-elle toujours? y'a-t-il des cas de divergences? des cas où elle ne converge pas vers la solution donnée, des cas où on ne peut pas l'appliquer? Discuter également la vitesse de convergence: si la solution exacte est connue, calculer l'erreur et la tracer en fonction du paramètre de discrétisation (penser au diagramme log-log), comparer avec la vitesse théorique. Si on ne connaît pas la solution exacte, tracer la solution approchée en fonction du paramètre de discrétisation, tracer d'éventuelles quantités conservées (ex: l'énergie totale pour un système mécanique sans dissipation). Discuter des propriétés qualitatives de la solution obtenue (périodicité, convergence vers 0,...). Discuter éventuellement du choix du pas de temps.

D'un point de vue pratique, pendant la durée de préparation (4h), on assemblera les programmes dans une séquence pédagogique et on conservera les résultats numériques issus de calculs longs dans un fichier. Les programmes doivent être aussi souples et interactifs que possible pour répondre plus facilement aux demandes du jury (ex: changer un paramètre, une condition initiale,...).

Le but de ce premier TD est de programmer les algorithmes (ou éventuellement d'utiliser les boîtes noires correspondantes si elles existent) pour l'intégration numérique des équations différentielles ordinaires scalaires ou pour les systèmes différentiels. On mettra en évidence sur un exemple simple les ordres de convergence respectifs. Cependant un ordre de consistance élevé n'est pas toujours la garantie d'une bonne approximation, notamment dans le cas des problèmes raides ou hamiltoniens.

On considère ici le problème de Cauchy:

$$\begin{aligned}y'(t) &= f(t, y(t)), \quad 0 \leq t \leq T, \\y(0) &= y_0,\end{aligned}\tag{1.1}$$

où $f : [0; T] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ est une fonction régulière et $y_0 \in \mathbb{R}^d$.

2 Programmation des algorithmes classiques

Pour l'approximation numérique, on note h le pas de la subdivision uniforme $(t_n)_{n=0..N}$ de l'intervalle $[0, T]$ et y_n est une approximation de $y(t_n)$ pour $0 \leq n \leq N$. On va étudier les méthodes suivantes

- Euler Explicit $y_{n+1} = y_n + hf(t_n, y_n)$
- Heun $y_{n+1} = y_n + \frac{h}{2}(f(t_n, y_n) + f(t_{n+1}, y_n + hf(t_n, y_n)))$
- Runge-Kutta 4 (RK4)

$$\begin{aligned} k_1^n &= f(t_n, y_n), \\ k_2^n &= f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1^n\right), \\ k_3^n &= f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2^n\right), \\ k_4^n &= f\left(t_{n+1}, y_n + hk_3^n\right), \\ y_{n+1} &= y_n + \frac{h}{6}(k_1^n + 2k_2^n + 2k_3^n + k_4^n) \end{aligned}$$

- Euler implicite $y_{n+1} = y_n + hf(t_{n+1}, y_{n+1})$
- Crank-Nicolson

$$\begin{aligned} y_{n+\frac{1}{2}} &= y_n + \frac{h}{2}f\left(t_n + \frac{h}{2}, y_{n+\frac{1}{2}}\right), \\ y_{n+1} &= y_n + \frac{h}{2}f\left(t_n + \frac{h}{2}, y_{n+\frac{1}{2}}\right). \end{aligned}$$

On remarque que les deux derniers schémas sont implicites: il faut résoudre une équation ou un système non linéaire à chaque itération. A chaque étape, on connaît une valeur approchée de y_{n+1} : c'est y_n , on utilise alors l'algorithme de Newton avec y_n comme valeur de départ pour calculer y_{n+1} . On remarque également que le schéma de Crank Nicolson correspond à une itération de Euler implicite sur un demi pas de temps puis d'Euler explicite sur un pas de temps. On rappelle que les méthodes d'Euler sont d'ordre 1, Heun et Crank Nicolson d'ordre 2 et Runge Kutta 4 d'ordre 4.

A faire: Programmer les différentes méthodes pour résoudre numériquement $y'(t) = F(t, y(t))$ où $F : [0, T] \times \mathbb{R} \rightarrow \mathbb{R}$ est une fonction régulière: on veut une fonction qui, à une donnée initiale y_0 , un temps T et un pas d'espace h , renvoie un vecteur avec les valeurs de la solution approchée $(y_n)_{n=0..N}$ aux points de discrétisation. Par exemple:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Methode d'Euler explicite
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function y=eulE(y0,h,T);
y=[y0];yy=y0;
N=floor(T/h);t=0;
```

```

for i=1:N,
yy=yy+h*F(t,yy);
y=[y,yy]; t=t+h;
end;

```

On étudie l'exemple suivant: $y' = -2y$ et $y(0) = 3$. Représenter sur un même schéma les résultats obtenus pour un temps $T = 5$ et $h = 0.1$ ainsi que la solution exacte. Mettre alors en évidence numériquement l'ordre de convergence de chaque méthode: on calculera pour chaque méthode et pour des pas de temps différents ($h = \frac{T}{10}, \dots, \frac{T}{100}$) la quantité $\max_{0..N} |y_n - y(t_n)|$ et on la représentera dans un graphe log-log. A l'aide d'une régression linéaire (commande *polyfit*), calculer les ordres de convergence numériques.

3 Problèmes raides

On a vu au paragraphe précédent que les méthodes utilisées étaient plus ou moins précises. Cependant la méthode *RK4* n'est pas toujours la plus performante. Considérons le problème de Cauchy

$$y' = -\lambda y + 1 + \lambda t, \text{ avec } y(0) = 2, \quad (3.2)$$

dont la solution exacte est $y(t) = t + 2e^{-\lambda t}$. On considère λ grand.

A faire Calculer l'erreur pour les différentes méthodes pour $T = 2$ et des pas de temps $h = 0.01$ et $h = 0.001$ en prenant comme paramètre $\lambda = 500$. Vérifier que pour $h = 0.01$, les méthodes implicites donnent les résultats les plus pertinents.

On peut faire l'étude des suites récurrentes associées aux schémas d'Euler explicite et Euler implicite notées y_n^e, y_n^i :

$$\begin{aligned} y_{n+1}^e &= y_n^e + h(\lambda(t_n - y_n^e) + 1), \\ y_{n+1}^i &= (1 + \lambda h)^{-1}(y_n^i + \lambda h t_{n+1} + h). \end{aligned} \quad (3.3)$$

Comme λ est grand, la solution exacte est rapidement proche de $t \mapsto t$: on cherche à savoir si cette propriété est conservée par les schémas. Pour h fixé, la quantité y_{n+1}^i est proche de t_{n+1} pour λ grand. En revanche si y_n^e est proche de t_n , y_{n+1}^e n'est proche de t_{n+1} que si $h\lambda \ll 1$. Cette condition impose un choix de pas très petit si on veut approcher correctement la solution exacte: une telle équation est dite raide. Le phénomène de raideur est difficile à définir mais en général, il met en jeu un phénomène lent et régulier (le terme t) et un terme rapide avec un effet court (la terme $2e^{-\lambda t}$). En particulier ces problèmes interviennent dans des applications où différentes échelles de temps coexistent (exemple en chimie pour des réactions chimiques avec des vitesses caractéristiques très différentes). Pour être efficace, les méthodes explicites doivent être utilisées avec un pas de temps très court (ce qui peut augmenter considérablement les temps de calcul) et on leur préfère les méthodes implicites dans ce cas précis.

4 Méthodes pour les systèmes conservatifs

On a vu qu'une méthode implicite d'ordre 1 pouvait être plus performante qu'une méthode d'ordre 4 pour un problème raide. Considérons le cas du problème

$$x' = -y, y' = x \text{ avec } x(0) = 1, y(0) = 0. \quad (4.4)$$

La solution est donnée par $[x(t) = \cos t, y(t) = \sin t]$ et la trajectoire dans le plan de phase est le cercle unité. La quantité $H = x^2(t) + y^2(t)$ est constante: on l'appelle l'hamiltonien du système.

A faire Tracer la solution approchée sur l'intervalle $[0, 4\pi]$ et un pas de temps $h = 0.1$ obtenue par les cinq méthodes et tracer également l'évolution de $H_n = x_n^2 + y_n^2$. Expliquer les résultats obtenus en calculant une récurrence linéaire sur H_n dans les cinq cas. Vérifier que le schéma de Crank Nicolson est le seul qui conserve H_n .

Un exemple typique conduisant à un système "hamiltonien" est le système de Lotka Volterra: le comportement des solutions approchées est similaire au cas linéaire mais dans ce cas l'hamiltonien n'est pas conservé (même pour Crank Nicolson, pour lequel on oscille autour de sa valeur théorique).

Références

- [1] M.Crouzeix, A.L. Mignot. *Analyse numérique des équations différentielles*
- [2] E. Hairer, S.P. Norsett, G. Wanner *Solving ordinary differential equations I, II*
- [3] M. Schatzman *Analyse Numérique*