Approximation de solutions d'équations différentielles, schémas numériques.

C. Dossal

Février 2013

1 Le cadre général

On va chercher à approcher numériquement les solutions d'équations différentielles de la forme :

$$y'(t) = f(t, y(t)) \quad \text{avec} \quad y(t_0) = y_0. \tag{1}$$

où f est une fonction continue Lipschitz par rapport à la deuxième variable. Le théorème de Cauchy Lipschitz assure l'existence et l'unicité de la solution. Il arrive que l'on sache résoudre ce problème de manière analytique mais dans un grand nombre de cas on ne connait pas de forme explicite de la solution. On peut alors essayer d'approcher la solution par un schéma numérique. Le principe de toutes les méthodes que nous allons voir est le même : On cherche à estimer la solution y de (EqRef1) en des points régulièrement espacés $(t_i)_{i\in I}$ en partant de $y(t_0)$ qui est connu et en estimant de proche en proche les valeurs de $y(t_i)$ en utilisant le fait que

$$y(t_{i+1}) = y(t_i) + \int_{t_i}^{t_{i+1}} f(t, y(t))dt. \tag{2}$$

De fait, ne connaissant pas y sur l'intervalle $[t_i, t_{i+1}]$ on doit souvent approcher cette intégrale avec les moyens du bord. L'étude des schémas numériques consiste à étudier la convergence et à borner l'erreur de ces schémas. Dans la suite on notera h la distance entre deux points t_i et t_{i+1} ; $\forall i \in \mathbb{N}, h = t_{i+1} - t_i$. On peut faire des schémas avec des pas variables mais nous concentrerons sur les schémas à pas constant. Chaque méthode va essayer d'approcher cette intégrale $\int_{t_i}^{t_{i+1}} f(t, y(t))dt$ par une fonction $h \times \phi(t_i, y(t_i), h)$.

2 Préliminaire

Les 4 équations de référence que nous utiliserons pour tester les différentes métodes sont les suivantes :

$$y'(t) = -y(t) \qquad \text{sous la contrainte } y(0) = 1. \tag{EqRef1}$$

$$y'(t) = y(t) \qquad \text{sous la contrainte } y(0) = 1. \tag{EqRef2}$$

$$y'(t) = -y^2(t) \qquad \text{sous la contrainte } y(0) = 1. \tag{EqRef3}$$

$$y'(t) = -\tan(t)y(t) \qquad \text{sous la contrainte } y(0) = 1. \tag{EqRef4}$$

1. Résoudre ces 4 équations différentielles.
2. Créer les fonctions f1.m et solf1.m suivantes

```
function z=f1(t,y)
z=-y;
```

et

```
function z=solf1(t);
z=-exp(t);
```

3. Créer les fonctions analogues pour les autres équations de référence.

3 Schéma d'Euler explicite

Euler a été le premier à proposer un schéma d'approximation. Ce schéma consiste à approcher l'intégrale $\int_{t_i}^{t_{i+1}} f(t,y(t))dt$ par $\phi(t_i,y(t_i),h) = f(t_i,y(t_i))$. Le schéma dit d'Euler explicite s'écrit alors

$$y_{n+1} = y_n + h \times f(t_n,y(t_n)). \tag{3}$$

4. On considère la solution approchée par la méthode d'Euler de l'équation (EqRef1). Si on pose $h = \frac{T}{n}$ montrer que pour tout T, la solution obtenue par la métode d'Euler converge vers la solution analytique au point T quand n tend vers $+\infty$.
5. Montrer que si n est fixé et que T tend vers $+\infty$, y_n s'éloigne de 0. On dit que la méthode d'Euler est un schéma instable. On ne l'utilise qu'en temps fini.
En Matlab la méthode d'Euler peut se coder de la manière suivante :

```
function y=MethodeEuler(name,t0,T,y0,h)
ff=str2func(name);
N=floor(T/h)+1;
t=[t0:h:t0+(N-1)*h];
y=zeros(N,1);
y(1)=y0;
for k=1:N-1
    y(k+1)=y(k)+h*ff(t(k),y(k));
end
```

On pourra utiliser la fonction suivante pour tester la méthode :

```
function er=TestMethode(Methode,ff,solff,t0,T,y0,h)
solf=str2func(solff);
N=floor(T/h)+1;
t=[t0:h:t0+(N-1)*h];
if Methode(1)=='A';
    Meth=str2func(Methode(1:end-1));
    r=str2num(Methode(end));
    [yn,tn,fn]=Meth(r,ff,t0,y0,h,N);
    else
Meth=str2func(Methode);
y=Meth(ff,t0,T,y0,h);
end
plot(t,y);
z=zeros(N,1);
for k=1:N
   z(k)=solf(t(k));
```

```
end
hold on;plot(t,z,'r');title(Methode)
hold off;
er=max(abs(z-y));
```

6. Tester la méthode d'Euler sur les équations de référence en faisant varier la valeur du pas h.
7. Pour ces différentes équations comment évolue l'erreur maximale en fonction du pas h ?

4 Méthode du point milieu

Dans la méthode du point milieu on essaie d'être un peu plus précis dans l'estimation de l'intégrale $\int_{t_i}^{t_{i+1}} f(t,y(t))dt$ pour cela on utilise un point intermédiaire entre y_n et y_{n+1} que nous noterons $y_{n+\frac{1}{2}}$:

$$y_{n+\frac{1}{2}} = y_n + \frac{h}{2} f(t_n, y_n).$$

On a alors

$$y_{n+1} = y_n + hf(t_n + \frac{h}{2}, y_{n+\frac{1}{2}}) \quad \text{et toujours} \quad t_{n+1} = t_n + h.$$

8. A partir du programme précédent créer une nouvelle fonction *MethodePointMilieu* qui résout numériquement une EDO par cette méthode.
9. Tester cette nouvelle méthode sur les équations de référence en faisant varier le pas h. Comparer les erreurs de cette méthode et de la méthode d'Euler.
10. Comment varie l'erreur en fonction de h ?
11. Un bon indicateur de la complexité d'une méthode est le nombre de fois par étape où l'on doit évaluer la fonction f. Pour un nombre égal d'évaluations de f quelle méthode assure l'erreur la plus faible ?

5 Consistance, stabilité et convergence

L'objectif d'un schéma numérique est de fournir une solution approchée qui converge vers la solution du problème de Cauchy (1) quand h tend vers 0.
Pour estimer cette convergence on définit la notion d'erreur de consistance. Si une méthode de résolution numérique est de la forme $y_{n+1} = y_n + h\phi(t_n, y_n, h_n)$,

Définition 1. *L'erreur de consitance e_n relative à la solution exacte z de* (1) *est définie par*

$$e_n = z(t_{n+1}) - z(t_n) - h\phi(t_n, z(t_n), h)$$

C'est l'erreur que l'on commetrait dans le calcul de $z(t_{n+1})$ si on connaissait exactement $z(t_n)$ par la méthode numérique utilisée. En pratique on connait rarement la valeur exacte de $z(t_n)$ ...

Définition 2. *On dit que la méthode numérique est consistante si pour toute solution exacte z,* $\lim_{N\to\infty} \sum_{0\leqslant n\leqslant N} |e_n| = 0$.

Définition 3. *On dit que la méthode est stable s'il existe une constante $S \geqslant 0$, appelée constante de stabilité, telle que pour toutes suites (y_n) et $(\tilde{y}_n)$ définie par*

$$\begin{aligned} y_{n+1} &= y_n + h\phi(t_n, y_n, h), \quad 0 \leqslant n \leqslant N \\ \tilde{y}_{n+1} &= \tilde{y}_n + h\phi(t_n, \tilde{y}_n, h) + \varepsilon_n, \quad 0 \leqslant n \leqslant N \end{aligned}$$

on ait

$$\max_{0 \leqslant n \leqslant N} |\tilde{y}_n - y_n| \leqslant S \left(|\tilde{y}_0 - y_0| + \sum_{0 \leqslant n \leqslant N} |\varepsilon_n| \right)$$

Ainsi si la méthode est stable, même si on commet une petite erreur à chaque étape par rapport au schéma souhaité, et en pratique c'est toujours le cas à cause des erreurs d'arrondis, on peut contrôler l'erreur globale pourvu que la somme de chacune des erreurs soit contrôlée.

Définition 4. *On dit que la méthode est convergente si pour toute solution exacte z, la suite $y_{n+1} = y_n + h\phi(t_n, y_n, h)$ vérifie*

$$\max_{0 \leqslant n \leqslant N} |y_n - z(t_n)| \to 0$$

quand $y_0 \to z(t_0)$ et quand $h \to 0$.

La quantité $\max_{0 \leqslant n \leqslant N} |y_n - z(t_n)|$ s'appelle l'erreur globale, c'est cette erreur qui importe dans la pratique.

12. En prenant $\tilde{y}_n = z(t_n)$ montrer que si la méthode est stable et consistante alors elle est convergente.
13. Soit z une solution exacte de (1), soit $T > 0$ fixé et $N \in \mathbb{N}$, on note $h = \frac{T}{N}$ et

$$e_n = z(t_{n+1}) - z(t_n) - h\phi(t_n, z(t_n), h).$$

(a) Comment appelle t-on e_n ?

(b) Montrer que pour tout $n < N$, il existe $c_n \in]t_n, t_{n+1}[$ tel que $e_n = h(f(c_n, z(c_n)) + \phi(t_n, z(t_n), h))$.

(c) On pose

$$\alpha_n = f(c_n, z(c_n)) - \phi(c_n, z(c_n), 0) \quad \text{et} \quad \beta_n = \phi(c_n, z(c_n), 0) - \phi(t_n, z(t_n), h)$$

Montrer que pour tout $\varepsilon > 0$ il existe $h_0 > 0$ tel que si $h \leqslant h_0$, $\forall n \leqslant N$, $|\beta_n| \leqslant \varepsilon$.

(d) En déduire que si $h < h_0$

$$\Big| \sum_{0 \leqslant n < N} |e_n| - \sum_{0 \leqslant n < N} h|\alpha_n| \Big| \leqslant T\varepsilon.$$

(e) Justifier que

$$\lim_{N \to \infty} \sum_{0 \leqslant n < N} h|\alpha_n| = \int_{t_0}^{T} |f(t, z(t)) - \phi(t, z(t), 0)| dt.$$

(f) En déduire que la méthode est consistante si et seulement si

$$\forall t \in [t_0, T], \quad \phi(t, z(t), 0) = f(t, z(t)).$$

Lemme de Gronval discret

14. Soit $(\theta_n)_{n\in\mathbb{N}} \in \mathbb{R}^+$ et $(\varepsilon_n)_{n\in\mathbb{N}} \in \mathbb{R}$ deux suites telles que $\theta_{n+1} \leqslant (1+\Lambda h)\theta_n + |\varepsilon_n|$. Justifier que $1+\Lambda h \leqslant e^{\Lambda h}$.
15. Montrer par récurrence que pour tout $n \in \mathbb{N}$

$$\theta_n \leqslant e^{\Lambda(t_n-t_0)}\theta_0 + \sum_{i=0}^{n-1} e^{\Lambda(t_n-t_{i+1})}|\varepsilon_i|. \tag{4}$$

16. Dans cette question on suppose que ϕ est Lipschitz de constante Λ par rapport à la deuxième variable. On considère deux suites (y_n) et $(\tilde{y}_n)$ définies par

$$\begin{aligned} y_{n+1} &= y_n + h\phi(t_n, y_n, h) \\ \tilde{y}_{n+1} &= \tilde{y}_n + h\phi(t_n, \tilde{y}_n, h) + \varepsilon_n \end{aligned}$$

   (a) Montrer que la suite $\theta_n = |\tilde{y}_n - y_n|$ vérifie

$$\theta_{n+1} \leqslant (1+\Lambda h)\theta_n + |\varepsilon_n|.$$

   (b) En utilisant le lemme de Gronval discret que

$$\max_{0\leqslant n\leqslant N} |\tilde{y}_n - y_n| \leqslant e^{\Lambda T}\left(|\tilde{y}_0 - y_0| + \sum_{n=0}^{N}|\varepsilon_n|\right).$$

   (c) En déduire que la méthode numérique associée à la fonction ϕ est stable de constante de stabilité $e^{\Lambda T}$.

17. Que peut-on dire d'une méthode de résolution associée à une fonction ϕ Lipshitz par rapport à la deuxième variable et telle que $\phi(t,y,0) = f(t,y)$?
18. En déduire que si f est Lipschitz de constante k par rapport à la deuxième variable, le schéma d'Euler explicite est convergent et donner sa constante de stabilité.
19. On rappelle que pour la méthode du point milieu

$$\phi(t,y,h) = f\left(t + \frac{h}{2}, y + \frac{h}{2}f(t,y)\right).$$

   Si f est k-Lipschitz montrer que

$$|\phi(t,y_1,h) - \phi(t,y_2,h)| \leqslant k\left(|y_1 - y_2| + \frac{h}{2}|f(t,y_1) - f(t,y_2)|\right).$$

20. En déduire que si f est $k-$Lipschitz la méthode du point milieu est stable et donner sa constante de stabilité.

6 Ordre d'une méthode numérique de résolution d'une EDO

Définition 5. *On dit qu'une méthode à 1 pas est d'ordre p si pour toute solution exacte z de (1) où f est de classe C^p, il existe une constante C telle que l'erreur de consistance relative à z vérifie*

$$|e_n| \leqslant Ch^{p+1}, \quad \forall n < N.$$

21. Montrer que si une méthode est d'ordre p alors

$$\sum_{n=0}^{N-1} |e_n| \leqslant CTh^p.$$

22. **Ordre de la méthode d'Euler.**
On suppose que f est de classe C^1 et on note z une solution exacte de (1) avec $f \in C^1$ et on note e_n l'erreur de consistance de la méthode d'Euler.
   (a) Montrer que

$$e_n = \frac{1}{2}h^2 z^{(2)}(t_n) + o(h^2).$$

   En déduire une expression de e_n en fonction des dérivées partielles de f au point (t_n, y_n) et que la méthode d'Euler est d'ordre 1.

23. **Ordre de la méthode du point milieu.**
On suppose maintenant que f est de classe C^2 et on considère la méthode du point milieu. Si z est une solution de (1), on a $z'(t) = f(t, z(t))$ et z est C^3, on en déduit que la dérivée troisième $z^{(3)}$ de z solution exacte de (1) s'exprime comme combinaison linéaire de produits de dérivées partielles de f d'ordre 1 et 2 et est donc bornée $[t_0, t_0 + T]$.
   (a) Montrer que l'erreur de consistance e_n s'écrit $e_n = \varepsilon_n + \varepsilon_n'$ avec

$$\varepsilon_n = z(t_{n+1}) - z(t_n) - hz'(t_n + \frac{h}{2})$$
$$\varepsilon_n' = hz'(t_n + \frac{h}{2}) - (y_{n+1} - z(t_n)).$$

   (b) Montrer qu'il existe C_1 tel que $\varepsilon_n \leqslant C_1 h^3$.
   (c) Montrer que

$$\varepsilon_n' = h\left(f\left(t_n + \frac{h}{2}, z(t_n + \frac{h}{2})\right) - f\left(t_n + \frac{h}{2}, y_{n+\frac{1}{2}}\right)\right)$$

   (d) En utilisant l'égalité des accroissements finis montrer qu'il existe C_2 tel que $\varepsilon_n' \leqslant C_2 h^3$.
   (e) En déduire que la méthode du point milieu est une méthode d'ordre 2.
   La fonction suivante permet d'estimer numériquement l'ordre d'une méthode en utilisant plusieurs valeurs du pas h.

```
function r=OrdreMethode(Methode,fonction,solution,H,t0,y0,T)
f=str2func(fonction);
sol=str2func(solution);
K=length(H);
for k=1:K
    h=H(k);
```

```
        N=floor(T/h)+1;
        t=[t0:h:t0+(N-1)*h];
        z=zeros(N,1);
        for j=1:N
            z(j)=sol(t(j));
        end
        if Methode(1)=='A';
            Meth=str2func(Methode(1:end-1));
            r=str2num(Methode(end));
            y=Meth(r,fonction,t0,T,y0,h);
        else
            Meth=str2func(Methode);
            y=Meth(fonction,t0,T,y0,h);
        end
        er(k)=max(abs(y-z));
    end
    plot(log(er),log(H),'ro');title('Diagramme Log log de l Erreur en fonction du pas');
    hold on;
    plot(log(er),log(H));
    hold off;
    coef=polyfit(log(H),log(er),1);
    r=coef(1);
```

On peut l'utiliser en lançant par exemple :

```
>> r=OrdreMethode('MethodeEuler','f1','solf1',[0.0001,0.001,0.01,0.1],0,1,1)
```

(f) Utiliser cette fonction pour estimer numériquement l'ordre des méthodes d'Euler et du point milieu sur les 4 équations de référence.

(g) Cela correspond il aux calculs théoriques ?

(h) Que se passe t-il si on utilise de très petites valeurs du pas pour estimer l'ordre ? Expliquer le phénomène.

24. La méthode de Heun consite à utiliser la formule de récurrence suivante

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

Ecrire un programme qui met en place cette méthode

25. Tester cette nouvelle méthode sur les équations de référence et évaluer numériquement son ordre.

7 Modèle de Volterra pour les systèmes proie-prédateur

En 1926, Volterra propose un modèle simple de système *proie-prédateur* afin d'expliquer les oscillations des campagnes de pêche dans l'adriatique. Il s'écrit

$$\begin{cases} S'(t) &= S(t)(a-bR(t)) \\ R'(t) &= R(t)(-c+dS(t)) \end{cases} \tag{5}$$

où t représente le temps, $S(t)$ et $R(t)$ représente respectivement le nombre de proies et de prédateur à l'instant t et a, b, c et d sont des paramètres positifs.

26. Montrer qu'en posant $u(\tau) = \frac{d}{c}S(t)$, $v(\tau) = \frac{b}{a}R(t)$, $\tau = at$ et $\alpha = \frac{c}{a}$ on obtient le système suivant.
$$\begin{cases} u'(\tau) &= u(\tau)(1-v(\tau)) \\ v'(\tau) &= \alpha v(\tau)(u(\tau)-1) \end{cases} \tag{6}$$
27. Soit $H(t) = \alpha u(t) + v(t) - \alpha \ln(u(t)) - \ln(v(t))$.
28. Calculer $H'(t)$. On dit que H est une intégrale première du système.
29. Soit f la fonction définie de $]0,+\infty[$ dans $\mathbb{R}$ par $f(x) = x - \ln x$. Etudier les variations de f.
30. En déduire que quelques soient les valeurs initiales de $u(0)$ et $v(0)$ choisies, les fonctions u et v sont bornées supérieurement par un réel $M > 0$ et inférieurement par un réel $m > 0$.
31. Montrer qu'à une valeur de $u(t)$ correspondent au plus 2 valeurs de $v(t)$.
32. Expliquer ce que réalise le programme suivant :
```
function [u,v,H]=Voltera(a,u0,v0,T,h)
N=ceil(T/h);
S=zeros(N,1);
R=zeros(N,1);
H=zeros(N,1);
u(1)=u0;
v(1)=v0;
for k=2:N
  u(k)=u(k-1)+h*u(k-1)*(1-v(k-1));
  v(k)=v(k-1)+a*h*v(k-1)*(u(k-1)-1);
  H(k)=a*v(k)+u(k)-a*log(u(k))-log(v(k));
end
%close all;
plot(u,v);
figure;plot(H);
```
A quel schéma d'approximation numérique correspond t-il ?
33. Faire un test avec $a = 1$, $u_0 = 0.4$, $v_0 = 0.4$, $T = 100$ et $h = 0.01$ et interpréter graphiquement les courbes.
34. Ces courbes sont-elles cohérentes avec l'analyse théorique ? Pourquoi ? Tracer plusieurs courbes avec les mêmes paramètres avec $T = 20$ en faisant varier h. Qu'observe t-on ?
35. Reprendre les valeurs précédentes en prenant $v_0 = 2$.
36. Que se passe t-il si $u_0 = 1$ et $v_0 = 1$?
37. Tester également $u_0 = 0.95$ et différentes valeurs de a ?
38. Modifier le programme de manière à approcher la solution réelle en utilisant la méthode du point milieu.
39. Comparer les trajectoires et l'évolution de la valeur de H pour différentes valeurs de h avec cette nouvelle méthode et la méthode précédente. Qu'en concluez vous ?

8 Equations différentielles d'ordre 2.

L'exemple précédent sur le modèle de Volterra a permis de voir que les méthodes numériques pouvaient être utilisés pour les systèmes différentiels d'ordre 1.
Cette remarque permet d'utiliser ces méthodes numériques pour résoudre certaines équations d'ordre 2. En effet si on doit résoudre une équation sclalaire du type

$$y^{(2)} = f(y', y, t) \quad \text{avec } y'(t_0) = z_0 \text{ et } y(t_0) = y_0 \tag{7}$$

on peut écrire le système différentiel d'ordre 1 équivalent :

$$\begin{cases} z' &= f(z,y,t) \quad \text{avec } z(t_0) = z_0 \\ y' &= z \quad \text{avec } y(t_0) = 0. \end{cases} \tag{8}$$

40. Ecrire le système différentiel équivalent à l'équation suivante :

$$y^{(2)} = \frac{ty'}{2} - y + 3 \quad \text{avec } y(0) = 1 \text{ et } y'(0) = 0.$$

41. Après avoir remarqué que $z(t) = t^2 + 1$ est solution, résoudre numériquement ce système en utilisant la méthode du point milieu.
Equation de la gravitation
Si on note Y le vecteur tridimensionnel de la position d'une planète dans un référentiel héliocentrique (le soleil est au point $(0,0,0)$) la force de gravitation du soleil sur ce corps est égal à $-GmM_S\frac{Y}{\|Y\|^3}$, où m désigne la masse du corps, M_S celle du soleil et G est la constante de gravitation. Nous rappelons que la relation fondamentale de la dynamique assure que la somme des forces qui s'exercent sur un corps est égale à la masse multipliée par l'accéleration de ce corps.
42. Montrer que le vecteur Y est solution d'une équation différentielle de la forme

$$Y^{(2)} = \frac{-KY}{\|Y\|^3}.$$

Pour simplifier les calculs on va se placer dans le plan, c'est à dire que $Y = (x, y)$.
43. Ecrire un système différentiel d'ordre 1 dont x et y sont les solutions.
44. Compléter le programme suivant en utilisant la méthode d'Euler pour résoudre ce système numériquement.

```
function UnCorpsEuler(x01,x02,v01,v02,h,T)
K=2;
z0=0;y0=0;
N=floor(T/h);
x1=zeros(N,1);
x2=x1;v1=x1;v2=x1;
x1(1)=x01;
x2(1)=x02;
v1(1)=v01;
v2(1)=v02;
for k=1:N
```

```
    x1(k+1)=x1(k)+
    x2(k+1)=x2(k)+
    v1(k+1)=v1(k)-
    v2(k+1)=v2(k)-
end
V=400;
a=6;
c1=0.5*(z0+x01);
c2=0.5*(y0+x02);
for k=1:V:N
plot(x1(k),x2(k),'ro');title('Methode d Euler');
hold on;
plot(x1(1:k),x2(1:k),'k');
plot(y0,z0,'ko');

axis([c1-a c1+a c2-a a+c2])
hold off;
pause(0.001);
end
```

45. A quoi correspondent les 4 données intitiales ?
46. Tester différentes vitesses initiales en utilisant des paramètres assurant que la planète fasse plusieurs tours du soleil.
47. Proposer un programme analogue pour la méthode du point milieu et comparer les deux méthodes.
48. Que se passe t-il si la vitesse initiale est trop importante ?