Approximation de solutions d'équations différentielles, schémas numériques. Méthodes Multipas, Méthodes implicites.

C. Dossal

Avril 2012

Cadre général

Comme dans le cas des méthodes à un pas, 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 de constante de Lipschitz k. On cherche à résoudre numériquement cette équation sur l'intervalle $[t_0, t_0 + T]$. On utilisera un pas constant de h et donc $t_n = t_0 + nh$ et on notera comme précédemment $y_n = y(t_n)$ la solution approchée de l'équation différentielle (1) issue de la méthode numérique utilisée. On notera z la solution exacte.
Dans cette feuille on va étudier les méthodes multipas c'est à dire les méthodes qui utilisent plussieurs valeurs de y_m avec $m \leqslant n$ pour estimer y_{n+1}. Plus précisément la plupart du temps on utilise les valeurs $f_m = f(t_m, y_m)$ pour plusieurs valeurs de $m \leqslant n$ pour estimer y_{n+1}. C'est pourquoi dans les codes matlab en plus de conserver les valeurs de $(y_n)_{n \leqslant N}$ on va également conserver les valeurs de $(f_n)_{n \leqslant N}$ qui pourront être utiles pour calculer plusieurs valeurs de y_k. On dira qu'une méthode est à r pas si elle utilise les valeurs de y_m pour $m \geqslant n+1-r$ pour calculer y_{n+1}.
Toutes ces méthodes nécessitent d'être initialisées pour les premières valeurs de y_n. Ainsi pour une méthode à r pas on initialise souvent la méthode pour les $r-1$ premières étapes en utilisant des méthodes à 1 pas d'ordre équivalent. On peut parfois utiliser des méthodes à pas multiples pour l'initialisation mais avec un nombre de pas inférieur.
L'intérêt de ces méthodes réside dans le fait de limiter les calculs, en effet dans les méthodes à 1 pas d'ordre élevé comme RK4, les valeurs intermédiaires ne sont utilisées qu'une seule fois. La plupart du temps dans les méthodes multipas, une seule évaluation supplémentaire de f est nécessaire à chaque étape. Cependant ces méthodes multipas peuvent souffrir d'un manque de stabilité.
D'une manière générale une méthode multipas peut s'écrire de la manière suivante :

$$\begin{cases} y_{n+1} = & \sum_{i=0}^{r} \alpha_i y_{n-i} + h \sum_{i=0}^{r} \beta_i f_{n-i} \\ t_{n+1} = & t_n + h \\ f_{n+1} = & f(t_{n+1} y_{n+1}) \end{cases}$$

Dans la suite nous allons tester les différentes méthodes sur 3 EDO de référence, déjà utilisées dans le feuille précédente.

$$y' = -y^2 \text{ avec } y_0 = 1$$
$$y' = -y \text{ avec } y_0 = 1$$
$$y' = y \text{ avec } y_0 = 1.$$

Pour tester ces différentes EDO nous allons créer des fonctions matlab associées :

```
function f=f1(t,y)
f=-y^2;
```

et

```
function z=solf1(t)
z=1/(1+t);
```

pour la première fonction.

1. Ecrire les 6 fonctions associées aux trois EDO. On utilisera comme nom de fonctions *f1, f2, f3 et solf1, solf2 et solf3*.
   Pour initialiser les différentes méthodes multipas on pourra utiliser la méthode de Runge-Kutta 4 donnée par la fonction suivante :

```
function [yn,tn,fn]=RK4(name,t0,y0,h,r)
f=str2func(name);
%Fonction qui renvoie les r+1 premieres valeurs d'un schema
% de resolution de l'EDO y=f(t,y) par la méthode de RK4
%ou f est une fonction dont le nom est donne en parametre sous
% la forme d une chaine de caracteres.
yn=zeros(r+1,1);tn=[t0:h:t0+r*h];fn=yn;
yn(1)=y0;
for k=1:r
tn1=tn(k);yn1=yn(k);fn(k)=f(tn1,yn1);fn1=fn(k);
tn2=tn1+h/2;yn2=yn1+h*fn1/2;fn2=f(tn2,yn2);
tn3=tn1+h/2;yn3=yn1+h*fn2/2;fn3=f(tn3,yn3);
tn4=tn1+h;yn4=yn1+h*fn3;fn4=f(tn4,yn4);
yn(k+1)=yn(k)+h*(fn1/6+fn2/3+fn3/3+fn4/6);
end
fn(r+1)=f(tn(r+1),yn(r+1));
```

L'ensemble des fonctions résolvant une EDO devra respecter la même syntaxe avec les mêmes paramètres d'entrée et de sortie dans le même ordre. Pour les méthodes de Adams-Bashforth et Adams-Moulton le paramètre *r* sera le premier paramètre d'entrée.
Pour tester une méthode sur une EDO on pourra utiliser la fonction suivante :

```
function er=TestMethode(Methode,ff,solff,t0,y0,T,h)
solf=str2func(solff);
N=floor(T/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);
[yn,tn,fn]=Meth(ff,t0,y0,h,N);
end
plot(tn,yn);
z=zeros(N,1);
for k=1:N+1
   z(k)=solf(tn(k));
end
hold on;plot(tn,z,'r');title(Methode)
hold off;
er=max(abs(z-yn));
```

Pour évaluer numériquement l'ordre d'une méthode sur une EDO on pourra utiliser la fonction suivante :

```
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);
    t=[t0:h:T];
    N=length(t);
    for j=1:N
        z(j)=sol(t(j));
    end
    if Methode(1)=='A';
        Meth=str2func(Methode(1:end-1));
        r=str2num(Methode(end));
         [yn,tn,fn]=Meth(r,fonction,t0,y0,h,N-1);
    else
        Meth=str2func(Methode);
         [yn,tn,fn]=Meth(fonction,t0,y0,h,N-1);
    end
    er(k)=max(abs(yn-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 utilisera le vecteur $H = [0.1, 0.03, 0.01, 0.003, 0.001, 0.0003, 0.0001]$. Si la courbe affichée n'est pas proche d'une droite, on évaluera l'ordre en restreignant les valeurs de H, on pourra prendre par exemple $H1 = H(2 : end-2)$.

2. Tester la méthode RK4 sur chacune des équations de référence.
3. Evaluer numériquement l'ordre de la méthode RK4 sur les 3 exemples de référence.

1 Méthodes implicites exemple de la méthode de Crank-Nicolson

Méthode de Newton.
La méthode de Newton est une méthode numérique permettant de déterminer les 0 d'une fonctions de manière itérative. Si x_0 est suffisamment proche d'un 0 d'une fonction f on peut montrer que la suite définie par

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

converge vers un zéro de f.

4. En utilisant une approximation numérique de la dérivée d'ordre utilisant $f(x+h)$ et $f(x-h)$ écrire une fonction Newton

```
function x=Newton(x0,name,h)
```

qui cherche une 0 de la fonction indiquée par *name* proche de $x0$. On pourra faire 100 itérations ou s'arrêter quand deux valeurs successives sont proches à 10^{-10} près.

5. Tester la fonction sur des équation polynomiales simples.
6. La Méthode de Crank-Nicolson peut être définie par la formule suivante :

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

Le calcul du point milieu est implicite. La seconde étape est explicite.
Ecrire une équation dont $y_{n+\frac{1}{2}}$ est un 0.

7. Ecrire une fonction

```
function [yn,tn,fn]=CrankNicolson(ff,t0,y0,h,N)
```

qui résout une EDO par cette méthode.

8. Tester cette méthode sur les EDO de référence et estimer son ordre sur ces mêmes EDO.

2 Méthodes de Nyström et Milne

2.1 Méthode de Nyström

La méthode de Nyström peut être résumée par la formule suivante :

$$y_{n+1} = y_{n-1} + hf_n \quad \forall n \geqslant 1$$

9. Pour quelle valeur de r, cette méthode est-elle à r pas ?
10. De quelle méthode à un pas se rappoche t-elle ?
11. Ecrire une fonction matlab *Nystrom* qui résout numériquement une EDO en utilisant cette méthode. On pourra utiliser une méthode du point milieu ou la méthode RK4 pour l'initialisation.
12. Estimer l'ordre de la méthode sur les EDO de référence.
13. Pour la seconde EDO, qu'observe t-on si on prend $h = 0.1$ et $T = 10$?
14. Faites la même expérience avec la méthode RK4.
Comment interprétez vous ce résultat numérique ?

2.2 Méthode de Milne

La méthode de Milne est décrite par la relation suivante :

$$y_{n+1} = y_{n-3} + h\left(\frac{8}{3}f_n - \frac{4}{3}f_{n-1} + \frac{8}{3}f_{n-2}\right) \quad \forall n \geqslant 1$$

15. Ecrire une fonction matlab qui résout une EDO en utilisant la méthode de Milne en utilisant une initialisation par la méthode de RK4.
16. Tester la fonction sur les 3 EDO de référence.
17. Estimer son ordre numériquement.
18. Pour la seconde EDO, qu'observe t-on si on prend $h = 0.1$ et $T = 10$?

3 Méthodes d'Adams-Bashford

Si z est la solution exacte de l'équation (1) elle vérifie

$$z(t_{n+1}) = z_{t_n} + \int_{t_n}^{t_{n+1}} f(t, z(t))dt$$

L'idée des méthodes d'Adams-Bashford est d'approcher $f(t, z(t))$ dans l'intégrale par un polynôme interpolateur de Lagrange calculé à partir des valeurs précédentes de de la solution. Ainsi si on note $p_{n,r}$ le polynôme interpolant la z aux r points $(t_{n-i})_{0 \leqslant i \leqslant r}$ on a l'approximation :

$$z(t_{n+1}) \approx z_{t_n} + \int_{t_n}^{t_{n+1}} p_{n,r}(t)dt$$

La méthode d'Adams-Bashforth à $r+1$ pas AB_{r+1} s'écrit donc

$$y_{n+1} = y_n + h\sum_{i=0}^{r} b_{i,r} f_{n-i} \text{ avec } b_{i,r} = \frac{1}{h}\int_{t_n}^{t_{n+1}} L_{i,r}(t)dt.$$

où les $L_{i,r}$ sont les polynômes de Lagrange associés.

19. Expliciter $L_{0,0}$ et la Méthode AB_{r+1} pour $r = 0$. Quel nom porte cette méthode ?
20. Expliciter la méthode AB_2 après avoir calculé les valeurs de $b_{0,1}$ et $b_{1,1}$.
    Les coeffcients de la méthode AB_{r+1} sont calculés par la fonction suivante :

```
function b=CoefsAB(r)
if r==0
    b=1;
else
    b=zeros(r+1,1);
    R=[-r:0];
    int=1./[r+1:-1:1];
    for k=1:length(R)
        if k==1
            R1=R(2:end);
        else
            if k==length(R)
            else
                R1=R(1:end-1);
            end
            R1=[R(1:k-1),R(k+1:end)];
        end
    P=poly(R1);
    P1=P/polyval(P,R(k));
    b(k)=sum(P1.*int);
    end
end
```

21. Ecrire une fonction Matlab AB

    ```
    function [yn,tn,fn]=AB(r,ff,t0,y0,h,N)
    ```

    qui résout une EDO en utilisant la méthode AB_{r+1} en initialisant les premières valeurs en utilisant la méthode RK4.
22. Tester cette méthode sur les 3 EDO de référence.

23. Estimer numériquement l'ordre de ces méthodes en utilisant les 3 EDO de référence.
Stabilité des méthodes AB_{r+1}.
On considère la suite récurrente perturbée

$$\begin{cases} \tilde{y}_{n+1} = & \tilde{y}_n + h\sum_{i=0}^{r} b_{i,r}\tilde{f}_{n-i} + \varepsilon_n, \quad r \leqslant n < N \\ \tilde{f}_{n-i} = & f(t_{n-i}, \tilde{y}_{n-i}). \end{cases}$$

Posons $\theta_n = \max_{0 \leqslant i \leqslant n} |\tilde{y}_i - y_i|$.
Montrer que

$$\theta_{n+1} \leqslant (1 + \beta_r kh)\theta_n + |\varepsilon_n|.$$

où β_r est une valeur à expliciter et k la constante de Lipschitz de f par rapport à la deuxième variable.

24. En utilisant le lemme de Gronval (cf feuille précédente) déterminer la constante de stabilité de la méthode AB_{r+1}.
25. Donner cette valeur dans le cas des méthodes AB_1, AB_2, AB_3 et AB_{10}.
Consistance des méthodes AB_{r+1}**.**

26. Justifier qu'il existe $\theta \in [t_n, t_{n+1}]$ tel que l'erreur de consistance e_n de la méthode AB_{r+1} s'écrive

$$e_n = h(z'(\theta) - p_{n,r}(\theta)).$$

On rappelle qu'il existe $x \in [t_{n-r}, t_n]$ tel que

$$z'(\theta) - p_{n,r}(\theta) = \frac{1}{(r+1)!} z^{(r+2)}(x)\pi_{n,r}(x) \quad \text{où} \quad \pi_{n,r}(t) = \prod_{i=0}^{r}(t - t_{n-i}).$$

27. Justifier que si $x \in [t_{n-j}, t_{n-j+1}]$ on a $|x - t_{n-i}| \leqslant h(1 + |j - i|)$.
28. En déduire que

$$|\pi_{n,r}(x)| \leqslant h^{r+1}(r+1)!$$

29. En déduire que la méthode AB_{r+1} est d'ordre $r+1$.

4 Méthodes d'Adams-Moulton

Les méthodes d'Adams-Moulton (AM_{r+1}) reprennent l'idée des méthodes d'Adams-Bashforth mais en y incluant le point y_{n+1}. Ces méthodes sont ainsi des méthodes implicites. Le schéma s'écrit :

$$y_{n+1} = y_n + h\sum_{i=-1}^{r} b^{\star}_{i,r} f_{n-i} \text{ avec } b^{\star}_{i,r} = \frac{1}{h}\int_{t_n}^{t_{n+1}} L^{\star}_{i,r}(t)dt.$$

où le polynôme $L^{\star}_{i,r}$ est le polynôme d'interplolation associé aux instants $(t_{n-i})_{-1 \leqslant i \leqslant r}$.
Cette méthode est implicite car la valeur de f_{n+1} est inconnue et dépend en théorie de y_{n+1}.

30. Calculer les coefficients $b^{\star}_{i,0}$ et $b^{\star}_{i,1}$.
31. En modifiant la fonction calculant les coefficients des méthodes AB_{r+1}, écrire une fonction :

```
function b=CoefsAM(r)
```

et vérifier les valeurs obtnenues avec les valeurs théoriques (en $r = 0$ vous venez de les calculer).

On note $u_n = y_n + h\sum_{0 \leqslant i \leqslant r} b^{\star}_{i,r} f_{n-i}$.

32. Déterminer une équation implicite utilisant u_n dont y_{n+1} est solution.
33. En déduire que si $h < \frac{1}{|b_{-1,r}|k}$ on peut mettre en place un schéma itératif que l'on décrira permettant d'approcher y_{n+1}.
34. Ecrire une fonction Matlab

```
function [yn,tn,fn]=AM(r,ff,t0,y0,h,N)
```

qui résout numériquement une équation différentielle en utilisant la méthode AM_{r+1}. On initialisera la méthode en utilisant RK4, et à chaque étape du calcul de y_{n+1} on utilisera une initialisation en utilisant la méthode AB_{r+1} et on s'arrête quand deux valeurs estimées consécutives de y_{n+1} sont proches à 10^{-10}.

35. Tester ces méthodes sur les EDO de référence.
36. Evaluer numériquement l'ordre de ces méthodes.
37. Les résultats sont-ils cohérents avec les résultats théoriques.
38. La constante de stabilité de la méthode de AB_{r+1} est proche de $e^{\beta^{\star}_r kT}$ où $\beta^{\star}_r = \sum_{i=-1}^{r} |b^{\star}_{i,r}|$. Comparer les valeurs des constantes de stabilités des méthodes de Admas-Bashforth et Adams-Moulton de même ordre, pour les ordre 1,2 et 10. **Attention il faut comparer les méthodes d'ordre égal, pas à nombre de pas égal.**
39. Quels sont les avantages et inconvénients des méthodes de Runge-Kutta, d'Adams-Bashforth et Adams-Mouton pour un même ordre ?

5 Méthodes de Prédiction Correction

Les méthodes de prédiction correction aussi appelées PECE (prédiction évaluation correction évaluation) couple une méthode explicite et une méthode implicte. D'abord on utilise une méthode explicite pour calculer y_{n+1} puis on évalue $\tilde{f}_{n+1} = f(t_{n+1}, y_{n+1})$ et on utilise cette valeur dans une méthode implicite.
On peut ainsi coupler une méthode AB avec une méthode AM. On peut montrer que le temps de calcul d'une méthode PECE est environ le double d'une méthode AB d'ordre égal mais avec une constante de stabilité bien meilleure. Le temps de calcul reste plus faible que pour les méthodes de RK pour des ordre supérieur à 3.

40. Expliciter le schéma PECE associé aux méthodes d'Euler et de AM_1.
41. Quel nom porte cette méthode ?
42. Expliciter le schéma PECE associé aux méthodes de Nyström et AM_2.
43. Tester et évaluer l'ordre de la méthode sur les EDO de référence.
44. Expliciter le schéma PECE associé aux méthodes AB_{r+1} et AM_{r+1}.
45. Ecrire une fonction matlab

    ```
    function [yn,tn,fn]=PECE1(r,ff,t0,y0,h,N))
    ```

    qui résout numériquement une EDO en utilisant la méthode PECE précédemement décrite. On utilisera la méthode RK4 pour l'initialisation.
46. Estimer numériquement l'ordre des méthodes ainsi obtenues sur les 3 exemples de référence.