Université Pierre et Marie Curie
Licence de Mathématiques LM383
Année universitaire 2007-2008
Thomas Capricelli, Driss Yakoubi

Équations différentielles

1 Mise en oeuvre de schémas de résolution numérique

On considère l'équation différentielle

$$\begin{cases} x' = \sin(tx), \\ x(0) = 1/10, \end{cases} \tag{1}$$

de solution unique[1] sur $[0, 1/2]$.

1. Résoudre (analytiquement) le problème de Cauchy approché

$$\begin{cases} y' = ty, \\ y(0) = 1/10, \end{cases} \tag{2}$$

2. Ecrire un script *Scilab* pour calculer la solution approchée de l'équation avec le schéma d'Euler Explicite. On comparera sur le même graphique la solution par Euler explicite des équations différentielles (1) et (2) ainsi que la solution exacte de la deuxième equation.

3. Vérifier la relation[1] d'approximation de (1) par (2) où $E(t) = y(t) - x(t)$ représente l'erreur.

$$\sup_{0 \le t \le 1/2} |E(t)| \le 5.10^{-6}. \tag{3}$$

4. Tester et comparer la solution numérique obtenue avec un schéma d'intégration plus précis. Runge Kutta d'ordre 2 par exemple.

5. *Jouer* sur les paramètres (dont le pas de temps) ainsi que sur le choix du schéma pour illustrer la convergence plus ou moins rapide.

2 Vérification de l'ordre de convergence

On veut maintenant mettre en évidence l'ordre de convergence d'un schéma numérique. Pour cela, on va se placer dans un cas où on connait la solution exacte de l'équation différentielle. Pour différentes valeurs du pas de discrétisation $\Delta t = t_i - t_{i-1}$, on calcule à un temps T fixé, la solution exacte U_e et approchée U_a, et on trace l'erreur - la valeur absolue de la différence entre ces deux valeurs - en fonction de Δt.

```
deff('y=fun(x,t)','y=2*t*sqrt(1.-x.^2)')
deff('y=exacte(t)','y=sin(t.^2)')
y0=0;
T=[0,1];
np=14;
h=zeros(np,1);
E=zeros(h);
nt=1;
for i=1:np
  nt=2*nt;
```

[1] fera l'objet d'un devoir

```
  h(i)=(T(2)-T(1))/nt;
  y=EulerExplicite(fun,y0,T,nt);
  E(i)=abs(y(2)-exacte(T(2)));
end
xset("window",2)
xbasc()
xtitle('verification de l''ordre de convergence du schema Euler explicite')
plot2d(h,E,2,leg='erreur euler explicite',strf='121',logflag="ll");
plot2d(h,h,[3,2],leg='ordre 1',strf='101',logflag="ll");
```

où `EulerExplicite(f,x0,T,n))` calcule une solution numérique `x` en différentes valeurs de t données dans `T` (`T(1)` étant le temps initial) de l'équation

$$\begin{cases} x'(t) = \texttt{fun}(x(t), t), \\ x(\texttt{T(1)}) = \texttt{x0}. \end{cases}$$

1. Quelle est l'équation différentielle résolue? Sur quel intervalle connait-on sa solution?
2. A quoi sert l'argument "logflag" dans l'appel à "plot2d"?
3. Que se passe-t-il si on prend des valeurs de T de plus en plus grand jusqu'à $\sqrt{\pi/2}$?
4. Adapter le script pour vérifier l'ordre numérique du schéma de Runge Kutta. Commenter la courbe obtenue.

3 Application à un problème vectoriel linéaire

L'évolution d'une population composée de moutons (m) et de loups (l) est décrite par le système d'équations différentielles

$$\begin{cases} m' = \frac{dm}{dt} = 4m - 2l, \\ l' = \frac{dl}{dt} = m + l, \end{cases}$$

avec la condition initiale m_0, l_0 fixée au temps $t = 0$ que l'on écrit sous la forme matricielle $p' = Ap$. En utilisant la forme diagonale de la matrice A, décrire l'évolution de la population en fonction du temps à partir d'un effectif donné. Calculer l'état de la population pour un temps infini. Discuter suivant l'état initial de la population.
On écrira un script *Scilab* qui calcule l'évolution de la population en utilisant d'une part la solution théorique et d'autre part un des deux schémas numériques vus précédemment. Représenter les deux solutions sur le même graphique.

Remarque . *La commande Scilab pour calculer une exponentielle de matrice est expm(A). La commande exp(A) calcule une matrice dont les composantes sont les exponentielles des composantes de A. On pourra vérifier dans Scilab qu'on a bien équivalence entre* `E=expm(A)` *et* `[D,M]=bdiag(A); E=M*exp(D)*inv(M)`

1. Que se passe-t-il si on démarre la simulation avec la proportion deux loups pour un mouton? Jusqu'à quel temps le modèle est-il valable?
2. Même question avec la condition initiale un loup pour un mouton.
3. Comparer sur le même graphique la solution exacte et approchée par le schéma d'Euler explicite.
4. Faire varier la précision du schéma d'Euler en jouant sur les paramètres np et nt. Comparer avec les performances de Runge Kutta d'ordre 4.

4 Le modèle prédateur-proie

On va maintenant étudier un modèle plus sophistiqué de l'évolution des populations loups/moutons, basé sur les équations de "Volterra" et qui s'écrit sous la forme:

$$\begin{cases} m' = \frac{dm}{dt} = k_1 m - Cml, \\ l' = \frac{dl}{dt} = Dml - k_2 l, \end{cases} \tag{4}$$

avec toujours la condition initiale $m(0) = m_0$ et $l(0) = l_0$

1. Quelle est la signification "physique" des constantes k_1, k_2, C et D?
2. Adapter le programme *Scilab* de la précédente question à ce cas non linéaire.
3. Faire tourner la simulation avec les valeurs des paramètres suivants

   $$k_1 = 2 \quad k_2 = 10 \quad C = 0.001 \quad D = 0.002,$$

   la condition initiale

   $$(m(0), l(0)) = (5000, 100),$$

   et un temps final de $T = 10$.

   Pour chacune des questions suivantes, notez par écrit vos observations:
4. Ajuster les paramètres np et nt de manière à avoir une bonne représentation de la solution.
5. Représenter dans une autre fenêtre l'évolution de la population des loups en fonction de celle des moutons.
6. Utiliser maintenant le schéma de Runge Kutta d'ordre 4.
7. Toujours avec ce schéma, changez les conditions initiales