

MT12 - TP6 : Résolution approchée d'équations différentielles

Il n'est pas toujours possible de calculer explicitement la solution d'une équation différentielle. Le but de ce TP est de présenter différentes méthodes numériques permettant d'approcher la solution de ces équations différentielles.

1 Introduction

Soit $f : y \mapsto f(y)$ une fonction définie de \mathbb{R}^n ($n \geq 1$) dans \mathbb{R}^n . Soit $T > 0$ et $y_0 \in \mathbb{R}^n$, on considère le système d'équations différentielles ordinaires (EDO) :

$$\begin{cases} y'(t) = f(y(t)) \text{ pour } t \in [0, T], \\ y(0) = y_0. \end{cases} \quad (1)$$

Par exemple si $n = 2$ avec $y(t) = (y_1(t), y_2(t))^T$ et $f(y(t)) = (y_1(t), -y_2(t))^T$, alors dans ce cas le système (1) s'écrit

$$\begin{cases} y_1'(t) = y_1(t) \text{ pour } t \in [0, T], \\ y_2'(t) = -y_2(t) \text{ pour } t \in [0, T], \end{cases}$$

avec $y(0) = y_0 = (y_1(0), y_2(0))^T$ une donnée initiale.

Dans ce TP nous allons considérer différents schémas numériques pour le problème (1). Pour cela, on se donne une subdivision régulière de $[0, T]$: $t_0 = 0 < t_1 < \dots < t_{N-1} < t_N = T$ de pas $\Delta t = T/N$ et on cherche à calculer une solution approchée $(Y^k)_{1 \leq k \leq N}$ (où $Y^k \in \mathbb{R}^n$) du problème (1) où $Y^k \approx y(t_k)$ pour $k = 1, \dots, N$. Voici trois méthodes différentes permettant de calculer les vecteurs $(Y^k)_{1 \leq k \leq N}$:

(a) **Méthode d'Euler explicite** :

$$Y^{k+1} = Y^k + \Delta t f(Y^k), \quad k = 0, \dots, N-1.$$

(b) **Méthode de Heun** (Runge-Kutta d'ordre 2) :

$$Y^{k+1} = Y^k + \frac{\Delta t}{2} \left(f(Y^k) + f(Y^k + \Delta t f(Y^k)) \right), \quad k = 0, \dots, N-1.$$

(c) **Méthode RK4 « classique »** : pour $k = 0, \dots, N-1$,

$$\begin{cases} k_1 = f(Y^k), \\ k_2 = f(Y^k + \frac{\Delta t}{2} k_1), \\ k_3 = f(Y^k + \frac{\Delta t}{2} k_2), \\ k_4 = f(Y^k + \Delta t k_3), \\ Y^{k+1} = Y^k + \frac{\Delta t}{6} (k_1 + 2k_2 + 2k_3 + k_4). \end{cases}$$

Dans tous les cas, on prend pour Y^0 la condition initiale y_0 .

2 Résolution numérique d'EDO

Afin de tester ces méthodes numériques nous allons étudier différentes EDO pour lesquels il est possible de calculer la solution exacte.

2.1 Programmation des méthodes

1. Calculer la solution exacte y de (1) dans le cas suivant :

$$f(y) = -50y, \quad y_0 = 10.$$

2. Définir une fonction scilab `sol` donnant l'expression de la solution exacte de (1). Écrire ensuite trois fonctions `EulerExpl`, `Heun` et `RK4` qui, partant de Y_0 , T et N , calculent les solutions approchées $(Y^k)_{0 \leq k \leq N}$ pour chacune des trois méthodes. Par exemple, dans la cas de la fonction `EulerExpl` aidez-vous du script suivant :

```
1 function y=f(x)
2     y=-50*x
3 endfunction
4
5 function Y=EulerExpl(Y0,T,N,f)
6     dt=T/N;
7     Y=zeros(length(Y0),N)
8     Y(:,1)=...; // remplir
9     for i=1:N-1
10        Y(:,i+1)=...; // remplir
11    end
12 endfunction
```

Pour $T = 1$ et $N = 50$ et en utilisant la fonction `subplot` de `Scilab`, tracer sur une même figure trois graphiques représentant la solution approchée obtenue par les trois méthodes et la solution exacte (comme sur la Figure 1).

3. Quelle méthode semble la plus efficace ?

2.2 Le cas d'un système de deux EDO découplées

On considère à présent le système d'EDO suivant

$$\begin{cases} y_1'(t) = -50y_1(t), & \text{pour } t \in [0, T], \\ y_2'(t) = 3y_2(t), & \text{pour } t \in [0, T], \\ y(0) = (10, 2)^\top. \end{cases} \quad (2)$$

Le système d'EDO (2) est dit découplé car on peut résoudre de manière indépendante l'équation vérifiée par y_1 et celle vérifiée par y_2 .

4. Montrer que le système (2) peut se mettre sous la forme (1) avec f définie de \mathbb{R}^2 dans \mathbb{R}^2 . Donner l'expression de la solution exacte du système (2).
5. Pour $T = 0.5$ et $N = 50$ et en utilisant la fonction `subplot` de `Scilab`, tracer sur une même figure trois graphiques représentant la solution approchée obtenue par la méthode (a), (b) et (c) et la solution exacte (comme dans le cas de la Figure 1).

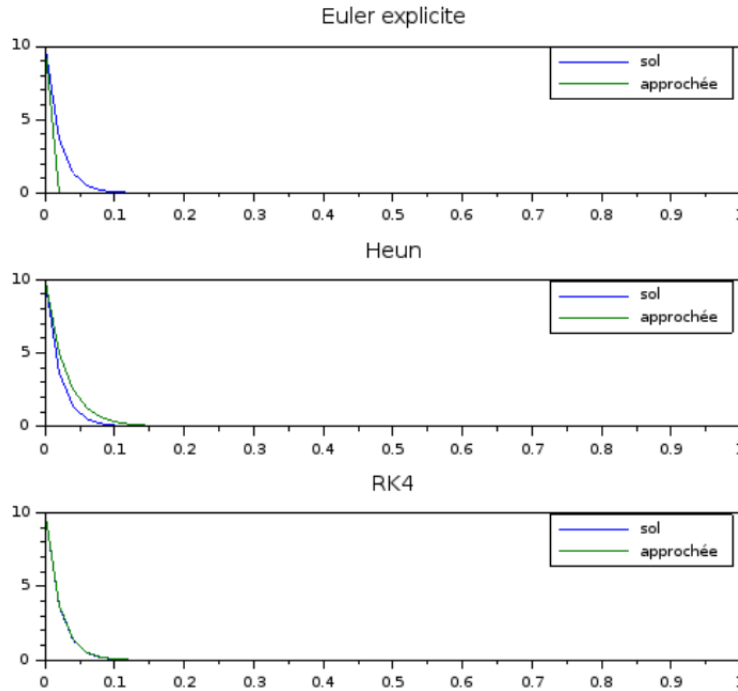


FIGURE 1 – Illustration des méthodes numériques (a), (b) et (c).

2.3 Le modèle proies-prédateurs, un système non linéaire

Dans le cas des EDO considérés jusqu'à présent l'utilisation d'une méthode numérique n'est pas vraiment utile car il est possible de calculer « à la main » la solution exacte. Les exemples précédents permettent juste d'illustrer les méthodes numériques et fournissent des cas tests utiles. Considérons à présent un système d'EDO où la solution exacte n'est pas calculable à la main.

On veut calculer les solutions approchées du système différentiel d'ordre 1 non linéaire :

$$\begin{cases} x_1'(t) = 3x_1(t) - x_1(t)x_2(t), \\ x_2'(t) = -2x_2(t) + x_1(t)x_2(t), \end{cases} \quad (3)$$

Ici x_1 représente le nombre de proies (par exemple des lapins) et x_2 le nombre de prédateurs (par exemple des renards). Le modèle (3) est un modèle de type proies-prédateurs, appelé aussi modèle de Lotka-Volterra. Ce type de système est utilisé pour décrire la dynamique de systèmes biologiques dans lesquels une proie et son prédateur interagissent.

On peut interpréter les équations du système (3) de la manière suivante : pour la première équation en absence de prédateurs ($x_2(t) = 0$) alors les lapins se reproduisent avec un taux d'accroissement $a = 3$. Cependant les lapins sont mangés proportionnellement à la présence de prédateurs, phénomène modélisé par le terme $-x_2(t)$. De la même manière, pour la deuxième équation, en l'absence de lapins ($x_1(t) = 0$) les prédateurs ne peuvent se nourrir et meurent, phénomène représenté par le terme $-2x_2(t)$ où 2 représente le taux de mortalité. Par contre, en présence de proies, la population de renards s'accroît proportionnellement, ce qui explique le terme $x_1(t)$. Schématiquement, on peut résumer la situation sous

la forme :

$$\begin{aligned}
 x_1'(t) &= \underbrace{3x_1(t)}_{\text{croissance}} + \underbrace{(-x_1(t)x_2(t))}_{\text{proies consommées}} , \\
 x_2'(t) &= \underbrace{-2x_2(t)}_{\text{mortalité prédateurs}} + \underbrace{x_1(t)x_2(t)}_{\text{croissance}} .
 \end{aligned}$$

6. Montrer que le système (3) peut se mettre sous la forme (1) avec f une fonction non linéaire définie de \mathbb{R}^2 dans \mathbb{R}^2 . Définir en Scilab la fonction f en vous aidant du script suivant :

```

1 function ans=f(X)
2     ans=[3*X(1)...;...]
3 endfunction

```

7. Pour $N = 5000$, calculer la solution approchée du système (3) pour la méthode d'Euler explicite, de Heun et RK4 sur l'intervalle $[0, 20]$ avec $x_1(0) = 1$ et $x_2(0) = 2$. Représenter graphiquement $x_2(t)$ en fonction de $x_1(t)$, pour ce faire on utilisera la fonction `comet` de Scilab.
8. Commenter les résultats obtenus.

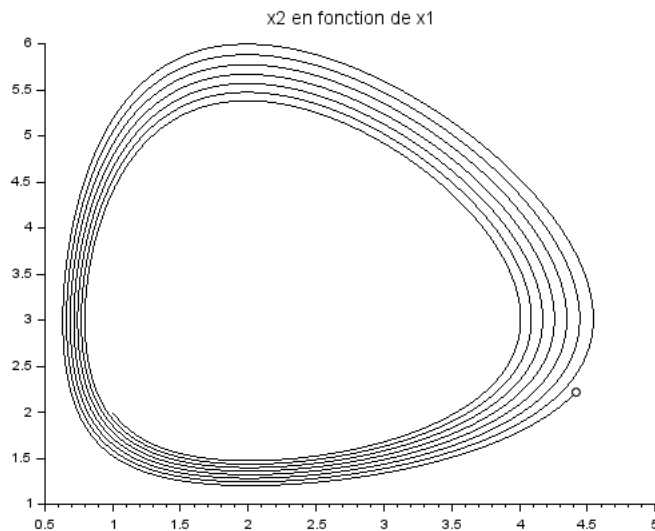


FIGURE 2 – $x_2(t)$ en fonction de $x_1(t)$ par la méthode d'Euler explicite.

3 Résolution numérique de l'équation de la chaleur sur domaine non borné

On considère à présent l'équation de la chaleur sur domaine non borné, i.e. sur \mathbb{R}

$$\begin{cases}
 \partial_t u(x, t) = a \partial_x^2 u(x, t) & x \in \mathbb{R}, t \in (0, T], \\
 u(x, 0) = u^0(x) & x \in \mathbb{R},
 \end{cases} \quad (4)$$

où u représente la densité de chaleur dans une barre de longueur infinie et $a > 0$ est la constante de diffusion de la chaleur. L'équation (4) est une équation aux dérivées partielles (EDP), ici la solution de cette équation est une fonction u dépendant du temps et de l'espace. Ainsi pour approcher la solution de (4) il faut une méthode numérique en espace et en temps. Bien entendu la résolution même numérique d'une EDP est plus complexe que la résolution d'une EDO. Cependant dans le cas de l'équation (4) et en utilisant la transformée de Fourier on se ramène à résoudre numériquement un système d'équations différentielles découplées.

Expliquons dans un premier temps l'approche générale utilisée dans la suite. L'objectif principal est d'obtenir une approximation de u solution de (4) aux instants t_1, t_2, \dots, t_N et au points $x_{-M/2} < \dots < 0 < \dots < x_{M/2-1}$ (M un entier naturel pair), i.e. des approximation de la valeur de

$$u(x_i, t_k) \quad \forall (i, k) \in \{-M/2, \dots, M/2 - 1\} \times \{1, \dots, N\}.$$

La résolution numérique de (4) sur tout l'espace \mathbb{R} est impossible ce qui nous pousse à considérer la suite de points $x_{-M/2}, \dots, x_{M/2-1}$.

Pour ce faire, on va tout d'abord appliquer la transformée de Fourier en espace à (4) pour en déduire une équation différentielle vérifiée par $\widehat{u}(t, \xi)$. Comme l'équation (4) est posée sur tout \mathbb{R} , alors l'EDO sur $\widehat{u}(t, \xi)$ doit être vérifiée pour tout $\xi \in \mathbb{R}$. Au niveau numérique on se restreint à résoudre cette EDO sur un carré (dans le domaine fréquentiel) centré en 0 et de taille M . On considère alors une discrétisation uniforme de ce carré, i.e., on considère une suite $\xi_{-M/2} < \dots < \xi_{M/2-1}$, ce qui conduit alors à un système de M EDO découplées portant sur $\widehat{u}(t, \xi_{-M/2}), \dots, \widehat{u}(t, \xi_{M/2-1})$. On applique alors la méthode (c) pour approcher les solutions de ces M EDO et enfin on applique la transformée de Fourier inverse pour en déduire la solution approchée u de (4) aux instants t_1, t_2, \dots, t_N et au points $x_{-M/2}, \dots, x_{M/2-1}$.

Comme annoncé précédemment pour la variable d'espace « x » on va utiliser la transformée de Fourier de u en **espace**. Formellement on a

$$\widehat{\partial_t u}(\xi, t) = \int_{\mathbb{R}} \partial_t u(x, t) e^{-2i\pi\xi x} dx = \partial_t \widehat{u}(\xi, t).$$

De même en utilisant des intégrations par parties comme dans le cours on obtient

$$\widehat{\partial_x u}(\xi, t) = (2i\pi\xi) \widehat{u}(\xi, t) \quad \text{et} \quad \widehat{\partial_x^2 u}(\xi, t) = (-4\pi^2\xi^2) \widehat{u}(\xi, t).$$

Ainsi en appliquant la transformée de Fourier à l'équation de la chaleur on se ramène à devoir résoudre l'EDO (on a « perdu » la dérivée en espace) suivante :

$$\partial_t \widehat{u}(\xi, t) = a(-4\pi^2\xi^2) \widehat{u}(\xi, t) \quad \xi \in \mathbb{R}, t \in (0, T]. \quad (5)$$

Encore une fois, sur ordinateur on ne peut résoudre numériquement l'équation précédente pour tout $\xi \in \mathbb{R}$. À la place on considère une suite de fréquences $-\xi_{M/2} < \dots < 0 < \dots < \xi_{M/2-1}$ (avec $M \geq 1$ un entier pair). Dans la suite on va noter $\widehat{u}_i(t) = \widehat{u}(\xi_i, t)$ et on va vouloir résoudre de manière approchée le système d'EDO suivant

$$\begin{cases} \partial_t \widehat{u}_i(t) &= (-4a\pi^2\xi_i^2) \widehat{u}_i(t) \quad \forall i \in \{-M/2, \dots, 0, \dots, M/2 - 1\}, t \in (0, T], \\ \widehat{u}_i(\xi_i, 0) &= \widehat{u^0}(\xi_i) \quad \forall i \in \{-M/2, \dots, 0, \dots, M/2 - 1\}. \end{cases} \quad (6)$$

9. Le système (6) est constitué de M EDO découplées. Montrer que le système (6) peut se mettre sous la forme (1) avec f définie de \mathbb{R}^M dans \mathbb{R}^M .
10. L'idée dans la suite est de résoudre de manière approchée le système d'EDO (6) via la méthode (c) puis d'appliquer la transformée de Fourier discrète inverse afin de revenir dans le domaine spatial pour obtenir une approximation de la fonction u solution de (4). Pour ce faire on commence par discrétiser le domaine spatial, puis le domaine fréquentiel associé et enfin le domaine temporel en utilisant les commandes suivantes :

```

1 // discretisation domaine spatial
2
3 L=100; // on considere le carre [-50,50]
4 M=100;
5
6 dx=L/M; // pas d'espace
7 x=-L/2:dx:L/2-dx;
8
9 // discretisation domaine frequentiel
10
11 xi= (1/L)*[-M/2:M/2-1]';
12 xi=fftshift(xi); // utile pour la suite
13
14 // discretisation domaine temporel
15
16 T=20;
17 dt=1e-3; // pas de temps
18
19 t=0:dt:T;
20 N=length(t);

```

Écrire une fonction `f` en Scilab prenant en entrée un vecteur v (à M lignes et une colonne) et le vecteur xi (introduit dans les lignes précédentes) et renvoyant en sortie la valeur du vecteur $f(v)$ avec

$$v = (v_{-M/2}, \dots, v_{M/2-1}(t))^T,$$

ici f renvoie à la fonction obtenue à la question 9. Enfin dans la suite on considère la donnée initiale

$$u^0(x) = \mathbf{1}_{[-10,10]}(x) \quad x \in \mathbb{R}.$$

Compléter les commandes suivantes afin d'implémenter u^0 en Scilab :

```

1 u0=zeros(1,length(x));
2 u0((L/2-L/10)/dx:(L/2+L/10)/dx)=...;
3 u0=u0';

```

11. Pour une constante de diffusion $a = 1$ et en utilisant la fonction `RK4` (avec pour donnée initiale `fft(u0)`) calculer la matrice `uhat` à M lignes et N colonnes où par exemple la colonne j de `uhat` donne une solution approchée de (6) (via la méthode (c)) au temps t_j , i.e., une valeur approchée de

$$\hat{u}(t_j) = (\hat{u}_{-M/2}(t_j), \dots, \hat{u}_{M/2-1}(t_j))^T.$$

12. Afin de revenir dans le domaine spatial compléter le script suivant :

```

1 for k=1:...
2     u(:,k)=ifft(...(...));
3 end

```

Enfin tracer sur une même figure le profil de u pour $t = 0, 2, 10$ et 20 . Que constatez-vous sur la régularité de la solution au cours du temps ?