

TP de Méthodes Numériques :

Simulation d'une chaîne d'oscillateurs non linéaires

On considère un système de n oscillateurs couplés, régi par une équation différentielle du second ordre dans \mathbb{R}^n :

$$\frac{d^2 u}{dt^2} + C \frac{du}{dt} = f(u) \quad (1)$$

avec $u(t) = (u_1(t), u_2(t), \dots, u_n(t))^T \in \mathbb{R}^n$, $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ fonction de classe C^2 et

$C \in M_n(\mathbb{R})$ symétrique définie positive.

Nous allons simuler numériquement des équations de la forme (1) à l'aide du schéma aux différences finies centré implicite

$$\frac{u^{(k+1)} - 2u^{(k)} + u^{(k-1)}}{h^2} + C \frac{u^{(k+1)} - u^{(k-1)}}{2h} = f(u^{(k)}) \quad (2)$$

où $h > 0$ est le pas de discrétisation et $u^{(k)}$ une approximation de $u(kh)$.

En introduisant la variable auxiliaire

$$v = \frac{du}{dt} + C u,$$

l'équation différentielle du second ordre (1) est équivalente à celle du premier ordre dans $\mathbb{R}^n \times \mathbb{R}^n$

$$\begin{cases} u' &= v - C u, \\ v' &= f(u), \end{cases} \quad (3)$$

pour laquelle nous donnerons une reformulation du schéma (2).

Les schémas numériques mis en oeuvre nous permettront d'étudier la convergence de différents systèmes vers un état d'équilibre et certains phénomènes transitoires. Une solution d'équilibre $u^* \in \mathbb{R}^n$ de (1) est une solution indépendante de t , caractérisée par la propriété $f(u^*) = 0$. De manière équivalente, une solution d'équilibre $(u^*, v^*) \in \mathbb{R}^n \times \mathbb{R}^n$ de (3) est caractérisée par

$$f(u^*) = 0, \quad v^* = C u^*.$$

1. Reformulation et étude du schéma centré

Etant donné une condition initiale

$$(u(0), v(0)) = (u^{(0)}, v^{(0)}),$$

on souhaite calculer numériquement $(u(t), v(t))$ pour $t \in [0, T]$. On approche $(u(kh), v(kh))$ par la solution $(u^{(k)}, v^{(k)})$ du schéma suivant

$$\forall k \geq 0, \quad \begin{cases} v^{(k+1/2)} &= v^{(k)} + \frac{h}{2} f(u^{(k)}) & (a) \\ u^{(k+1)} &= u^{(k)} + h v^{(k+1/2)} - h C \frac{u^{(k+1)} + u^{(k)}}{2} & (b) \\ v^{(k+1)} &= v^{(k+1/2)} + \frac{h}{2} f(u^{(k+1)}). & (c) \end{cases} \quad (4)$$

A chaque itération, il est nécessaire de résoudre l'équation (4)-(b) pour $u^{(k+1)}$, qui correspond au système linéaire

$$A u^{(k+1)} = \left(I - \frac{h}{2} C\right) u^{(k)} + h v^{(k+1/2)}$$

de matrice symétrique définie positive

$$A = I + \frac{h}{2} C. \quad (5)$$

Question 1

Montrer que la condition initiale et la relation de récurrence (4) déterminent la suite $(u^{(k)}, v^{(k)})_{k \geq 0}$.

Réponse :

On notera $\varphi : (\mathbb{R}^n)^2 \rightarrow (\mathbb{R}^n)^2$ l'application définie par (4)

$$(u^{(k)}, v^{(k)}) \mapsto \varphi(u^{(k)}, v^{(k)}) = (u^{(k+1)}, v^{(k+1)}).$$

Question 2

Montrer que toute solution de (4) vérifie (2) pour tout $k \geq 1$.

Réponse :

Question 3

Vérifier que les points fixes de φ sont les équilibres (u^*, v^*) de (3).

Réponse :

Question 4

Déterminer l'ordre de consistance du schéma (4).

Réponse :

Question 5

On admet que lorsque f est lipschitzienne sur \mathbb{R}^n , le schéma (4) est stable par rapport aux erreurs sur l'intervalle $[0, T]$. Justifier dans ce cas la convergence du schéma (4) lorsque $h \rightarrow 0$ et donner son ordre de convergence.

Réponse :

2. Test du schéma pour un oscillateur linéaire

On considère l'équation (1) pour $u(t) \in \mathbb{R}$, $C > 0$ et

$$f(u) = 1 - u.$$

Ce système décrit par exemple le déplacement d'un ressort linéaire amorti soumis à une force constante unitaire. Les solutions $u(t)$ de (1) peuvent être calculées explicitement dans ce cas (l'équation est linéaire non homogène à coefficients constants). Ce calcul permet de tester le schéma (4) avant de l'appliquer aux exemples plus complexes étudiés dans les sections 3 et 4.

Question 6

Calculer la solution d'équilibre (u^*, v^*) de (3).

Réponse :

```
In [ ]: %matplotlib inline
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation
import numpy as np
```

Question 7

Ecrire une fonction **iter** qui calcule la solution $[u^{(k+1)}, v^{(k+1)}]$ de (4) en fonction des arguments $[u^{(k)}, v^{(k)}]$ et h . Pour des valeurs de votre choix de C et h , tester cette fonction sur le vecteur $[u^{(k)}, v^{(k)}] = [u^*, v^*]$.

```
In [ ]:
```

Question 8

Ecrire une fonction **solnum** (prenant en arguments la condition initiale $[u(0), v(0)]$, le temps maximal d'intégration T et le pas h) qui calcule la solution numérique $[u^{(k)}, v^{(k)}]$ aux temps $t_k = k h \in [0, T]$, c'est à dire jusqu'à l'itéré $k_{\max} = E(T/h)$ où E désigne la partie entière. La fonction fournit en sortie les valeurs de $[u^{(k)}, v^{(k)}]$ pour $0 \leq k \leq k_{\max}$ sous la forme d'une matrice de taille $2 \times (k_{\max} + 1)$. Pour des valeurs de votre choix de C , T et h , tester cette fonction sur la condition initiale $[u(0), v(0)] = [u^*, v^*]$.

```
In [ ]:
```

Dans les questions 9 à 11, on fixe $C = 0.3$ et on suppose le système initialement au repos :

$$u(0) = 0, \quad u'(0) = 0, \quad (6)$$

et donc $v(0) = 0$.

Question 9

Calculer la solution explicite $(u(t), v(t))$ de (3) pour cette condition initiale. Que vaut $\lim_{t \rightarrow +\infty} (u(t), v(t))$? Ecrire une fonction **solexacte** prenant en argument un vecteur de temps $[t_k]_{0 \leq k \leq k_{\max}}$, qui évalue cette solution aux temps t_k en stockant les valeurs dans une matrice de taille $2 \times (k_{\max} + 1)$.

Réponse (calcul explicite) :

```
In [ ]:
```

Question 10

On fixe $T = 25$. Tracer le graphe de l'erreur globale

$$\text{Err}(h) = \max_{0 \leq k \leq k_{\max}} \|(u^{(k)}, v^{(k)}) - (u(t_k), v(t_k))\|_2$$

pour $h \in [10^{-5}, 10^{-1}]$ en utilisant une échelle logarithmique. Retrouvez-vous l'ordre de convergence obtenu dans la question 5 ?

```
In [ ]:
```

Question 11

Etudier numériquement pour quelles valeurs de $h > 0$ on a

$\lim_{k \rightarrow +\infty} (u^{(k)}, v^{(k)}) = \lim_{t \rightarrow +\infty} (u(t), v(t))$. Pour cela, on pourra afficher pour différentes valeurs de h , et sur des intervalles de temps $[0, T]$ bien choisis, les graphes de $u^{(k)}$ et $v^{(k)}$ en fonction de t_k .

In []:

3. Oscillateur non linéaire soumis à une force constante

On considère l'équation (1) pour $u(t) \in \mathbb{R}$, $C > 0$ et

$$f(u) = 1 - u^3.$$

Ce système décrit le déplacement d'un ressort non-linéaire (non-linéarité cubique) avec amortissement linéaire, sous l'action d'une force constante unitaire.

Question 12

Calculer la solution d'équilibre (u^*, v^*) de (3).

Réponse :

Question 13

Pour un oscillateur initialement au repos (condition initiale (6)), on calcule numériquement la solution de (3) sur l'intervalle de temps $[0, T]$ en utilisant le schéma (4). On choisit $T = 30$ et le pas de temps $h = 10^{-3}$. Pour différentes valeurs de C , tracer les graphes de $u^{(k)}$ et $v^{(k)}$ en fonction de t_k . Etudier numériquement pour quelles valeurs de C la convergence vers l'équilibre s'effectue de manière monotone (régime sur-amorti), et pour quelles valeurs du paramètre cette convergence est oscillante (régime sous-amorti).

In []:

4. Chaîne d'oscillateurs non linéaires forcée à une extrémité

On étudie l'équation (1) avec $C = \mu K$ où $\mu > 0$ est un paramètre et la matrice $K \in M_n(\mathbb{R})$ ($n \geq 2$) s'écrit:

$$K = \begin{pmatrix} 2 & -1 & 0 & 0 & \dots & 0 \\ -1 & 2 & -1 & 0 & & \vdots \\ 0 & -1 & 2 & -1 & \ddots & \vdots \\ \vdots & 0 & -1 & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & 2 & -1 \\ 0 & \dots & \dots & 0 & -1 & 1 \end{pmatrix}.$$

On fixe par ailleurs

$$f(u) = -K u^3 + e_1 \quad (7)$$

où $e_1 = (1, 0, 0, \dots, 0)^T$ est le premier vecteur de la base canonique de \mathbb{R}^n et on note

$$u^3 = (u_1^3, u_2^3, \dots, u_n^3)^T.$$

Ce système décrit une chaîne de n oscillateurs non linéaires cubiques (du même type que dans la section 3) couplés à leurs proches voisins. Une extrémité de la chaîne est soumise à une force constante unitaire, l'autre extrémité étant fixe.

Question 14

Montrer que la matrice K est symétrique définie positive.

Réponse :

Question 15

Montrer que l'équation (3) admet une unique solution d'équilibre $(u^*, v^*) \in \mathbb{R}^n \times \mathbb{R}^n$ qu'on calculera explicitement.

Réponse :

On veut approcher numériquement la solution $(u(t), v(t))$ de (3) pour une condition initiale $(u(0), v(0))$ donnée et $t \in [0, T]$ en utilisant le schéma (4).

Question 16

Définir une fonction **fvec** qui à un vecteur u associe le vecteur $f(u)$ défini par (7). Tester cette fonction sur le vecteur u^* .

In []:

Question 17

Ecrire une fonction **factorise** (prenant en arguments n , μ et h) qui calcule la factorisation $A = LU = L D L^t$ de la matrice tridiagonale symétrique définie positive A définie par (5). On rappelle que L et D sont de la forme

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ l_0 & 1 & 0 & 0 & & \vdots \\ 0 & l_1 & 1 & 0 & \ddots & \vdots \\ \vdots & 0 & l_2 & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & \dots & 0 & l_{n-2} & 1 \end{pmatrix}, \quad D = \begin{pmatrix} d_0 & 0 & 0 & 0 & \dots & 0 \\ 0 & d_1 & 0 & 0 & & \vdots \\ 0 & 0 & d_2 & 0 & \ddots & \vdots \\ \vdots & 0 & 0 & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & \dots & 0 & 0 & d_{n-1} \end{pmatrix}$$

avec $d_i > 0$. La fonction fournit en sortie deux vecteurs contenant les coefficients sous-diagonaux de L et diagonaux de D . Tester cette fonction pour des valeurs de votre choix de n , μ et h .

In []:

Question 18

Ecrire une fonction **solchaine** (prenant en arguments la condition initiale $(u(0), v(0))$, le temps maximal d'intégration T et le pas h) qui calcule la solution numérique $(u^{(k)}, v^{(k)})$ aux temps $t_k = k h \in [0, T]$. La fonction fournit en sortie les valeurs de $u^{(k)}, v^{(k)}$ pour $0 \leq k \leq k_{\max}$ sous la forme de deux matrices de taille $n \times (k_{\max} + 1)$

$$[u^{(0)}, u^{(1)}, \dots, u^{(k_{\max})}], \quad [v^{(0)}, v^{(1)}, \dots, v^{(k_{\max})}].$$

La factorisation $A = L D L^t$ est calculée avant de démarrer les itérations (4), puis à chaque itération on résout l'équation (4)-(b) par simple descente/remontée. Tester cette fonction sur la condition initiale $[u(0), v(0)] = [u^*, v^*]$ pour des valeurs de votre choix de n, μ, T et h .

In []:

Dans les questions 19 et 20, on simule le système (1) pour la condition initiale (6) correspondant à la chaîne au repos. On note $u_{\text{app}}(j, t_k)$ l'approximation numérique de la solution $u_j(t_k)$ de (1)-(6), et $u'_{\text{app}}(j, t_k)$ celle de $u'_j(t_k)$. On a donc

$$u'_{\text{app}}(\cdot, t_k) = v^{(k)} - \mu K u^{(k)}.$$

Question 19

On choisit comme pas d'intégration $h = 10^{-3}$. Pour $n = 5$ et $\mu = 0.5$, calculer la solution numérique de (1)-(6) sur un intervalle $[0, T]$ bien choisi et tracer les graphes de $u_{\text{app}}(j, t)$ et $u'_{\text{app}}(j, t)$ en fonction de t pour $j = 1, 3, 5$ (i.e. pour les oscillateurs situés en début, milieu et fin de chaîne). Quel est l'ordre de grandeur du temps t_{eq} nécessaire pour atteindre un état proche de l'équilibre ? On pourra fixer comme critère $\|u(t) - u^*\|_{\infty} < 10^{-3}$ pour tout $t \geq t_{\text{eq}}$. Comparer ce temps à celui observé pour $n = 1$ (modèle étudié dans la section 3 avec $C = 0.5$).

In []:

Question 20

Calculer u_{app} pour $n = 100$ et $\mu = 5$, en fixant $T = 2500$ et $h = 10^{-2}$. Décrire les différents phénomènes qu'on peut observer avant que le système atteigne un état voisin de l'équilibre. Pour étayer votre analyse, afficher un échantillon de graphes bien choisis :

- graphes des fonctions $t \mapsto u_{\text{app}}(j, t)$ à différents noeuds j ,
- graphe de $j \mapsto u_{\text{app}}(j, t)$ à différents temps t ,
- graphe animé de $j \mapsto u_{\text{app}}(j, t)$ lorsque t décrit l'intervalle $[0, T]$, en utilisant la fonction **FuncAnimation()** du module animation de Matplotlib.

In []:

```
# Enable interactive plot
%matplotlib notebook
```