TRAVAUX PRATIQUES - MÉTHODES DES DIFFÉRENCES FINIES

TP 3 : EQUATION DE LA CHALEUR 1D INSTATIONNAIRE (VERSION PROVISOIRE)

1 Conditions de Dirichlet

On souhaite résoudre numériquement l'E.D.P. suivante

$$\frac{\partial u}{\partial t}(t,x) - \nu \frac{\partial^2 u}{\partial x^2}(t,x) = f(t,x),\ \forall (t,x) \in]t_0; t_0+T] \times]a;b[, \tag{1.1}$$

$$u(t_0, x) = u_0(x),\ \forall x \in [a;b], \tag{1.2}$$

$$u(t,a) = u_a(t),\ \forall t \in [t_0; t_0+T], \tag{1.3}$$

$$u(t,b) = u_b(t),\ \forall t \in [t_0; t_0+T]. \tag{1.4}$$

avec ν un réel strictement positif, $t_0 \in \mathbb{R}$, $T > 0$, $(a,b) \in \mathbb{R}^2$, $a < b$.

On note t^n, $n \in [\![0, N_t]\!]$ et x_i, $i \in [\![0, N_x]\!]$ les discrétisations régulières des intervalles $[t_0; t_0+T]$ et $[a;b]$ avec N_t pas de discrétisation en temps et N_x pas de discrétisation en espace.

On souhaite implémenter deux schémas de résolution de cette E.D.P. :

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} - \nu \frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{\Delta x^2} = f_i^{n+1}. \tag{1.5}$$

et

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} - \nu \frac{u_{i+1}^{n} - 2u_i^{n} + u_{i-1}^{n}}{\Delta x^2} = f_i^{n}. \tag{1.6}$$

où $\Delta t = T/N_t$, $\Delta x = (b-a)/N_x$, $f_i^n = f(t^n, x_i)$ et $u_i^n \approx u(t^n, x_i)$.

On rappelle que le premier schéma est le **schéma d'Euler implicite** et le second le **schéma d'Euler explicite**. Le schéma d'Euler implicite est inconditionnelement stable et le schéma d'Euler explicite est stable sous la condition de C.F.L.

$$\nu \frac{\Delta t}{\Delta x^2} \leqslant \frac{1}{2}.$$

On note, $\forall n \in [\![0, N_t]\!]$, $\boldsymbol{U}^n$ les vecteurs de dimension $N_x + 1$, de composantes $\boldsymbol{U}_i^n = u_{i-1}^n$, $\forall i \in [\![1, N_x+1]\!]$.

Q. 1 *Pour chaque schéma, écrire sur feuille et* ***de manière détaillée*** *la discrétisation de l'E.D.P. (1.1) à (1.4)* ■

On étudie cette E.D.P. avec les données $t_0 = 0$, $T = 2$, $a = 0$, $b = 2\pi$, $\nu = 2$, $k = 5$

$$\begin{aligned} f(t,x) &= -k\sin(kt)\cos(x) + \nu\cos(kt)\cos(x), \\ u_0(x) &= \cos(kt_0)\cos(x), \\ u_a(t) &= \cos(kt)\cos(a), \\ u_b(t) &= \cos(kt)\cos(b). \end{aligned}$$

Dans ce cas, la solution exacte est donnée par $u_{\text{ex}}(t,x) = \cos(kt)\cos(x)$.

Q. 2 1. *Pour résoudre l'E.D.P. par un schéma d'**Euler implicite**, le programme `mainChaleurImplicite.m` (script Matlab) est fourni, ainsi que les fonctions `CalculF.m`, `NormInf.m` et `PlotSol.m` (ou `PlotSol2.m`) (voir fichier `TP3.tar.gz`). Dans les codes fournis, il manque le fichier `EulerImplicite.m` correspondant à la fonction :*

```
[t,x,u]=EulerImplicite(EDP,Nt,Nx)
```

résolvant l'E.D.P. par un schéma d'Euler implicite avec

- *EDP : structure, définie dans* `mainChaleurImplicite.m`*, contenant l'ensemble des données de l'E.D.P. à résoudre.*
- `Nt` *: nombre de pas de discrétisation en temps,*
- `Nx` *: nombre de pas de discrétisation en espace,*
- `t` *: discrétisation en temps (dimension* `Nt+1`*),* $\mathtt{t}(n) = t^{n-1}$, $\forall n \in [\![1, N_t + 1]\!]$,
- `x` *: discrétisation en espace (dimension* `Nx+1`*),* $\mathtt{x}(i) = x_{i-1}$, $\forall i \in [\![1, N_x + 1]\!]$,
- `u` *: u(i,n) solution approchée au temps t(n) et point x(i) (dimension (*`Nx+1,Nt+1`*)).*

Ecrire cette fonction.

2. *Utiliser le **Profiler** de Matlab pour comparer les temps d'execution de la fonction* `EulerImplicite` *dans les cas ou la matrice est stockée de manière pleine ou creuse. Pour celà, executer dans le profiler la commande* `[t,x,u]=EulerImplicite(EDP,5000,500);` *dans les deux cas et réaliser dans chaque cas un SnapShot (capture d'écran) que l'on nommera* `ProfilerMatFull` *et* `ProfilerMatSparse` *(format PNG sous Linux).*
3. *Que pourrait-on faire, au niveau algorithmique, pour améliorer les performances de ce code ?*
4. *Implémenter ces améliorations et vérifier s'il y a gain de performances avec le profiler de Matlab.* ▪

Q. 3 1. *Pour résoudre l'E.D.P. par un schéma d'**Euler explicite**, le programme* `mainChaleurExplicite.m` *(script Matlab) est fourni, ainsi que les fonctions* `CalculF.m`*,* `NormInf.m` *et* `PlotSol.m` *(ou* `PlotSol2.m`*) (voir fichier* `TP3.tar.gz`*). Dans les codes fournis, il manque le fichier* `EulerExplicite.m` *correspondant à la fonction :*

```
[t,x,u]=ChaleurExplicite(EDP,Nt,Nx)
```

résolvant l'E.D.P. par un schéma d'Euler explicite. Les paramètres sont identiques à ceux de la fonction `ChaleurImplicite.m`

2. *Dans le programme* `mainChaleurExplicite.m`*, changer le paramètre* `Nt` *de* 2100 *à* 2000*. Que se passe-t'il lors de la résolution ? Quel est le résultat théorique qui intervient ?*
3. *Ecrire un programme permettant de mettre en lumière ce phénomène.* ▪

Comme application possible nous allons résoudre numériquement par le schéma implicite d'Euler le problème de conduction thermique dans une barre de longueur $L = 6$:

$$\begin{aligned}
\frac{\partial u}{\partial t}(t,x) - \nu \frac{\partial^2 u}{\partial x^2}(t,x) &= 0,\ \forall (t,x) \in]0;T]\times]0;L[, &(1.7)\\
u(t_0,x) &= u_0(x),\ \forall x \in [0;L], &(1.8)\\
u(t,0) &= u_g(t),\ \forall t \in [0;T], &(1.9)\\
u(t,L) &= u_d(t),\ \forall t \in [0;T]. &(1.10)
\end{aligned}$$

avec $T = 10$, $u_0(x) = 100, \forall x \in [0, L]$

$$u_g(t) = \begin{cases} 100 - 90t, & \forall t \in [0,1] \\ 10,\ \forall t > 1 \end{cases}$$

$$u_d(t) = \begin{cases} 100 - 80t, & \forall t \in [0,1] \\ 20,\ \forall t > 1 \end{cases}$$

Q. 4 1. *Ecrire le programme* `barre1.m` *permettant de résoudre ce problème par le schéma d'Euler implicite.*

2. *Executer ce programme pour différentes valeurs de* ν *(par exemple* $\nu = 0.1$, $\nu = 1$ *et* $\nu = 10$*. Qu'observe-t'on ?* ▪

Pour la seconde application, la seule modification par rapport à l'application précédente est que la conductivité thermique est une fonction constante par morceaux en espace :

$$\nu(x) = \begin{cases} 1, & \forall x \in [0, \frac{L}{3}] \\ 0.1 & \forall x \in]\frac{L}{3}, \frac{2L}{3}] \\ 1 & \forall x \in]\frac{2L}{3}, L]. \end{cases}$$

Q. 5 *Ecrire le programme* `barre2.m` *permettant de résoudre ce problème par le schéma d'Euler implicite.* ▪

2 Conditions mixtes

On souhaite résoudre numériquement l'E.D.P. suivante

$$\frac{\partial u}{\partial t}(t,x) - \nu \frac{\partial^2 u}{\partial x^2}(t,x) = f(t,x), \qquad \forall (t,x) \in]t_0; t_0+T] \times]a; b[, \tag{2.1}$$

$$u(t_0, x) = u_0(x), \qquad \forall x \in [a; b], \tag{2.2}$$

$$\delta_a u(t,a) + \mu_a \frac{\partial u}{\partial x}(t,a) = g_a(t), \qquad \forall t \in [t_0; t_0+T], \tag{2.3}$$

$$\delta_b u(t,b) + \mu_b \frac{\partial u}{\partial x}(t,b) = g_b(t), \qquad \forall t \in [t_0; t_0+T], \tag{2.4}$$

avec ν un réel strictement positif, $t_0 \in \mathbb{R}$, $T > 0$, $(a,b) \in \mathbb{R}^2$, $a < b$.

Q. 6 *Reprendre les questions 1 à 3 pour le problème* (2.1)-(2.4)*. Lors de la programmation, les conditions aux limites seront intégrées à la structure* `EDP`*.* ▪

3 Schéma de Crank-Nicolson

Le schéma de Crank-Nicolson pour la discrétisation de l'équation de la chaleur (2.1) est donné par

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} - \frac{\nu}{2}\left(\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} + \frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{\Delta x^2}\right) = \frac{1}{2}\left(f_i^n + f_i^{n+1}\right). \tag{3.1}$$

Ce schéma est d'ordre 2 en temps et en espace et il est inconditionnellement stable.

Q. 7 *A vous d'écrire un programme permettant de résoudre le problème* (2.1)-(2.4) *en utilisant le schéma* (3.1)*. Il faudra bien évidemment proposer une méthodologie de tests/validations.* ▪