

TRAVAUX PRATIQUES - E.D.P.

Groupe 3

**Travail individuel et personnel**

Table des matières

<b>1 Différences finies 1D</b>	<b>1</b>
1.1 Problème modèle DD . . . . .	1
1.2 Problème modèle DN . . . . .	2
1.2.1 Neumann ordre 1 . . . . .	2
1.2.2 Neumann ordre 2 . . . . .	2
<b>2 Les Différences Finies pour l'équation de la chaleur</b>	<b>3</b>

**1 Différences finies 1D**

**1.1 Problème modèle DD**

On cherche à résoudre par un schéma de type différences finies le problème suivant

$$-u''(x) = f(x), \forall x \in ]a; b[ \tag{1.1}$$

$$u(a) = u_a \in \mathbb{R} \tag{1.2}$$

$$u(b) = u_b \in \mathbb{R} \tag{1.3}$$

On peut prendre comme jeux de données  $a = 0$ ,  $b = 2\pi$ ,  $f(x) = \sin(x) - 9 \cos(3x)$ ,  $u_a = -1$  et  $u_b = -1$ . Dans ce cas la solution exacte est  $u(x) = \sin(x) - \cos(3x)$ .

On définit la matrice associée au laplacien (en 1D opérateur dérivée seconde) discrétisé à l'ordre 2 par  $\mathbb{K} \in \mathcal{M}_d(\mathbb{R})$  avec

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

**Q. 1** *Ecrire une fonction Lap1D (fichier Lap1D.m) permettant de générer cette matrice.* ■

Le programme fourni (fichier Edp0.m), utilisant la fonction précédente, permet de résoudre numériquement le problème (1.1)-(1.2)-(1.3) par un schéma différences finies d'ordre 2. Dans ce programme, la variable  $\mathbf{x}$  contient l'ensemble des  $x_i$ ,  $i \in \llbracket 0, N \rrbracket$ , discrétisation régulière de l'intervalle  $[a; b]$  définie par

$$x_i = a + ih, \text{ avec } h = (b - a)/N.$$

**Q. 2** *Ecrire le programme OrdreEdp0 (fichier OrdreEDP0.m) permettant de représenter l'erreur en fonction du pas  $h$  de discrétisation en échelle logarithmique d'afficher l'ordre de la méthode.* ■

**Q. 3** 1. Ecrire le code Matlab (fichier Edp1.m) permettant de calculer une solution approchée de :

$$-u''(x) + \nu u(x) = f(x), \forall x \in ]a; b[ \quad (1.4)$$

$$u(a) = u_a \in \mathbb{R} \quad (1.5)$$

$$u(b) = u_b \in \mathbb{R} \quad (1.6)$$

avec  $\nu \geq 0$ . Le choix des données est laissé libre.

2. Ecrire le programme ValidEdp1 (fichier ValidEdp1.m) permettant de valider le code précédent. ■

**A faire**

- ◇ Créer une archive compressée nommée <NOM>-TP2-Q1aQ3 contenant l'ensemble des fichiers nécessaires à l'exécution des programmes demandés en Q1 à Q3. Ici <NOM> correspond évidemment à votre nom.
- ◇ Envoyer un mail à [cuvelier@math.univ-paris13.fr](mailto:cuvelier@math.univ-paris13.fr) ayant pour **sujet** "<NOM> TP2 Q1 a Q3".

## 1.2 Problème modèle DN

On cherche à résoudre par un schéma de type différences finies le problème suivant

$$-u''(x) = f(x), \forall x \in ]a; b[ \quad (1.7)$$

$$u(a) = u_a \in \mathbb{R} \quad (1.8)$$

$$u'(b) = v_b \in \mathbb{R} \quad (1.9)$$

On peut prendre comme jeux de données  $a = 0$ ,  $b = 2\pi$ ,  $f(x) = \sin(x) + 6$ ,  $u_a = 0$  et  $v_b = 1 - 12\pi$ . Dans ce cas la solution exacte est  $u(x) = \sin(x) - 3x^2$ .

### 1.2.1 Neumann ordre 1

Dans la condition aux limites de Neumann (1.9), on va approcher  $u'(b)$  à l'ordre 1 par  $\frac{u(b)-u(b-h)}{h}$

**Q. 4** Ecrire le programme Edp2 (fichier Edp2.m) permettant de représenter une solution approchée du problème précédent. ■

**Q. 5** Ecrire le programme OrdreEdp2 (fichier OrdreEDP2.m) permettant de représenter l'erreur en fonction du pas  $h$  de discrétisation en échelle logarithmique et d'afficher l'ordre de la méthode. ■

### 1.2.2 Neumann ordre 2

Dans la condition aux limites de Neumann (1.9), on va approcher  $u'(b)$  à l'ordre 2 par  $\frac{u(b-2h)-4u(b-h)+3u(b)}{2h}$

**Q. 6** Ecrire le programme Edp2 (fichier Edp2.m) permettant de représenter une solution approchée du problème précédent. ■

**Q. 7** Ecrire le programme OrdreEdp2 (fichier OrdreEDP2.m) permettant de représenter l'erreur en fonction du pas  $h$  de discrétisation en échelle logarithmique et d'afficher l'ordre de la méthode. ■

**A faire**

- ◇ Créer une archive compressée nommée <NOM>-TP2-Q4aQ7 contenant l'ensemble des fichiers nécessaires à l'exécution des programmes demandés en Q4 à Q7. Ici <NOM> correspond évidemment à votre nom.
- ◇ Envoyer un mail à [cuvelier@math.univ-paris13.fr](mailto:cuvelier@math.univ-paris13.fr) ayant pour **sujet** "<NOM> TP2 Q4 a Q7".

## 2 Les Différences Finies pour l'équation de la chaleur

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

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

$$u(t_0, x) = u_0(x), \quad \forall x \in [a; b], \quad (2.2)$$

$$u(t, a) = u_a(t), \quad \forall t \in [t_0; t_0 + T], \quad (2.3)$$

$$u(t, b) = u_b(t), \quad \forall t \in [t_0; t_0 + T]. \quad (2.4)$$

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

On note  $t^n$ ,  $n \in \llbracket 0, N_t \rrbracket$  et  $x_i$ ,  $i \in \llbracket 0, N_x \rrbracket$  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. :

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} - \nu \frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{\Delta x^2} = f_i^{n+1}. \quad (2.5)$$

et

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} - \nu \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} = f_i^n. \quad (2.6)$$

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 inconditionnellement stable et le schéma d'Euler explicite est stable sous la condition de C.F.L.

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

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

**Q. 8** Pour chaque schéma, écrire sur feuille et **de manière détaillée** la discrétisation de l'E.D.P. (2.1) à (2.4) ■

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. 9** 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` (voir fichier `CodesChaleurImplicite.tar.gz`). Dans les codes fournis, il manque le fichier `ChaleurImplicite.m` correspondant à la fonction :

```
[t, x, u] = ChaleurImplicite(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  $Nt+1$ ),  $t(n) = t^{n-1}$ ,  $\forall n \in \llbracket 1, N_t + 1 \rrbracket$ ,
- **x** : discrétisation en espace (dimension  $Nx+1$ ),  $x(i) = x_{i-1}$ ,  $\forall i \in \llbracket 1, N_x + 1 \rrbracket$ ,
- **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'exécution de la fonction `ChaleurImplicite` dans les cas où la matrice est stockée de manière pleine ou creuse. Pour cela, exécuter dans le profiler la commande `[t,x,u]=ChaleurImplicite(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). ■

**Q. 10** 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`. (voir fichier `CodesChaleurImplicite.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 ? ■

### A faire

- ◇ Rendre la(les) feuille(s) contenant la réponse à la question Q8. N'oubliez pas d'écrire votre nom sur chacune des feuilles et de les numéroter.
- ◇ Créer les 2 archives compressées nommées <NOM>-TP2-Q9 et <NOM>-TP2-Q10 contenant respectivement l'ensemble des fichiers nécessaires à l'exécution des programmes demandés en Q9 et Q10. Ici <NOM> correspond évidemment à votre nom.
- ◇ Envoyer un mail à `cuvelier@math.univ-paris13.fr` ayant pour **sujet** "<NOM> TP2 Q9a10" et en fichiers joints les 2 archives compressées créées précédemment ainsi que les captures d'écran (fichiers `ProfilerMatFull.png` et `ProfilerMatSparse.png`) de Q9.

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

$$\frac{\partial u}{\partial t}(t, x) - \nu \frac{\partial^2 u}{\partial x^2}(t, x) = 0, \quad \forall (t, x) \in ]0; T] \times ]0; L[, \quad (2.7)$$

$$u(t_0, x) = u_0(x), \quad \forall x \in [0; L], \quad (2.8)$$

$$u(t, 0) = u_g(t), \quad \forall t \in [0; T], \quad (2.9)$$

$$u(t, L) = u_d(t), \quad \forall t \in [0; T]. \quad (2.10)$$

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. 11** 1. Ecrire le programme `barre1.m` permettant de résoudre ce problème par le schéma d'Euler implicite.  
2. Exécuter ce programme pour différentes valeurs de  $\nu$  (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. 12** *Ecrire le programme `barre2.m` permettant de résoudre ce problème par le schéma d'Euler implicite.* ■

**A faire**

- ◇ Créer l'archive compressée nommée `<NOM>-TP2-Q11a12` contenant l'ensemble des fichiers nécessaires à l'exécution des programmes demandés en Q11 et Q12. Ici `<NOM>` correspond évidemment à votre nom.
- ◇ Envoyer un mail à `cuvelier@math.univ-paris13.fr` ayant pour **objet** "`<NOM> TP2 Q11a12`" et en fichier joint l'archive compressée créée précédemment.