

TRAVAUX PRATIQUES - E.D.P.

Groupe 1

Travail individuel et personnel

Table des matières

1	Les différences finies pour une EDP 1D stationnaire	1
1.1	Problème modèle Dirichlet/Dirichlet	1
1.2	Problème modèle Neumann/Dirichlet	3
1.2.1	Neumann ordre 1	3
1.2.2	Neumann ordre 2	3
2	Les Différences Finies pour l'équation de la chaleur	3

1 Les différences finies pour une EDP 1D stationnaire

1.1 Problème modèle Dirichlet/Dirichlet

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

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

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

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

Par exemple, on peut prendre comme jeu de données $a = -\frac{1}{3}\pi$, $b = \pi$, $f(x) = -9 \cos(3x - 2) + \sin(-\frac{1}{7}\pi + x)$, $u_a = -\cos(\pi + 2) - \sin(\frac{10}{21}\pi)$ et $u_b = -\cos(3\pi - 2) + \sin(\frac{1}{7}\pi)$. Dans ce cas la solution exacte est $u(x) = -\cos(3x - 2) + \sin(-\frac{1}{7}\pi + x)$.

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 et un petit programme (fichier `prg0.m`) permettant de la tester/valider rapidement. ■

L'archive fourni sur le serveur Discord (fichier `CodesEDPOG1.zip`) contient le script/programme `EDPO`, et les fonctions `setEDPO` et `solveEDPO`, permettant de résoudre numériquement le problème (1.1)-(1.2)-(1.3) par un schéma différences finies d'ordre 2. Ce programme va afficher la solution numérique obtenue, la solution exacte et l'erreur. La fonction `solveEDPO` utilise la fonction `Lap1D` que vous avez implémentée.

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 et d'afficher l'ordre de la méthode. Un exemple de représentation est donné en Figure 1. ■

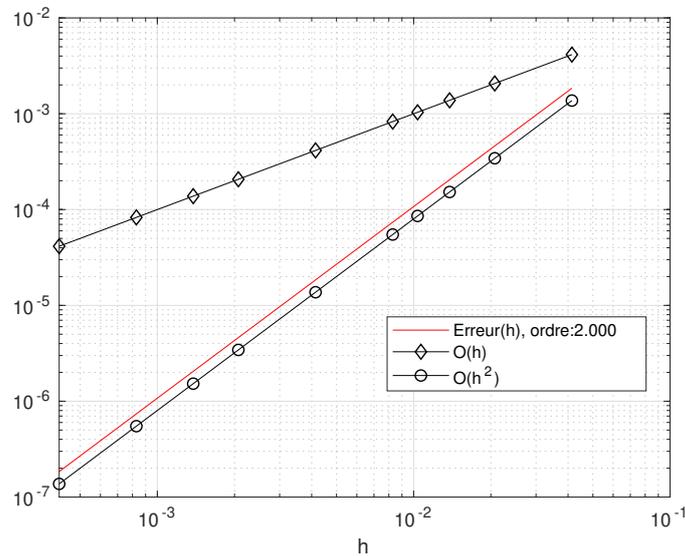


FIGURE 1 – Représentation de l'erreur en fonction de h

Q. 3 1. Sur le même principe que l'exemple précédent, écrire les fonctions `setEDP1` et `solveEDP1`, ainsi que le programme `EDP1` 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 $\nu \geq 0$. Avec un choix judicieux de données, ce programme représentera la solution approchée ainsi que la solution exacte.

2. Ecrire le programme `OrdreEDP1` (fichier `OrdreEDP1.m`) permettant de représenter, en échelle logarithmique, l'erreur en fonction du pas h de discrétisation et d'afficher l'ordre de la méthode. Le jeu de données sera choisi judicieusement. ■

A faire en 1h00 (temps indicatif)

- ◇ Créer une archive compressée nommée `<NOM>-TP2-part1-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 sur Discord un **message privé** à l'enseignant ayant pour **contenu** "`<NOM> TP2 part1 Q1 a Q3`" et y joindre l'archive.

1.2 Problème modèle Neumann/Dirichlet

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

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

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

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

Par exemple, on peut prendre comme jeu de données $a = -\frac{1}{3}\pi$, $b = 2$, $f(x) = 12x^2 + 9 \cos(3x)$, $v_a = \frac{4}{27}\pi^3$ et $u_b = \cos(6) - 16$. Dans ce cas la solution exacte est $u(x) = -x^4 + \cos(3x)$.

1.2.1 Neumann ordre 1

Dans la condition aux limites de Neumann (1.8), on va approcher $u'(a)$ à l'ordre 1 par $\frac{u(a+h)-u(a)}{h}$

Q. 4 Ecrire les fonctions `setEDP2` et `solveEDP2`, ainsi que le programme `EDP2` 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 (1.8), on va approcher $u'(a)$ à l'ordre 2 par $\frac{-u(a+2h)+4u(a+h)-3u(a)}{2h}$

Q. 6 Ecrire la fonction `solveEDP3` permettant de résoudre le problème (1.7)-(1.8)-(1.9) à l'ordre 2 ainsi que le programme `EDP3` permettant de représenter une solution approchée à l'ordre 2 de ce problème. ■

Q. 7 Ecrire le programme `OrdreEdp3` (fichier `OrdreEDP3.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 en 2h00 (temps indicatif) —

- ◊ Créer une archive compressée nommée `<NOM>-TP2-part1-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 sur Discord un **message privé** à l'enseignant ayant pour contenu "`<NOM> TP2 part1 Q4 a Q7`" et y joindre l'archive.

2 Les Différences Finies pour l'équation de la chaleur

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

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

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

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

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

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

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. :

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

et

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

où $\Delta t = T/N_t$, $\Delta x = (b-a)/N_x$, $f_i^n = f(t^n, x_i)$, $c_i = c(x_i)$ 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 $U_i^n = u_{i-1}^n$, $\forall i \in \llbracket 1, N_x + 1 \rrbracket$.

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 $t_0 = 0$, $T = 2$, $a = 0$, $b = \pi$, $\nu = 2$, $k = 5$

$$\begin{aligned} c(x) &= x^2 + 1, \\ f(t, x) &= (x^2 + 1) \cos(kt) \sin\left(-\frac{1}{3}\pi + 2x\right) - k \sin\left(-\frac{1}{3}\pi + 2x\right) \sin(kt) + 8 \cos(kt) \sin\left(-\frac{1}{3}\pi + 2x\right), \\ u_0(x) &= \sin\left(-\frac{1}{3}\pi + 2x\right), \\ u_a(t) &= -\frac{1}{2}\sqrt{3} \cos(kt), \\ u_b(t) &= -\frac{1}{2}\sqrt{3} \cos(kt). \end{aligned}$$

Dans ce cas, la solution exacte est donnée par $u_{\text{ex}}(t, x) = \cos(kt) \sin\left(-\frac{1}{3}\pi + 2x\right)$.

Q. 9 Pour résoudre l'E.D.P. par un schéma d'Euler implicite, des codes sont fournis sur le serveur Discord dans l'archive `CodesEDPChaleurG1.zip`. Cette archive contient le programme `mainChaleurImplicite.m` (script Matlab), ainsi que les fonctions `setEDP.m`, `CalculF.m`, `NormInf.m` et `PlotSol.m`. Pour exécuter le programme `mainChaleurImplicite`, 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, retournée par la fonction `setEDP`, 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`),
- `x` : discrétisation en espace (dimension `Nx+1`),
- `u` : `u(i,n)` solution approchée au temps `t(n)` et point `x(i)` (dimension (`Nx+1,Nt+1`)),

Écrire la fonction `EulerImplicite`. ■

Q. 10 1. Pour résoudre l'E.D.P. par un schéma d'Euler explicite, le programme `mainChaleurExplicite` (script Matlab) est fourni, ainsi que les fonctions `setEDP.m`, `CalculF.m`, `NormInf.m` et `PlotSol.m` (voir question précédente).

Il manque le fichier `EulerExplicite.m` correspondant à la fonction :

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

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

Écrire la fonction `EulerImplicite`.

2. Dans le programme `mainChaleurExplicite`, changer le paramètre `Nt` de 4000 à 3900. Que se passe-t-il lors de la résolution ? ■

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 `Q11main.m` permettant de résoudre ce problème par le schéma d'Euler implicite.
2. Exécuter 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. 12** 1. Expliquer rapidement, sur feuille, quels sont les modifications à apporter au programme précédent.
2. Ecrire le programme `Q12main.m` permettant de résoudre ce problème par le schéma d'Euler implicite. ■

A faire en 3h00 (temps indicatif)

- ◇ Créer les 4 archives compressées nommées <NOM>-TP2-Q9 à <NOM>-TP2-Q12 contenant respectivement l'ensemble des fichiers nécessaires à l'exécution des programmes demandés en Q9 à Q12. Ici <NOM> correspond évidemment à votre nom.
- ◇ Envoyer sur Discord un **message privé** à l'enseignant ayant pour contenu "<NOM> TP2 Q9a12" et y joindre les 4 archives compressées créées précédemment ainsi qu'un scan (format pdf) de la réponse manuscrite à la question Q.12 1.