

4.6 Problème modèle évolutif 1D : équation de la chaleur



EDP modèle instationnaire en dimension 1 : équation de la chaleur

Trouver $u : [0, T] \times [a, b] \rightarrow \mathbb{R}$ telle que

$$\frac{\partial u}{\partial t}(t, x) - D \frac{\partial^2 u}{\partial x^2}(t, x) = f(t, x), \quad \forall (t, x) \in]0, T[\times]a, b[, \quad (4.76)$$

$$u(0, x) = u_0(x), \quad \forall x \in [a, b] \quad (4.77)$$

$$-D \frac{\partial u}{\partial x}(t, a) = \alpha(t), \quad \forall t \in [0, T] \quad (4.78)$$

$$u(t, b) = \beta(t), \quad \forall t \in [0, T] \quad (4.79)$$

où $a < b$, $D > 0$ (coefficient de diffusivité), $\alpha : [0, T] \rightarrow \mathbb{R}$, $\beta : [0, T] \rightarrow \mathbb{R}$, $u_0 : [a, b] \rightarrow \mathbb{R}$ et $f : [0, T] \times [a, b] \rightarrow \mathbb{R}$ donnés. **condition de compatibilité :**

$$u_0(b) = \beta(0). \quad (4.80)$$

$$x_i = a + i\Delta_x, \quad \forall i \in \llbracket 0, N_x \rrbracket, \quad \text{avec } \Delta_x = \frac{b-a}{N_x}$$

$$t^n = n\Delta_t, \quad \forall n \in \llbracket 0, N_t \rrbracket, \quad \text{avec } \Delta_t = \frac{T}{N_t}.$$



EDP modèle d'évolution en dimension 1 : équation de la chaleur, formulation aux points de discrétisation

Trouver $u(t^n, x_i) \in \mathbb{R}$, $\forall n \in \llbracket 0, N_t \rrbracket$, $\forall i \in \llbracket 0, N_x \rrbracket$, tels que

$$\frac{\partial u}{\partial t}(t^n, x_i) - D \frac{\partial^2 u}{\partial x^2}(t^n, x_i) = f(t^n, x_i), \quad \forall n \in \llbracket 0, N_t \rrbracket, \quad \forall i \in \llbracket 0, N_x \rrbracket, \quad (4.81)$$

$$u(0, x_i) = u_0(x_i), \quad \forall i \in \llbracket 0, N_x \rrbracket, \quad (4.82)$$

$$-D \frac{\partial u}{\partial x}(t^n, x_0) = \alpha(t^n), \quad \forall n \in \llbracket 0, N_t \rrbracket \quad (4.83)$$

$$u(t^n, x_{N_x}) = \beta(t^n), \quad \forall n \in \llbracket 0, N_t \rrbracket \quad (4.84)$$

- Approximation d'ordre 2 de $\frac{\partial^2 u}{\partial x^2}(t^n, x_i)$:

$$\frac{\partial^2 u}{\partial x^2}(t^n, x_i) = \frac{u(t^n, x_{i+1}) - 2u(t^n, x_i) + u(t^n, x_{i-1}))}{\Delta_x^2} + \mathcal{O}(\Delta_x^2) \quad (4.85)$$

- Approximations d'ordre 1 de $\frac{\partial u}{\partial t}(t, x)$:

$$\frac{\partial u}{\partial t}(t^n, x_i) = \frac{u(t^{n+1}, x_i) - u(t^n, x_i)}{\Delta_t} + \mathcal{O}(\Delta_t) \quad (4.86)$$

$$\frac{\partial u}{\partial t}(t^n, x_i) = \frac{u(t^n, x_i) - u(t^{n-1}, x_i)}{\Delta_t} + \mathcal{O}(\Delta_t). \quad (4.87)$$

- Approximations d'ordre 1 de $\frac{\partial u}{\partial x}(t^n, x_0)$:

$$\frac{\partial u}{\partial x}(t^n, x_0) = \frac{u(t^n, x_1) - u(t^n, x_0)}{\Delta_x} + \mathcal{O}(\Delta_x)$$

- Approximations d'ordre 2 de $\frac{\partial u}{\partial x}(t^n, x_0)$:

$$\frac{\partial u}{\partial x}(t^n, x_0) = \frac{-3u(t^n, x_0) + 4u(t^n, x_1) - u(t^n, x_2)}{2\Delta_x} + \mathcal{O}(\Delta_x^2) \quad (4.88)$$

4.6.1 Schéma explicite pour (4.81)-(4.84)

On espère trouver une méthode permettant de déterminer

$$\mathbf{u}_i^n \approx u(t^n, x_i), \quad \forall n \in \llbracket 0, N_t \rrbracket, \quad \forall i \in \llbracket 0, N_x \rrbracket$$

- Un schéma numérique d'ordre 1 en temps et d'ordre 2 en espace pour (4.81): $\forall n \in \llbracket 0, N_t \rrbracket, \quad \forall i \in \llbracket 0, N_x \rrbracket$

$$\frac{\mathbf{u}_i^{n+1} - \mathbf{u}_i^n}{\Delta t} - D \frac{\mathbf{u}_{i+1}^n - 2\mathbf{u}_i^n + \mathbf{u}_{i-1}^n}{\Delta x^2} = \mathbf{f}_i^n \quad (4.89)$$

avec $\mathbf{f}_i^n = f(t^n, x_i)$ et (en espérant) $\mathbf{u}_i^n \approx u(t^n, x_i)$. (4.89) est équivalent à

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^n + D \frac{\Delta t}{\Delta x^2} (\mathbf{u}_{i+1}^n - 2\mathbf{u}_i^n + \mathbf{u}_{i-1}^n) + \Delta t \mathbf{f}_i^n \quad (4.90)$$

- Schémas pour (4.82) et (4.84):

$$\begin{aligned} \mathbf{u}_i^0 &= u_0(x_i), & \forall i \in \llbracket 0, N_x \rrbracket \\ \mathbf{u}_{N_x}^n &= \beta(t^n), & \forall n \in \llbracket 0, N_t \rrbracket \end{aligned} \quad (4.91)$$

- Un schéma numérique d'ordre 2 en espace pour (4.83):

$$-D \frac{-3\mathbf{u}_0^n + 4\mathbf{u}_1^n - \mathbf{u}_2^n}{\Delta x} = \alpha(t^n)$$

ou encore

$$\mathbf{u}_0^n = \frac{1}{3} \left(\frac{2\Delta x}{D} \alpha(t^n) + 4\mathbf{u}_1^n - \mathbf{u}_2^n \right). \quad (4.105)$$

Principe : On connaît $(\mathbf{u}_i^0)_{i=0}^{N_x}$, on va calculer successivement :

$$(\mathbf{u}_i^1)_{i=0}^{N_x}, \text{ puis } (\mathbf{u}_i^2)_{i=0}^{N_x}, \dots, \text{ puis } (\mathbf{u}_i^{N_t})_{i=0}^{N_x}$$

On calcule $(\mathbf{u}_i^{n+1})_{i=0}^{N_x}$ à partir de $(\mathbf{u}_i^n)_{i=0}^{N_x}$:

$$\mathbf{u}_i^{n+1} = C\mathbf{u}_i^n + E(\mathbf{u}_{i+1}^n + \mathbf{u}_{i-1}^n) + \Delta t \mathbf{f}_i^n, \quad \forall i \in \llbracket 0, N_x \rrbracket, \quad (4.107)$$

$$\mathbf{u}_{N_x}^{n+1} = \beta(t^{n+1}), \quad (4.108)$$

$$\mathbf{u}_0^{n+1} = \frac{1}{3} \left(\frac{2\Delta x}{D} \alpha(t^{n+1}) + 4\mathbf{u}_1^{n+1} - \mathbf{u}_2^{n+1} \right). \quad (4.109)$$

L'algorithme formel est donc le suivant avec $E = D \frac{\Delta t}{\Delta x^2}$ et $C = 1 - 2E$:

- 1: $\mathbf{u}_i^0 \leftarrow u_0(x_i), \quad \forall i \in \llbracket 0, N_x \rrbracket$
- 2: **Pour** $n \leftarrow 0$ à $N_t - 1$ **faire**
- 3: $\mathbf{u}_i^{n+1} \leftarrow C\mathbf{u}_i^n + E(\mathbf{u}_{i+1}^n + \mathbf{u}_{i-1}^n) + \Delta t \mathbf{f}_i^n, \quad \forall i \in \llbracket 0, N_x \rrbracket$
- 4: $\mathbf{u}_{N_x}^{n+1} \leftarrow \beta(t^{n+1})$
- 5: $\mathbf{u}_0^{n+1} \leftarrow \frac{1}{3} \left(\frac{2\Delta x}{D} \alpha(t^{n+1}) + 4\mathbf{u}_1^{n+1} - \mathbf{u}_2^{n+1} \right)$
- 6: **Fin Pour**

Exercice 4.6.1

Q. 1 Ecrire une fonction algorithmique `HEAT1DEX` permettant de retourner la discrétisation en temps, la discrétisation en espace et l'ensemble des \mathbf{u}_i^n , $i \in \llbracket 0, N_x \rrbracket$, $n \in \llbracket 1, N_t \rrbracket$ calculés par le schéma explicite en temps pour l'EDP (4.81) à (4.84).

Q. 2 Ecrire un programme permettant de calculer la solution numérique d'un problème dont on connaît la solution exacte.

Etude de la **stabilité** au sens de Von Neumann du schéma explicite donne la **condition de C.F.L.** (R. Courant, K. Friedrichs, and H. Lewy en 1928):

le schéma est **stable** si

$$D \frac{\Delta t}{dx^2} \leq \frac{1}{2}. \quad (4.110)$$

4.6.2 Schéma implicite pour (4.81)-(4.84)

Pour approcher $\frac{\partial u}{\partial t}(t^n, x_i)$ dans (4.81) on utilise cette fois

$$\frac{\partial u}{\partial t}(t^n, x_i) = \frac{u(t^n, x_i) - u(t^{n-1}, x_i)}{\Delta_t} + \mathcal{O}(\Delta_t). \quad (4.87)$$

Un schéma numérique d'ordre 1 en temps et d'ordre 2 en espace pour (4.81): $\forall n \in \llbracket 0, N_t \rrbracket, \forall i \in \llbracket 0, N_x \rrbracket$

$$\frac{\mathbf{u}_i^n - \mathbf{u}_i^{n-1}}{\Delta_t} - D \frac{\mathbf{u}_{i+1}^n - 2\mathbf{u}_i^n + \mathbf{u}_{i-1}^n}{\Delta_x^2} = \mathbf{f}_i^n, \quad (4.111)$$

avec $\mathbf{f}_i^n = f(t^n, x_i)$ et (en espérant) $\mathbf{u}_i^n \approx u(t^n, x_i)$.

(4.114) est équivalent à

$$\mathbf{u}_i^n - D \frac{\Delta_t}{\Delta_x^2} (\mathbf{u}_{i+1}^n - 2\mathbf{u}_i^n + \mathbf{u}_{i-1}^n) = \mathbf{u}_i^{n-1} + \Delta_t \mathbf{f}_i^n. \quad (4.112)$$

Ce schéma est **implicite en temps** : il n'est pas possible de calculer explicitement \mathbf{u}_i^n en fonction des \mathbf{u}_i^{n-1} (au temps précédent).

Le calcul des $\mathbf{u}_i^n, \forall i \in \llbracket 0, N_x \rrbracket$, est possible en résolvant les $N_x + 1$ équations **linéaires** suivantes où l'on note $E = D \frac{\Delta_t}{\Delta_x^2}$ et $C = 1 + 2E$:

$$3\mathbf{u}_0^n - 4\mathbf{u}_1^n + \mathbf{u}_2^n = 2 \frac{\Delta_x}{D} \alpha(t^n). \quad (4.113)$$

$$C\mathbf{u}_i^n - E(\mathbf{u}_{i+1}^n + \mathbf{u}_{i-1}^n) = \mathbf{u}_i^{n-1} + \Delta_t \mathbf{f}_i^n, \quad \forall i \in \llbracket 0, N_x \rrbracket \quad (4.114)$$

$$\mathbf{u}_{N_x}^n = \beta(t^n), \quad (4.115)$$

$$\mathbb{A} \mathbf{U}^n = \mathbf{b}^n \quad (4.116)$$

avec $\mathbb{A} \in \mathcal{M}_{N_x+1}(\mathbb{R}), \mathbf{b}^n \in \mathbb{R}^{N_x+1}$ et

$$\mathbf{U}^n = \begin{pmatrix} \mathbf{u}_0^n \\ \mathbf{u}_1^n \\ \vdots \\ \mathbf{u}_{N_x-1}^n \\ \mathbf{u}_{N_x}^n \end{pmatrix} \in \mathbb{R}^{N_x+1}, \quad \text{i.e. } \mathbf{U}^n(i) = \mathbf{u}_{i-1}^n, \quad \forall i \in \llbracket 1, N_x + 1 \rrbracket$$

$$\mathbb{A} \mathbf{U}^n = \mathbf{b}^n \iff$$

$$\begin{pmatrix} 3 & -4 & 1 & 0 & \cdots & 0 & 0 \\ -E & C & -E & 0 & \cdots & 0 & 0 \\ 0 & -E & C & -E & \ddots & \vdots & \vdots \\ \vdots & 0 & \ddots & \ddots & \ddots & 0 & \vdots \\ \vdots & \vdots & \ddots & -E & C & -E & 0 \\ 0 & 0 & \cdots & 0 & -E & C & -E \\ 0 & 0 & \cdots & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{u}_0^n \\ \mathbf{u}_1^n \\ \mathbf{u}_2^n \\ \vdots \\ \mathbf{u}_{N_x-2}^n \\ \mathbf{u}_{N_x-1}^n \\ \mathbf{u}_{N_x}^n \end{pmatrix} = \begin{pmatrix} 2 \frac{\Delta_x}{D} \alpha(t^n) \\ \mathbf{u}_1^{n-1} + \Delta_t \mathbf{f}_1^n \\ \mathbf{u}_2^{n-1} + \Delta_t \mathbf{f}_2^n \\ \vdots \\ \mathbf{u}_{N_x-2}^{n-1} + \Delta_t \mathbf{f}_{N_x-2}^n \\ \mathbf{u}_{N_x-1}^{n-1} + \Delta_t \mathbf{f}_{N_x-1}^n \\ \beta(t^n) \end{pmatrix} \quad (4.117)$$

L'algorithme formel est donc le suivant :

- 1: $\mathbf{U}_i^0 \leftarrow u_0(x_i), \forall i \in \llbracket 0, N_x \rrbracket$
- 2: **Pour** $n \leftarrow 1$ **à** N_t **faire**
- 3: Calcul de \mathbf{U}^n en résolvant le système linéaire (4.117)
- 4: **Fin Pour**



Exercice 4.6.2

Q. 1 Ecrire la fonction `ASSEMBLEMATGEN1D` retournant la matrice $\mathbb{M} \in \mathcal{M}_d(\mathbb{R})$ définie par

$$\mathbb{M} = \begin{pmatrix} a_1 & a_2 & a_3 & 0 & \dots & \dots & 0 \\ \beta & \alpha & \beta & 0 & \dots & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & \dots & 0 & \beta & \alpha & \beta \\ 0 & \dots & \dots & 0 & b_3 & b_2 & b_1 \end{pmatrix} \quad (4.118)$$

où $\alpha, \beta, a_1, a_2, a_3, b_1, b_2$ et b_3 sont des réels donnés.

Q. 2 Ecrire la fonction `SNDMBRGEN1D` retournant le vecteur $\mathbf{B} \in \mathbb{R}^d$ défini par

$$\mathbf{B} = \begin{pmatrix} \alpha \\ c_1 \\ \vdots \\ c_{d-2} \\ \beta \end{pmatrix} \quad (4.119)$$

où $\alpha, \beta, c_1, \dots, c_{d-2}$ sont des réels donnés.



Exercice 4.6.3

Q. 1 Ecrire une fonction algorithmique `HEAT1DIM` permettant de retourner la discrétisation en temps, la discrétisation en espace et l'ensemble des \mathbf{u}_i^n , $i \in \llbracket 0, N_x \rrbracket$, $n \in \llbracket 1, N_t \rrbracket$ calculés par le schéma *implicite* en temps pour l'EDP (4.81) à (4.84).

Q. 2 Ecrire un programme utilisant cette fonction et permettant de calculer la solution numérique d'un problème dont on connaît la solution exacte.