4.6 Problème modèle évolutif 1D : équation de la chaleur

EDP modèle instationnaire en dimension 1 : équation de la chaleur

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

$$\frac{\partial u}{\partial t}(t,x) - D\frac{\partial^2 u}{\partial x^2}(t,x) = f(t,x), \quad \forall (t,x) \in]0,T]\times]a,b[, \tag{4.76}$$
$$u(0,x) = u_0(x), \quad \forall x \in [a,b] \tag{4.77}$$
$$-D\frac{\partial u}{\partial x}(t,a) = \alpha(t), \quad \forall t \in [0,T] \tag{4.78}$$
$$u(t,b) = \beta(t), \quad \forall t \in [0,T] \tag{4.79}$$

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

$$u_0(b) = \beta(0). \tag{4.80}$$

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

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

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

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

$$\frac{\partial u}{\partial t}(t^n,x_i) - D\frac{\partial^2 u}{\partial x^2}(t^n,x_i) = f(t^n,x_i), \quad \forall n \in]\!]0, N_t]\!],\ \forall i \in]\!]0, N_x[\![, \tag{4.81}$$
$$u(0,x_i) = u_0(x_i), \quad \forall i \in [\![0, N_x]\!], \tag{4.82}$$
$$-D\frac{\partial u}{\partial x}(t^n,x_0) = \alpha(t^n), \quad \forall n \in [\![0, N_t]\!] \tag{4.83}$$
$$u(t^n,x_{N_x}) = \beta(t^n), \quad \forall n \in [\![0, N_t]\!] \tag{4.84}$$

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

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

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

$$\frac{\partial u}{\partial t}(t^n,x_i) = \frac{u(t^{n+1},x_i) - u(t^n,x_i)}{\Delta_t} + \mathcal{O}(\Delta_t) \tag{4.86}$$
$$\frac{\partial u}{\partial t}(t^n,x_i) = \frac{u(t^n,x_i) - u(t^{n-1},x_i)}{\Delta_t} + \mathcal{O}(\Delta_t). \tag{4.87}$$

- 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 $\frac{\partial u}{\partial x}(t^n, x_0)$:

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

4.6.1 Schéma explicite pour (4.81)-(4.84)

On espère trouver une méthode permettant de déterminer

$$\boldsymbol{u}_i^n \approx u(t^n, x_i), \quad \forall n \in [\![0, N_t]\!], \ \forall i \in [\![0, N_x]\!]$$

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

$$\frac{\boldsymbol{u}_i^{n+1} - \boldsymbol{u}_i^n}{\Delta_t} - D\frac{\boldsymbol{u}_{i+1}^n - 2\boldsymbol{u}_i^n + \boldsymbol{u}_{i-1}^n}{\Delta_x^2} = \boldsymbol{f}_i^n \tag{4.89}$$

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

$$\boldsymbol{u}_i^{n+1} = \boldsymbol{u}_i^n + D\frac{\Delta_t}{\Delta_x^2}\left(\boldsymbol{u}_{i+1}^n - 2\boldsymbol{u}_i^n + \boldsymbol{u}_{i-1}^n\right) + \Delta_t \boldsymbol{f}_i^n \tag{4.90}$$

- Schémas pour (4.82) et (4.84):

$$\begin{aligned} \boldsymbol{u}_i^0 &= u_0(x_i), && \forall i \in [\![0, N_x]\!] \\ \boldsymbol{u}_{N_x}^n &= \beta(t^n), && \forall n \in [\![0, N_t]\!] \end{aligned} \tag{4.91}$$

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

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

  ou encore

$$\boldsymbol{u}_0^n = \frac{1}{3}\left(\frac{2\Delta_x}{D}\alpha(t^n) + 4\boldsymbol{u}_1^n - \boldsymbol{u}_2^n\right). \tag{4.105}$$

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

$$(\boldsymbol{u}_i^1)_{i=0}^{N_x}, \text{ puis } (\boldsymbol{u}_i^2)_{i=0}^{N_x}, \ldots, \text{ puis } (\boldsymbol{u}_i^{N_x})_{i=0}^{N_x}$$

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

$$\boldsymbol{u}_i^{n+1} = C\boldsymbol{u}_i^n + E(\boldsymbol{u}_{i+1}^n + \boldsymbol{u}_{i-1}^n) + \Delta_t \boldsymbol{f}_i^n, \ \forall i \in]\!]0, N_x[\![, \tag{4.107}$$

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

$$\boldsymbol{u}_0^{n+1} = \frac{1}{3}\left(\frac{2\Delta_x}{D}\alpha(t^{n+1} + 4\boldsymbol{u}_1^{n+1} - \boldsymbol{u}_2^{n+1}\right). \tag{4.109}$$

L'algorithme formel est donc le suivant avec $E = D\frac{\Delta_t}{\Delta_x^2}$ et $C = 1 - 2E$:

$\boldsymbol{u}_i^0 \leftarrow u_0(x_i), \ \forall i \in [\![0, N_x]\!]$
Pour $n \leftarrow 0$ **à** $N_t - 1$ **faire**
  $\boldsymbol{u}_i^{n+1} \leftarrow C\boldsymbol{u}_i^n + E(\boldsymbol{u}_{i+1}^n + \boldsymbol{u}_{i-1}^n) + \Delta_t \boldsymbol{f}_i^n, \ \forall i \in]\!]0, N_x[\![$
  $\boldsymbol{u}_{N_x}^{n+1} \leftarrow \beta(t^{n+1})$
  $\boldsymbol{u}_0^{n+1} \leftarrow \frac{1}{3}\left(\frac{2\Delta_x}{D}\alpha(t^{n+1}) + 4\boldsymbol{u}_1^{n+1} - \boldsymbol{u}_2^{n+1}\right)$
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* $\boldsymbol{u}_i^n$, $i \in [\![0, N_x]\!]$, $n \in [\![1, N_t]\!]$ *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 connait 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

$$D\frac{\Delta_t}{dx^2} \leqslant \frac{1}{2}. \tag{4.110}$$

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

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

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

$$\frac{\boldsymbol{u}_i^n - \boldsymbol{u}_i^{n-1}}{\Delta_t} - D\frac{\boldsymbol{u}_{i+1}^n - 2\boldsymbol{u}_i^n + \boldsymbol{u}_{i-1}^n}{\Delta_x^2} = \boldsymbol{f}_i^n, \tag{4.111}$$

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

(4.114) est équivalent à

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

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

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

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

$$C\boldsymbol{u}_i^n - E(\boldsymbol{u}_{i+1}^n + \boldsymbol{u}_{i-1}^n) = \boldsymbol{u}_i^{n-1} + \Delta_t \boldsymbol{f}_i^n, \ \forall i \in]\!]0, N_x[\![\tag{4.114}$$

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

$$\mathbb{A}\boldsymbol{U}^n = \boldsymbol{b}^n \tag{4.116}$$

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

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

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

$$\left(\begin{array}{c|cccccc|c} 3 & -4 & 1 & 0 & \cdots & 0 & & 0 \\ \hline -E & C & -E & 0 & \cdots & 0 & & 0 \\ 0 & -E & C & -E & \ddots & \vdots & & \vdots \\ \vdots & 0 & \ddots & \ddots & \ddots & 0 & & \vdots \\ \vdots & \vdots & \ddots & -E & C & -E & & 0 \\ 0 & 0 & \dots & 0 & -E & C & & -E \\ \hline 0 & 0 & \dots & 0 & 0 & 0 & & 1 \end{array}\right) \left(\begin{array}{c} \boldsymbol{u}_0^n \\ \hline \boldsymbol{u}_1^n \\ \boldsymbol{u}_2^n \\ \vdots \\ \boldsymbol{u}_{N_x-2}^n \\ \boldsymbol{u}_{N_x-1}^n \\ \hline \boldsymbol{u}_{N_x}^n \end{array}\right) = \left(\begin{array}{c} 2\frac{\Delta_x}{D}\alpha(t^n) \\ \hline \boldsymbol{u}_1^{n-1} + \Delta_t \boldsymbol{f}_1^n \\ \boldsymbol{u}_2^{n-1} + \Delta_t \boldsymbol{f}_2^n \\ \vdots \\ \boldsymbol{u}_{N_x-2}^{n-1} + \Delta_t \boldsymbol{f}_{N_x-2}^n \\ \boldsymbol{u}_{N_x-1}^{n-1} + \Delta_t \boldsymbol{f}_{N_x-1}^n \\ \hline \beta(t^n) \end{array}\right) \tag{4.117}$$

L'algorithme formel est donc le suivant :

$\boldsymbol{U}_i^0 \leftarrow u_0(x_i), \ \forall i \in [\![0, N_x]\!]$
Pour $n \leftarrow 1$ **à** N_t **faire**
  Calcul de $\boldsymbol{U}^n$ en résolvant le système linéaire (4.117)
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*

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

où α, β, 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* $\boldsymbol{B} \in \mathbb{R}^d$ *défini par*

$$\boldsymbol{B} = \begin{pmatrix} \alpha \\ c_1 \\ \vdots \\ c_{d-2} \\ \beta \end{pmatrix} \tag{4.119}$$

où α, β, $c_1, \dots, c_{d-2}$ *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* $\boldsymbol{u}_i^n$, $i \in [\![0, N_x]\!]$, $n \in [\![1, N_t]\!]$ *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 connait la solution exacte.*