Heat Equation

Ania Polidori & Manal Derghal

Decembre 2023

Table des matières

1 Outils mathématiques

1.1 Approximations des dérivées

On peut approcher la dérivée partielle en temps par l'approximation décentrée à gauche :

$$\frac{\partial u}{\partial t}(t_j, x_i) \approx \frac{u(t_j, x_i) - u(t_{j-1}, x_i)}{\Delta t} \approx \frac{u_i^{(j)} - u_i^{(j-1)}}{\Delta t} \tag{1}$$

En ce qui concerne les dérivées secondes, on démontre facilement en utilisant la formule de Taylor à l'ordre 2 l'approximation :

$$\frac{\partial^2 u}{\partial^2 x}(t_j, x_i) \approx \frac{u(t_j, x_{i-1}) - 2u(t_j, x_i) + u(t_j, x_{i+1})}{\Delta x^2} \approx \frac{u_{i-1}^j - 2u_i^j + u_{i+1}^j}{\Delta x^2} \tag{2}$$

1.2 Algorithme de Thomas

Le step forward consiste à calculer les nouveaux coefficients comme suit, en notant les nouveaux coefficients avec des primes :

$$c'_i = \begin{cases} \dfrac{c_i}{b_i}, & i = 1, \\ \dfrac{c_i}{b_i - a_i c'_{i-1}}, & i = 2, 3, \ldots, n-1 \end{cases}$$

et

$$d'_i = \begin{cases} \dfrac{d_i}{b_i}, & i = 1, \\ \dfrac{d_i - a_i d'_{i-1}}{b_i - a_i c'_{i-1}}, & i = 2, 3, \ldots, n \end{cases}$$

La solution est ensuite obtenue par substitution arrière :

$$x_n = d'_n$$

et

$$x_i = d'_i - c'_i x_{i+1}, \quad i = n-1, n-2, \ldots, 1.$$

La méthode ci-dessus ne modifie pas les vecteurs de coefficients d'origine, mais doit également garder une trace des nouveaux coefficients. Si les vecteurs de coefficients peuvent être modifiés, alors un algorithme avec moins de comptabilité est :

Pour $i = 2, 3, \ldots, n$, faire

$$w = \frac{a_i}{b_{i-1}}, \quad b_i = b_i - wc_{i-1}, \quad d_i = d_i - wd_{i-1}$$

suivi de la substitution arrière

$$x_n = \frac{d_n}{b_n}, \quad x_i = \frac{d_i - c_i x_{i+1}}{b_i} \quad \text{pour } i = n-1, n-2, \ldots, 1.$$

2 Premier cas : la barre

On considère le problème aux limites :

$$\begin{cases} \frac{\partial u}{\partial t} = \frac{\lambda}{\rho c}\frac{\partial^2 u}{\partial x^2} + \frac{F}{\rho c}, & \forall (t,x) \in [0, t_{\max}] \times [0, L], \\ u(0,x) = u_0, & \forall x \in [0, L], \\ \frac{\partial u}{\partial x}(t,0) = 0, & \forall t \in [0, t_{\max}], \\ u(t,L) = u_0, & \forall t \in [0, t_{\max}]. \end{cases} \tag{3}$$

2.1 Approximation de (1) par différences finies (méthode implicite)

Soit $N \in \mathbb{N}$ fixé. On définit les points de discrétisation du maillage par : $x_i = ih, \quad i \in \{0, 1, \ldots, N+1\}$, où $h = \frac{L}{N+1}$.

De même $t_i = iT, \quad i \in \{0, 1, \ldots, N+1\}$, où $T = \frac{t_{\max}}{N+1}$.
De plus,

$$\frac{\partial u}{\partial t}(t,0) = \frac{F}{\rho c} \tag{4}$$

Donc $u(t,0) = \frac{F}{\rho c}t + \text{const}$ et $u(0,x) = u_0$, $F(0) = 0$ d'où $u(t,0) = u_0$

$$\begin{cases} \frac{u_i^{(j)} - u_i^{(j-1)}}{T} = \frac{\lambda}{\rho c}\frac{u_{i-1}^{(j)} - 2u_i^{(j)} + u_{i+1}^{(j)}}{h^2} + \frac{F}{\rho c}, & \text{pour } (j,i) \in [1, N+1] \times [1, N], \\ u_i^{(0)} = u_0, & \text{pour } i \in [1, N], \\ u_0^{(j)} = u_0, & \text{pour } j \in [1, N+1], \\ u_{N+1}^{(j)} = u_0, & \text{pour } j \in [1, N+1]. \end{cases} \tag{5}$$

$$\begin{cases} u_i^{(j-1)} = u_i^{(j)} + T(\frac{\lambda}{h^2 \rho c}(-u_{i-1}^{(j)} + 2u_i^{(j)} - u_{i+1}^{(j)}) - \frac{F}{\rho c}), & \text{pour } (j,i) \in [1, N+1] \times [1, N], \\ u_i^{(0)} = u_0, & \text{pour } i \in [1, N], \\ u_0^{(j)} = u_0, & \text{pour } j \in [1, N+1], \\ u_{N+1}^{(j)} = u_0, & \text{pour } j \in [1, N+1]. \end{cases} \tag{6}$$

On pose $\boxed{s = \frac{\lambda T}{\rho c h^2}}$

$$\begin{cases} u_i^{(j-1)} = u_i^{(j)}(1+2s) - s u_{i-1}^{(j)} - s u_{i+1}^{(j)} - T\frac{F}{\rho c}, & \text{pour } (j,i) \in [1, N+1] \times [1, N], \\ u_i^{(0)} = u_0, & \text{pour } i \in [1, N], \\ u_0^{(j)} = u_0, & \text{pour } j \in [1, N+1], \\ u_{N+1}^{(j)} = u_0, & \text{pour } j \in [1, N+1]. \end{cases} \tag{7}$$

Avec la condition pour j = 1 (initialization t=0) :

$$\boxed{\begin{bmatrix} u_1^{(0)} \\ u_2^{(0)} \\ \vdots \\ u_N^{(0)} \end{bmatrix} = \begin{bmatrix} u_0 \\ u_0 \\ \vdots \\ u_0 \end{bmatrix}}$$

Sinon :

$$\begin{bmatrix} u_1^{(j-1)} \\ u_2^{(j-1)} \\ \vdots \\ u_{N-1}^{(j-1)} \\ u_N^{(j-1)} \end{bmatrix} = \begin{bmatrix} 1+2s & -s & 0 & \cdots & 0 \\ -s & 1+2s & -s & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & -s & 1+2s & -s \\ 0 & \cdots & 0 & -s & 1+2s \end{bmatrix} \begin{bmatrix} u_1^{(j)} \\ u_2^{(j)} \\ \vdots \\ u_{N-1}^{(j)} \\ u_N^{(j)} \end{bmatrix} - \begin{bmatrix} \frac{TF}{\rho c} + su_0 \\ \frac{TF}{\rho c} \\ \vdots \\ \frac{TF}{\rho c} \\ \frac{TF}{\rho c} + su_0 \end{bmatrix} \tag{8}$$

$$\boxed{\begin{bmatrix} 1+2s & -s & 0 & \cdots & 0 \\ -s & 1+2s & -s & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & -s & 1+2s & -s \\ 0 & \cdots & 0 & -s & 1+2s \end{bmatrix} \begin{bmatrix} u_1^{(j)} \\ u_2^{(j)} \\ \vdots \\ u_{N-1}^{(j)} \\ u_N^{(j)} \end{bmatrix} = \begin{bmatrix} u_1^{(j-1)} + T\frac{F}{\rho c} + su_0 \\ u_2^{(j-1)} + T\frac{F}{\rho c} \\ \vdots \\ u_{N-1}^{(j-1)} + T\frac{F}{\rho c} \\ u_N^{(j-1)} + T\frac{F}{\rho c} + su_0 \end{bmatrix}}$$

On peut résoudre ceci avec l'algorithme de Thomas puisqu'on a une matrice tridiagonale.
On devra résoudre ce système autant de fois qu'il y a d'itérations en temps.

3 Deuxième cas : la plaque

$$\begin{cases} \frac{\partial u}{\partial t} = \frac{\lambda}{\rho c}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) + \frac{F}{\rho c}, & \forall (t,x,y) \in [0, t_{\max}] \times [0, L]^2, \\ u(0,x,y) = u_0, & \forall (x,y) \in [0, L]^2, \\ \frac{\partial u}{\partial x}(t,0,y) = 0, & \forall (t,y) \in [0, t_{\max}] \times [0, L], \\ \frac{\partial u}{\partial y}(t,x,0) = 0, & \forall (t,x) \in [0, t_{\max}] \times [0, L], \\ u(t,L,y) = u_0, & \forall (t,y) \in [0, t_{\max}] \times [0, L], \\ u(t,x,L) = u_0, & \forall (t,x) \in [0, t_{\max}] \times [0, L]. \end{cases} \tag{9}$$

3.1 Approximation de (8) par différences finies (méthode implicite)

Soit $N \in \mathbb{N}$ fixé. On définit les points de discrétisation du maillage par :
$x_i = ih$, $y_i = ih$, $\quad i \in \{0, 1, \ldots, N+1\}$ où $h = \frac{L}{N+1}$.
De même $t_i = iT$, $\quad i \in \{0, 1, \ldots, N+1\}$, où $T = \frac{t_{\max}}{N+1}$.

$$\begin{cases} \frac{u_{i,k}^{(j)} - u_{i,k}^{(j-1)}}{T} = \frac{\lambda}{\rho c}\left(\frac{u_{i-1,k}^{(j)} - 2u_{i,k}^{(j)} + u_{i+1,k}^{(j)}}{h^2} + \frac{u_{i,k-1}^{(j)} - 2u_{i,k}^{(j)} + u_{i,k+1}^{(j)}}{h^2}\right) + \frac{F}{\rho c}, & \forall (j,i,k) \in [1, N+1] \times [1, N]^2, \\ u_{i,k}^{(0)} = u_0, & \forall (i,k) \in [1, N]^2, \\ u_{0,k}^{(j)} = u_0, & \forall (j,k) \in [1, N+1] \times [1, N], \\ u_{i,0}^{(j)} = u_0, & \forall (j,i) \in [1, N+1] \times [1, N], \\ u_{N+1,k}^{(j)} = u_0, & \forall (j,k) \in [1, N+1] \times [1, N], \\ u_{i,N+1}^{(j)} = u_0, & \forall (j,i) \in [1, N+1] \times [1, N]. \end{cases} \tag{10}$$

$$\begin{cases} u_{i,k}^{(j-1)} = u_{i,k}^{(j)} + \frac{T\lambda}{\rho c h^2}\left(-u_{i-1,k}^{(j)} + 4u_{i,k}^{(j)} - u_{i+1,k}^{(j)} - u_{i,k-1}^{(j)} - u_{i,k+1}^{(j)}\right) - T\frac{F}{\rho c}, & \forall (j,i,k) \in [1, N+1] \times [1, N]^2, \\ u_{i,k}^{(0)} = u_0, & \forall (i,k) \in [1, N]^2, \\ u_{0,k}^{(j)} = u_0, & \forall (j,k) \in [1, N+1] \times [1, N], \\ u_{i,0}^{(j)} = u_0, & \forall (j,i) \in [1, N+1] \times [1, N], \\ u_{N+1,k}^{(j)} = u_0, & \forall (t,y) \in [1, N+1] \times [1, N], \\ u_{i,N+1}^{(j)} = u_0, & \forall (t,x) \in [1, N+1] \times [1, N]. \end{cases} \tag{11}$$

On pose $\boxed{s = \frac{\lambda T}{\rho c h^2}}$

$$\begin{cases} u_{i,k}^{(j-1)} = (1+4s)u_{i,k}^{(j)} - su_{i-1,k}^{(j)} - su_{i+1,k}^{(j)} - su_{i,k-1}^{(j)} - su_{i,k+1}^{(j)} - T\frac{F}{\rho c}, & \forall (j,i,k) \in [1, N+1] \times [1,N]^2, \\ u_{i,k}^{(0)} = u_0, & \forall (i,k) \in [1,N]^2, \\ u_{0,k}^{(j)} = u_0, & \forall (j,k) \in [1, N+1] \times [1,N], \\ u_{i,0}^{(j)} = u_0, & \forall (j,i) \in [1, N+1] \times [1,N], \\ u_{N+1,k}^{(j)} = u_0, & \forall (j,k) \in [1, N+1] \times [1,N], \\ u_{i,N+1}^{(j)} = u_0, & \forall (j,i) \in [1, N+1] \times [1,N]. \end{cases} \tag{12}$$

Avec la condition pour j=1 on a (initialisation à t=0) :

$$\boxed{\begin{bmatrix} U_{1,1}^0 \\ \vdots \\ U_{1,N}^0 \\ \cdots\cdots \\ U_{2,1}^0 \\ \vdots \\ U_{2,N}^0 \\ \cdots\cdots \\ \vdots \\ \cdots\cdots \\ U_{N,1}^0 \\ \vdots \\ U_{N,N}^0 \end{bmatrix} = \begin{bmatrix} u_0 \\ \vdots \\ u_0 \\ \cdots \\ u_0 \\ \vdots \\ u_0 \\ \cdots \\ \vdots \\ \cdots \\ u_0 \\ \vdots \\ u_0 \end{bmatrix}}$$

Sinon :

$$\begin{bmatrix} A & -sI & 0 & \cdots & 0 \\ -sI & A & -sI & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & -sI & A & -sI \\ 0 & \cdots & 0 & -sI & A \end{bmatrix} \begin{bmatrix} U_{1,1}^{j} \\ \vdots \\ U_{1,N}^{j} \\ \cdots\cdots \\ U_{2,1}^{j} \\ \vdots \\ U_{2,N}^{j} \\ \cdots\cdots \\ \vdots \\ \cdots\cdots \\ U_{N,1}^{j} \\ \vdots \\ U_{N,N}^{j} \end{bmatrix} = \begin{bmatrix} U_{1,1}^{j-1} + T\frac{F}{\rho c} + 2su_0 \\ U_{1,2}^{j-1} + T\frac{F}{\rho c} + su_0 \\ \vdots \\ U_{1,N-1}^{j-1} + T\frac{F}{\rho c} + su_0 \\ U_{1,N}^{j-1} + T\frac{F}{\rho c} + 2su_0 \\ \cdots\cdots\cdots\cdots\cdots \\ U_{2,1}^{j-1} + T\frac{F}{\rho c} + su_0 \\ U_{2,2}^{j-1} + T\frac{F}{\rho c} \\ \vdots \\ U_{2,N-1}^{j-1} + T\frac{F}{\rho c} \\ U_{2,N}^{j-1} + T\frac{F}{\rho c} + su_0 \\ \cdots\cdots\cdots\cdots\cdots \\ U_{3,1}^{j-1} + T\frac{F}{\rho c} + su_0 \\ U_{3,2}^{j-1} + T\frac{F}{\rho c} \\ \vdots \\ U_{3,N-1}^{j-1} + T\frac{F}{\rho c} \\ U_{3,N}^{j-1} + T\frac{F}{\rho c} + su_0 \\ \cdots\cdots\cdots\cdots\cdots \\ \vdots \\ \cdots\cdots\cdots\cdots\cdots \\ U_{N,1}^{j-1} + T\frac{F}{\rho c} + 2su_0 \\ U_{N,2}^{j-1} + T\frac{F}{\rho c} + su_0 \\ \vdots \\ U_{N,N-1}^{j-1} + T\frac{F}{\rho c} + su_0 \\ U_{N,N}^{j-1} + T\frac{F}{\rho c} + 2su_0 \end{bmatrix}$$

Avec

$$A = \begin{bmatrix} 1+4s & -s & 0 & \cdots & 0 \\ -s & 1+4s & -s & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & -s & 1+4s & -s \\ 0 & \cdots & 0 & -s & 1+4s \end{bmatrix}$$

4 Animation

La fonction `animationBarre` et `animationPlaque` visent à visualiser graphiquement l'évolution de la température le long d'une barre ou d'une plaque. Prenons le cas de la barre. Cette animation s'appuie sur une matrice `solution`, représentant les températures à différents emplacements de la barre et à différents instants.

4.1 Éléments Graphiques

La fonction commence par configurer les propriétés graphiques d'un rectangle, qui servira à représenter la barre de température. Ce rectangle a une largeur fixe de 1 unité et une hauteur de 50 unités. La position verticale (`rect.y`) est fixée à 250 unités, correspondant à la position de base sur l'axe des ordonnées.

4.2 Calcul des Températures Maximales

Ensuite, la fonction identifie la température maximale pour chaque position le long de la barre, stockant ces valeurs dans un vecteur appelé `maxS`. Elle détermine également la température maximale globale, que nous appellerons `maxSolution`.

4.3 Animation Visuelle

La partie principale de la fonction consiste à associer des couleurs aux températures pour chaque position de la barre. Plus la température est élevée, plus la couleur tend vers le rouge. Cela est réalisé en ajustant les composantes Rouge (`r`), Verte (`g`), et Bleue (`b`) en fonction de la température normalisée.

4.4 Rafraîchissement Visuel

L'animation est affichée à l'aide de la bibliothèque SDL. Des pauses entre chaque image (`SDL_Delay`) sont introduites pour permettre une perception visuelle appropriée.

5 Difficultés

L'implémentation de l'algorithme de Thomas pour résoudre le système tridiagonal par bloc dans le contexte de la modélisation de la plaque thermique a été confrontée à des défis significatifs. Contrairement au cas de la barre, où la complexité de l'algorithme est linéaire en fonction de la taille du système (O(n)), la résolution de la matrice tridiagonale par bloc dans le cas de la plaque présente une complexité beaucoup plus élevée. L'utilisation répétée d'inversions de matrices nécessaires à la résolution du système entraîne une complexité trop grande, rendant la résolution du système prohibitivement coûteuse pour des discrétisations plus élevées (n=1001, par exemple). En conséquence, le choix a été fait de restreindre la résolution aux discrétisations plus basses (n=10), permettant de réduire considérablement la charge computationnelle. Cette limitation impacte la finesse de la discrétisation spatiale, mais s'avère nécessaire pour maintenir une performance calculatoire raisonnable dans le cadre de la simulation thermique de la plaque.

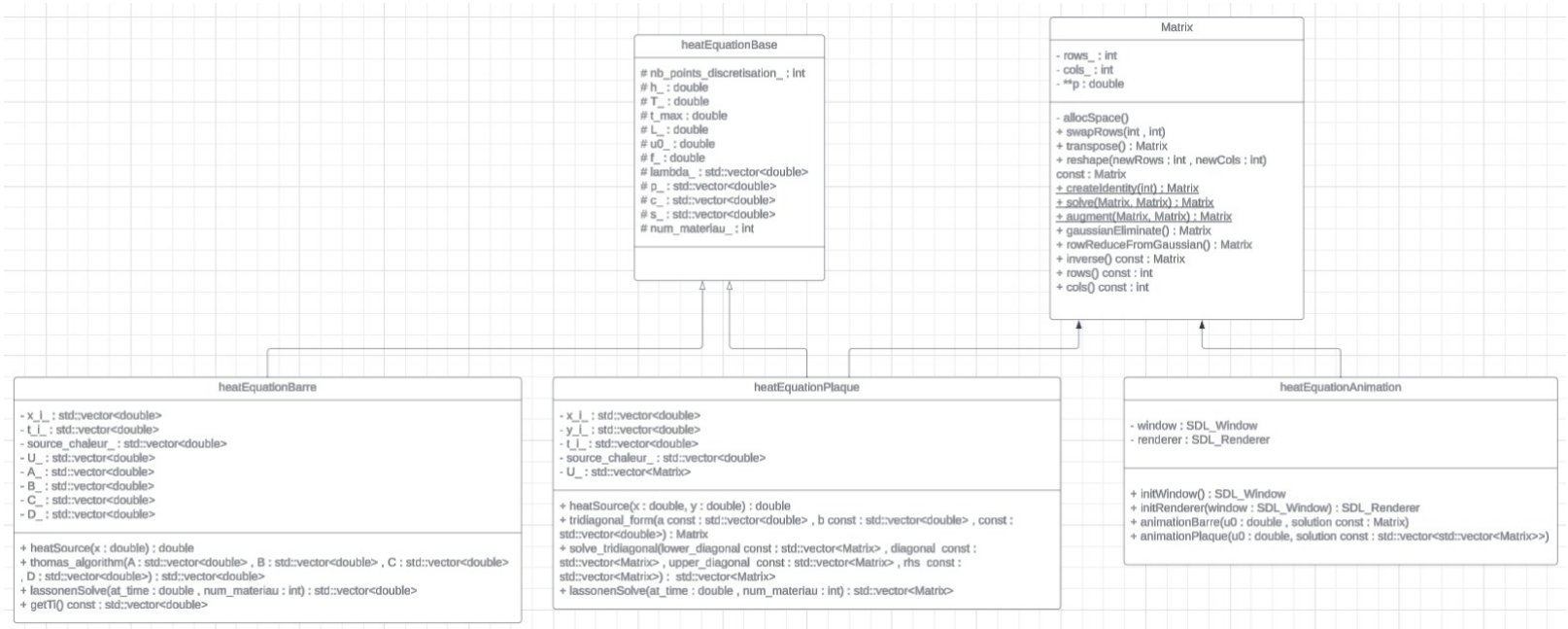

FIGURE 1 – Diagramme UML.