**Analyse matricielle appliquée**
Année : 2025-2026
Formation : L3 Ingénieurie Mathématique

# Projet : Résolution numérique de l'équation de la chaleur avec évolution temporelle

On considère l'équation aux dérivées partielles

$$\frac{\partial u}{\partial t}(t,x) - \mu \frac{\partial^2 u}{\partial x^2}(t,x) = 0 \text{ pour } x \in ]0,1[ \text{ et } t > 0. \tag{1}$$

$\mu$ désigne le coefficient de diffusion. Les conditions aux limites sont

$$u(t,0) = u(t,1) = 0 \text{ pour tout } t > 0,$$

et la condition initiale est

$$u(0,x) = \exp(-4096(x-1/2)^2) \quad \forall x \in ]0,1[. \tag{2}$$

On construit un maillage du domaine $]0,1[$ par $N+1$ segments de taille $h = \frac{1}{N+1}$. Les $N+2$ points sont

$$x_i = i\,h \quad \forall\, 0 \leq i \leq N+1.$$

On choisira un pas de temps

$$\Delta t = \alpha \frac{\Delta x^2}{2\mu},$$

$\alpha$ étant un paramètre à fixer.

On considère la discrétisation de ce problème :

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} - \mu \frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{\Delta x^2} = 0, \quad \forall\, n \geq 0, \quad \forall\, 1 \leq i \leq N, \tag{3}$$

avec $u_i^n$ l'approximation de $u(n\,\Delta t, x_i)$ qui est la solution exacte en $x = x_i$, $t = n\,\Delta t$. Cette discrétisation peut se ré-écrire

$$u_i^{n+1} - \Delta t\,\mu \frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{\Delta x^2} = u_i^n, \quad \forall\, n \geq 0, \quad \forall\, 1 \leq i \leq N, \tag{4}$$

On note de la manière suivante le vecteur $U^n$ des inconnues à l'instant $t = n\,\Delta t$ :

$$U^n = \begin{pmatrix} u_0^n \\ u_1^n \\ \vdots \\ u_{N+1}^n \end{pmatrix}.$$

La formule (4) peut donc se ré-écrire sous la forme :

$$U^{n+1} - \Delta t\, A U^{n+1} = U^n, \quad \forall\, n \geq 0.$$

avec $A$ une matrice.

# 1 Partie théorique

**1.** Ecrire la forme de la matrice $A$.

**2.** Montrer que la matrice $I - \Delta t A$ est inversible (en supposant si besoin que $\Delta t$ est suffisamment petit).

**3.** Montrer que la matrice $I - \Delta t A$ est symétrique et définie positive (en supposant si besoin que $\Delta t$ est suffisamment petit).

**4.** Vérifier que dans le cas de la condition initiale (2), la fonction $u$ définie ci-dessous est bien solution de l'équation (1) :

$$u(t,x) = \frac{1}{\sqrt{1+16384t\mu}} \exp(-\frac{4096(x-1/2)^2}{1+16384t\mu}) \quad \forall x \in ]0,1[. \tag{5}$$

**5.** Proposer un stockage de cette matrice qui permet de représenter la matrice avec une structure de données comprenant exactement 7 valeurs.

# 2 Partie implémentation et tests numériques

Dans la suite on prend $\mu = 0.3$.

La programmation se fera en C++.

**1.** Vous programmerez les algorithmes du gradient à pas simple, à pas optimal, et gradient conjugué en utilisant le stockage de matrice précédemment défini, ainsi que le stockage classique qui contient la matrice entière.

**2.** Vous programmerez le calcul de l'erreur entre solution exacte et solution approchée, en norme du max pour un temps final $T$f.

**3.** Vous stockerez les résultats obtenus pour un temps final $T$f (que vous choisirez) dans un fichier de manière à pouvoir les afficher, par exemple avec le logiciel gnuplot (voir ce lien pour une introduction à ce logiciel).

**4.** Vous testerez ces algorithmes pour résoudre le problème de la chaleur instationnaire, avec les conditions initiales et la condition aux limites présentées ci-dessus, pour différentes valeurs de $\alpha$. Vous vérifierez que votre solution numérique converge bien vers la solution exacte.

**5.** Pour cette question et la suivante, choisissez une seule valeur de $\alpha$ qui vous a permis d'obtenir une solution numérique satisfaisante. Comparez les temps de calcul des trois algorithmes, dans le cas du stockage creux que vous aurez créé, pour $N = 32 \times 2^i$ pour $i = 0, 1, 2, \cdots$. Vous argumenterez avec des exemples que vous aurez choisis.

**6.** Vous présenterez les affichages graphiques correspondant aux solutions obtenues avec $N = 32 \times 2^i$ pour $i = 0, 1, 2, \cdots$, en commentant sur les valeurs de $i$ qui sont atteignables sur votre ordinateur pour les deux stockages. Quel est le facteur le plus limitant dans les deux cas : temps de calcul ou place mémoire ? Vous argumenterez avec des exemples que vous aurez choisis.

**7.** Pour obtenir une précision en terme d'erreur au temps final de $10^{-5}$, quel est le meilleur couple de paramètres $(\alpha, N)$ ? Vous argumenterez votre réponse.