Master 1 Mathématiques et Applications
U.E. ANAC++, année 2025–2026

FACULTÉ DES SCIENCES ET TECHNOLOGIES
Département Mathématiques

Université de Lille

ANAC++P2, TP 3 : Méthodes itératives pour les matrices creuses : application à la température optimale d'un four pour la cuisson d'un gâteau

Le travail sera à faire en binôme. Il sera considéré pour la note Projet

Les programmes en C++, à rendre sur la plateforme Moodle, doivent être accompagnés d'un compte-rendu répondant aux questions théoriques et faisant une analyse bien soignée des résultats.

On étudie dans ce projet le champ de température d'un gâteau à cuire dans un four à partir de la valeur connue des résistences électriques. On suppose le phénomène stationnaire, c'est-à-dire indépendant du temps (c'est le cas quand le four est arrivé à la température imposée avec le gâteau à cuir déjà à l'intérieur du four). Le but du projet est la résolution par diverses méthodes itératives du système linéaire $Ax = b$ qui provient de la discrétisation par différences finies des équations definissant notre problème.

Le four est représenté par le domaine $\Omega =]0,1[^2$ de frontière $\partial\Omega$, avec n le vecteur normale extérieure unitaire. On considère la frontière décomposée en deux parties d'intersection vide : $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$, avec $\partial\Omega_D$ de mesure non nulle. Le gâteau est représenté par le domaine $G = [0.2, 0.8] \times [0.3, 0.4]$.

Donnée la fonction $v = (v_1, v_2) : \Omega \to \mathbb{R}$, on définit l'opérateur divergence par $div\, v = \dfrac{\partial v_1}{\partial x_1} + \dfrac{\partial v_2}{\partial x_2}$. Alors, on écrit le problème de diffusion de la chaleur sous la forme suivante : trouver la fonction température $T : \Omega \to \mathbb{R}$ de classe $\mathcal{C}^4(\Omega)$ telle que

$$\begin{cases} -div\,(\rho\,\nabla T) = 0 & \text{dans } \Omega, \\ T = T_D & \text{sur } \partial\Omega_D, \\ \nabla T \cdot n = 0 & \text{sur } \partial\Omega_N. \end{cases} \tag{1}$$

On a noté $\rho : \Omega \to \mathbb{R}^{*,+}$ le coefficient de conductivité thermique et T_D la température fixée sur le bord $\partial\Omega_D$ (par exemple $T_D = 180$ degrés sur le bord supérieur du four et $T_D = 20$ degrés sur le bord inférieur.) Les conditions aux limites de Neumann sont nulles car on considère une isolation thermique parfaite du four.

Afin de discrétiser le problème par la méthode des différences finies, on introduit un maillage structuré de $\bar{\Omega}$, avec $N \in \mathbb{N}^*$ et le pas $h = 1/(N+1)$ dans chaque direction. On définit $x_{i,j} = (ih, jh)$ les points du maillage pour $i, j = 0, ..., N+1$ et on cherche une approximation $T_{i,j}$ de la température $T(x_{i,j})$.

Première partie : la discrétisation par différences finies

(1.1) On considère le coefficient de conductivité $\rho(x_1, x_2) = 1$ pour tout $(x_1, x_2) \in \Omega$. Montrer que pour $T \in \mathcal{C}^4(\Omega)$, si h est petit alors le système d'équations s'écrit :

$$\begin{cases} \dfrac{4T_{i,j} - T_{i-1,j} - T_{i+1,j} - T_{i,j-1} - T_{i,j+1}}{h^2} = 0, & \forall\, 1 \le i,j \le N, \\ T_{i,0} = 20, \quad T_{i,N+1} = 180, & \forall\, i = 0, \dots N+1 \quad \text{(C.L. de Dirichlet sur } \partial\Omega_D), \\ T_{-1,j} = T_{1,j}, \quad T_{N+2,j} = T_{N,j}, & \forall\, j = 1, \dots N, \quad \text{(C.L. de Neumann sur } \partial\Omega_N). \end{cases}$$

(1.2) Montrer que pour $T \in \mathcal{C}^4(\Omega)$ et $\rho \in \mathcal{C}^2(\Omega)$, si h est petit, on a :

$$\frac{[\rho(x_1 + \frac{h}{2}, x_2) + \rho(x_1 - \frac{h}{2}, x_2)]T(x_1, x_2) - \rho(x_1 + \frac{h}{2}, x_2)T(x_1 + h, x_2) - \rho(x_1 - \frac{h}{2}, x_2)T(x_1 - h, x_2)}{h^2}$$

$$= -\frac{\partial}{\partial x_1}\Big(\rho \frac{\partial T}{\partial x_1}\Big)(x_1, x_2) + \mathcal{O}(h^2).$$

En inversant les rôles de x_1 et x_2, nous obtenons une différence finie similaire pour l'autre dérivée partielle dans (1). De plus, on note $x_{i\pm\frac{1}{2},j} = (ih \pm \frac{h}{2}, jh)$, $x_{i,j\pm\frac{1}{2}} = (ih, jh \pm \frac{h}{2})$ les points intermédiaires du maillage.

Vérifier que pour tout entier $N > 1$, on obtient le système d'équations linéaires suivant :

$$\begin{cases} \dfrac{\widetilde{\rho}_{i,j} T_{i,j} - \rho_{i+\frac{1}{2},j} T_{i+1,j} - \rho_{i-\frac{1}{2},j} T_{i-1,j} - \rho_{i,j+\frac{1}{2}} T_{i,j+1} - \rho_{i,j-\frac{1}{2}} T_{i,j-1}}{h^2} = 0, & \forall\, 1 \le i,j \le N, \\ T_{i,0} = 20, \quad T_{i,N+1} = 180, & \forall\, i = 0, \dots N+1, \\ T_{-1,j} = T_{1,j}, \quad T_{N+2,j} = T_{N,j}, & \forall\, j = 1, \dots N, \end{cases}$$

avec

$$\widetilde{\rho}_{i,j} = \rho_{i+\frac{1}{2},j} + \rho_{i-\frac{1}{2},j} + \rho_{i,j+\frac{1}{2}} + \rho_{i,j-\frac{1}{2}}.$$

De plus, on considère un prolongement symétrique de ρ en dehors de Ω à cause des C.L. de Neumann :

$$\rho_{-\frac{1}{2},j} = \rho_{\frac{1}{2},j}, \quad \rho_{N+\frac{3}{2},j} = \rho_{N+\frac{1}{2},j} \quad \text{pour } j = 1, \dots N.$$

Pour l'application du gâteau G dans le four on prendra : $\rho = 100$ si $(x_1, x_2) \in G$ et $\rho = 1$ si $(x_1, x_2) \in \Omega \setminus G$.

(1.3) Nous allons numéroter les inconnues (et les équations) suivant l'ordre lexicographique

$$T_{0,0}, T_{1,0}, \dots, T_{N+1,0}, T_{0,1}, T_{1,1}, \dots, T_{N+1,1}, \dots$$

Pour $N = 3$, écrire la matrice de coefficients A et le second membre b correspondant. La matrice A est-elle symétrique ? Sinon, que faut-il faire pour la rendre symétrique ? Montrer aussi que A est définie positive. Montrer que seulement les premiers et derniers $N+2$ éléments de b sont non nuls.

(1.4) Constituer la classe `matricebande` qui hérite les champs et les méthodes de la classe `matrice` et qui mettra en place le stockage bande de A. On rappelle que pour A symétrique, on peut stocker juste les 3 diagonales non nulles de sa partie triangulaire inférieure à l'aide d'une matrice rectangulaire `ATAB` et d'un vecteur d'entiers `IND` qui donne la position des diagonales de A. La classe `matricebande` ajoutera le champ `IND` aux champs de la classe mère `matrice` et plusieurs fonctions membres décrites dans les questions suivantes.

(1.5) Ecrire une fonction membre de la classe `matricebande` nommée `laplacien` qui prend en entrée un entier $N > 1$ et remplit les tableaux `ATAB` et `IND` correspondants à la partie triangulaire inférieure de A correspondante à la question (1.1).

(1.6) Ecrire une fonction membre de la classe `matricebande` nommée `laplacien_rho` qui prend en entrée un entier $N > 1$ et remplit les tableaux `ATAB` et `IND` correspondants à la partie triangulaire inférieure de A correspondante à la question (1.2).

(1.7) Ecrire une fonction membre de la classe `matricebande` qui surcharge l'opérateur * et qui prend en entrée un vecteur x (vérifier sa taille) afin de retourner le vecteur $y = Ax$ avec A symétrique donnée en stockage bande.

(1.8) On testera dans le programme principal l'assemblage de la matrice A et le produit Ax en affichant le résultat à écran (prendre par exemple $x = (1, \ldots, 1)^T$ avec N petit).

(1.9) Ecrire les fonctions amies de la classe `matricebande` nommées respectivement `smbr` et `smbr_rho` qui remplissent le second membre b correspondants respectivement aux questions (1.1) et (1.2).

Deuxième partie : la méthode de gradient à pas optimal et à pas fixe

Considérer le système linéaire donné dans le cas (1.1). Partant de $x_0 = 0$, $r_0 = b - Ax_0$, la méthode de gradient à pas optimal (steepest descent) est donnée par les récurrences

$$x_{k+1} = x_k + \alpha_k r_k, \quad r_{k+1} = r_k - \alpha_k A r_k, \quad \alpha_k = \frac{r_k^T r_k}{r_k^T A r_k}. \tag{2}$$

On propose d'arrêter les itérations quand $\|r_k\| \leq tol\, \|r_0\|$, avec $tol = 10^{-4}$, sachant que l'on ne souhaite pas effectuer plus que $10 \cdot size(x_0, 1)$ itérations. Aussi, on ne souhaite pas que la taille de $\|r_k\|$ augmente trop : on arrête également les itérations (avec un message d'erreur `OVERFLOW`) quand $\|r_k\| \geq \|r_0\|/tol$.

(2.1) Ecrire une fonction membre de la classe `matricebande` nommée `steepest_descent` pour la méthode de gradient à pas optimal, en prenant comme entrée le vecteur b et le paramètre tol, comme entrée-sortie le vecteur x (qui contient en entrée le vecteur x_0 et en sortie le dernier itéré x_k) ainsi qu'en sortie le nombre d'itérations effectuées pour atteindre la condition d'arrêt et un vecteur qui stocke les valeurs $\|r_k\|/\|b\|$ calculées à chaque itération k.

(2.2) Sachant que steepest descent donne lieu à un comportement en zig-zag (deux résidus consécutifs sont orthogonaux), on compare les performances de steepest descent avec la méthode de gradient à pas fixe α (paramètre d'entrée).

On sait que l'algorithme de Richardson donne $\alpha_{opt} = \dfrac{2}{\lambda_1 + \lambda_n}$ (voir feuille de TD) avec λ_1 (resp. λ_n) la plus petite (resp. grande) valeur propre de A. Dans notre cas, $1/\alpha_{opt}$ est approximativement donnée par le plus grand élément sur la diagonale de A.

Ecrire une fonction membre de la classe `matricebande` nommée `gradient_pas_fixe` réalisant cette nouvelle méthode. Tracer dans le rapport un tableau indiquant le nombre d'itérations pour atteindre la condition d'arrêt en fonction du choix du paramètre α, et le comparer aux nombre d'itérations du steepest descent. On choisira $\alpha = j\alpha_{opt}/5$ pour $j = 1, 2, \ldots, 6$.

Faire un graphique qui trace la fonction $k \to \|r_k\|/\|b\|$ sur une échelle semi-logarithmique pour les deux méthodes, avec les différents choix du α fixe. Commenter les résultats obtenus.

Troisième partie : la méthode PCG de gradient conjugué avec préconditionnement

Dans la méthode PCG, on résout itérativemet le système $Ax = b$ préconditionné par $C = TT^t$ avec une matrice T triangulaire inférieure. Partant de $x = x_0$, $r = r_0 = b - Ax$, $z = C^{-1}r$, $p = p_0 = z$, et $\gamma = \gamma_0 = z^T r$, on calcule récursivement tant que $\gamma \geq tol^2 \gamma_0$

$$q = Ap,\ \alpha = \frac{\gamma}{p^T q},\ x = x + \alpha p,\ r = r - \alpha q,\ z = C^{-1}r,\ \beta = \frac{1}{\gamma},\ \gamma = z^T r,\ \beta = \beta\gamma,\ p = z + \beta p.$$

On suppose que la matrice T triangulaire inférieure est stockée aussi au format bande à l'aide des tableaux `TTAB,TIND`. Nous allons discuter trois choix pour la matrice C :

1) la matrice identité $C = I$ et alors $T = I$;
2) la matrice diagonale $C = \mathrm{diag}(A)$ obtenue par la diagonale de A et alors $T = C^{1/2}$;
3) la matrice $IC(0)$ qui pour le schéma à 5 points du Laplacien se simplifie : on cherche $T = (T_{j,k})$ triangulaire inférieure avec des éléments non nuls aux mêmes positions que dans la partie triangulaire inférieure de A de sorte que $A + \omega \mathrm{diag}(A) - TT^t$ admet des zéros dans les positions correspondantes aux éléments non nuls de A. On peut montrer, en tenant compte de la structure particulière de A, que

$$i > j,\ A_{i,j} \neq 0 : \qquad (TT^t)_{i,j} = T_{i,j} T_{j,j}.$$

Par conséquent, les éléments non nuls de T sur chaque ligne i sont définis par

$$T_{i,j} = \frac{A_{i,j}}{T_{j,j}} \quad \text{pour } j < i \quad \text{et} \quad T_{i,i} = \sqrt{A_{i,i}(1+\omega) - \sum_{\ell=1}^{i-1} T_{i,\ell}^2}.$$

sachant que $T_{i,i}$=`TTAB(TIND(2))`.

(3.1) Ecrire une fonction membre de la classe `matricebande` nommée `assemblageT(choix,omega)` qui pour `choix` $\in \{1, 2, 3\}$ et un paramètre optionnel ω remplit les tableaux `TTAB,TIND` pour un des trois préconditionneurs ci-dessus.

(3.2) Ecrire une fonction membre de la classe `matricebande` nommée `preconditionne(r)` qui réalise une descente $Ty = r$ et une remontée $T^t z = y$.

(3.3) Ecrire une fonction membre de la classe `matricebande` nommée `PCG` prénant comme entrée le paramètre tol, comme entrée-sortie le vecteur x (en entrée le vecteur x_0 et en sortie le dernier itéré x_k), les tableaux nécessaires pour stocker A et T, ainsi qu'en sortie le nombre d'itérations effectuées pour atteindre la condition d'arrêt et un vecteur qui stocke les valeurs $\|r_k\|/\|b\|$ calculées à chaque itération k.

(3.4) Résoudre les systèmes linéaires donnés dans les parties (1.1) et (1.2), en prenant par exemple $N = 20$. Dans le rapport, faire un tableau qui indique le nombre d'itérations du PCG pour les trois préconditionneurs considérés. Faire un graphique qui trace la fonction $k \to \|r_k\|/\|b\|$ sur une échelle semi-logarithmique pour les PCG avec les différents préconditionneurs et différents choix du paramètre ω (si $A \in \mathcal{M}_n(\mathbb{R})$, prendre par exemple $\omega \in \{0, \frac{1}{n}\}$). Commenter les résultats obtenus.

BONUS : Quatrième partie : optimiser la cuisson du gâteau

Le problème résolu dans la section précédente s'appelle **problème direct** car les valeurs des résistances sont les données du problème et la température du gâteau en est l'inconnue. Or, puisque

la qualité du gâteau dépend de la températures idéale de cuisson, il faut déterminer les valeurs des résistances qui permettent de chauffer le gâteau à cette température. Il s'agit alors d'un **problème inverse** : la température idéale du gâteau est une donnée du problème et les valeurs des résistances sont les inconnues.

On remarque que le problème (1) est **linéaire**. Cela signifie que si T_1 est la solution du problème correspondant aux conditions aux bord $T_{D,1}$ et T_2 est celle correspondante aux conditions aux bord $T_{D,2}$, alors $T_1 + T_2$ est la solution du problème correspondant aux conditions aux bord $T_{D,1} + T_{D,2}$. Désormais $T_{D,1}$ correspondra à la résistance du four en bas et $T_{D,2}$ correspondra à la résistance du four en haut. Donc, la température T du four pour les 2 résistances chauffantes s'écrit :

$$T = \sum_{k=1}^{2} \alpha_k T_k, \tag{3}$$

où α_k est la valeur de la résistance k et T_k la température associée quand la résistance k est la seule à chauffer le four (l'autre sera donc nulle).

Pour obtenir la température idéale T_{opt} dans le gâteau, il faut déterminer donc les valeurs α_k des résistances. On les obtient par minimisation de la fonctionnelle

$$J(\alpha) = \sum_{(x,y)\in G} \Big(T_{opt}(x,y) - \sum_{k=1}^{2} \alpha_k T_k(x,y)\Big)^2 + \frac{1}{100} \sum_{k=1}^{2} (\alpha_k T_{D,k})^2$$

qui est strictement convexe en α. La fonctionnelle J mesure d'une part l'écart entre la température optimale souhaitée et la température obtenue dans le gâteau, et d'autre part tient compte de l'énergie dépensée pour chauffer le four, afin de minimiser la consommation électrique.

La fonctionnelle J atteint son unique minimum pour la valeur α qui annule son gradient. La composante k du vecteur gradient est

$$\frac{\partial J}{\partial \alpha_k} = 2 \sum_{(x,y)\in G} \Big(T_{opt}(x,y) - \sum_{j=1}^{2} \alpha_j T_j(x,y)\Big) T_k(x,y) + \frac{1}{50} \sum_{k=1}^{2} \alpha_k T_{D,k}^2.$$

Définissons la matrice $A \in \mathcal{M}_2(\mathbb{R})$ et le second membre $b \in \mathbb{R}^2$ par

$$A_{k,j} = \sum_{(x,y)\in G} T_k(x,y) T_j(x,y) + \frac{1}{100} T_{D,k}^2 \delta_{k,j} \qquad \text{et} \qquad b_k = \sum_{(x,y)\in G} T_k(x,y) T_{opt}(x,y). \tag{4}$$

La valeur optimale des résistances est donc obtenue comme solution du système linéaire $A\alpha = b$.

Afin de résoudre le problème inverse et obtenir la température T proche de $T_{opt} = 200$ degrés, programmer les étapes suivantes (prendre par exemple N=20) :

1. Résoudre le problème direct qui permet de calculer T_1. On utilisera la meilleure procédure mise au point dans la section précédente, en prenant les données sur les bords Dirichlet
$$T_{i,0} = 200, \quad T_{i,N+1} = 0, \quad \forall\, i = 0, \dots N+1.$$
2. Résoudre le problème direct qui permet de calculer T_2. On utilisera la meilleure procédure mise au point dans la section précédente, en prenant les données sur les bords Dirichlet
$$T_{i,0} = 0, \quad T_{i,N+1} = 200, \quad \forall\, i = 0, \dots N+1.$$
3. Construire pour le problème inverse les coefficients $A_{k,j}$ et b_k du système linéaire $A\alpha = b$.
4. Calculer la solution $\alpha \in \mathbb{R}^2$ puis calculer la solution T grâce à la formule (3). Visualiser la solution T obtenue (tracer par exemple les lignes de niveau).