Université de Versailles Saint-Quentin-en-Yvelines
Université Paris-Saclay

Rapport de TP

Méthodes directes et itératives pour l'équation de la chaleur 1D (Poisson 1D)

Auteur : Hao QIAN

Encadrants : T. Dufaud, J. Gurhem

3 janvier 2026

Table des matières

1 Introduction

Ce TP vise à implémenter et comparer plusieurs méthodes numériques (directes et itératives) pour résoudre un système linéaire issu de la discrétisation par différences finies de l'équation stationnaire de la chaleur en 1D (problème de Poisson 1D). Les développements sont réalisés en langage C en s'appuyant sur les bibliothèques **BLAS** et **LAPACK**.

L'analyse porte sur :

— la validation (solution analytique / résidu relatif) ;
— la convergence des méthodes itératives (Richardson, Jacobi, Gauss–Seidel) ;
— les performances : temps d'exécution en fonction de la taille n.

2 Problème continu et solution analytique

On considère l'équation stationnaire sur l'intervalle $(0, 1)$:

$$-k\frac{d^2T}{dx^2}(x) = g(x), \quad x \in (0,1), \tag{1}$$

avec conditions de Dirichlet :

$$T(0) = T_0, \qquad T(1) = T_1. \tag{2}$$

Dans la suite, on se place dans le cas **sans source** $g(x) = 0$. La solution analytique est alors linéaire :

$$T(x) = T_0 + x\,(T_1 - T_0). \tag{3}$$

Principe de validation. On compare la solution numérique x_{num} à la solution exacte x_{exact} évaluée sur la grille. On calcule en particulier l'erreur relative avant :

$$\text{err} = \frac{\|x_{\text{num}} - x_{\text{exact}}\|_2}{\|x_{\text{exact}}\|_2}, \tag{4}$$

ainsi que le résidu relatif $\|r\|_2/\|b\|_2$.

3 Discrétisation par différences finies

On discrétise $[0, 1]$ en $n + 2$ nœuds $x_i = ih$ avec $i = 0, \ldots, n + 1$ et :

$$h = \frac{1}{n+1}. \tag{5}$$

La dérivée seconde est approchée par un schéma centré d'ordre 2 :

$$\left.\frac{d^2T}{dx^2}\right|_{x_i} \approx \frac{T_{i-1} - 2T_i + T_{i+1}}{h^2}. \tag{6}$$

Dans le cas $g = 0$, on obtient un système linéaire $Au = b$ de taille n, où $u = (T_1, \ldots, T_n)^T$ et A est tridiagonale.

Une écriture classique est :

$$A = \begin{pmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 2 & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{pmatrix}. \tag{7}$$

Remarque : selon la mise à l'échelle choisie, A peut représenter $\text{tridiag}(-1, 2, -1)$ ou $\frac{k}{h^2}\text{tridiag}(-1, 2, -1)$, le second membre b étant ajusté de manière cohérente.

4 Stockage bande (GB) et rappels BLAS/LAPACK

4.1 Stockage General Band (GB)

Pour une matrice bande de paramètres (kl, ku), le format GB de LAPACK stocke la matrice dans un tableau de taille $(kl + ku + 1) \times n$ en ordre colonne (`COL_MAJOR`). L'élément a_{ij} est rangé à :

$$AB(ku + 1 + i - j,\ j). \tag{8}$$

Dans notre cas (Poisson 1D), la matrice est tridiagonale : $kl = ku = 1$.

4.2 Routines utilisées

— `dgbmv` : produit matrice-vecteur en format bande (SpMV bande).
— `dgbtrf` : factorisation LU (avec pivotage).
— `dgbtrs` : résolution triangulaire après factorisation.
— `dgbsv` : driver LAPACK (factorisation + solve).

5 Méthodes directes (EX3–EX6)

5.1 Résolution via LAPACK (EX5)

La méthode directe utilise les routines LAPACK `dgbtrf`+`dgbtrs` ou le driver `dgbsv`. Pour une matrice tridiagonale en stockage bande, le coût en temps est $O(n)$ et la mémoire est $O(n)$.

5.2 LU tridiagonale dédiée (EX6)

Une factorisation LU spécifique à une matrice strictement tridiagonale peut être implémentée en ne mettant à jour que les coefficients nécessaires (pas de remplissage hors-bande). Cela réduit les constantes et évite le pivotage.

5.3 Résultats : temps et erreur

Les mesures ci-dessous proviennent de l'exécution : `./tpPoisson1D_direct {0,1,2} N`. Les trois modes correspondent aux trois variantes implémentées (LAPACK/Tridiag/Driver selon le code).

Table 1 – Méthodes directes : temps de résolution et erreur relative avant.

Taille N	Mode	Temps (s)	Erreur relative
10	0	3.94588e-04	2.692638e-16
10	1	3.82090e-04	2.692638e-16
10	2	4.19755e-04	2.692638e-16
100	0	4.14260e-05	6.052400e-15
100	1	4.02340e-05	6.052400e-15
100	2	3.97230e-05	6.052400e-15
1000	0	1.97088e-03	1.054400e-13
1000	1	1.01145e-03	1.054400e-13
1000	2	1.28592e-03	1.054400e-13

Discussion. Les erreurs sont proches de la précision machine, ce qui valide la partie "directe". On observe un coût global compatible avec une croissance proche de $O(n)$ pour une matrice tridiagonale. Pour de petites tailles, les mesures sont dominées par les coûts constants (initialisations, appels BLAS/LAPACK).

6 Méthodes itératives (EX7–EX9)

On note $r^{(k)} = b - Ax^{(k)}$ et on utilise un critère d'arrêt sur le résidu relatif :

$$\frac{\|r^{(k)}\|_2}{\|b\|_2} < \varepsilon, \qquad \varepsilon = 10^{-3}. \tag{9}$$

6.1 EX7 : Richardson

$$x^{(k+1)} = x^{(k)} + \alpha r^{(k)}. \tag{10}$$

6.2 EX8 : Jacobi

Préconditionneur diagonal $M = D$:

$$x^{(k+1)} = x^{(k)} + D^{-1} r^{(k)}. \tag{11}$$

6.3 EX9 : Gauss–Seidel

Préconditionneur triangulaire inférieur $M = D + L$:

$$x^{(k+1)} = x^{(k)} + (D + L)^{-1} r^{(k)}. \tag{12}$$

6.4 Résultats : nombre d'itérations

Pour $n = 100$, les itérations observées sont :

TABLE 2 – Méthodes itératives (n=100) : nombre d'itérations.

Méthode	Itérations
Richardson (EX7)	126
Jacobi (EX8)	126
Gauss–Seidel (EX9)	64

6.5 Courbes de convergence

Commentaire. Les courbes Richardson et Jacobi se superposent ici (mêmes itérations), tandis que Gauss–Seidel converge environ deux fois plus vite sur ce cas, ce qui est attendu pour une matrice SPD tridiagonale.

7 Formats creux CSR/CSC (EX10)

L'objectif de l'EX10 est d'étendre la bibliothèque pour gérer des stockages creux :

— construction de A au format CSR et CSC ;
— produits matrice–vecteur `dcsrmv` et `dcscmv` ;
— adaptation des algorithmes itératifs sur ces formats.

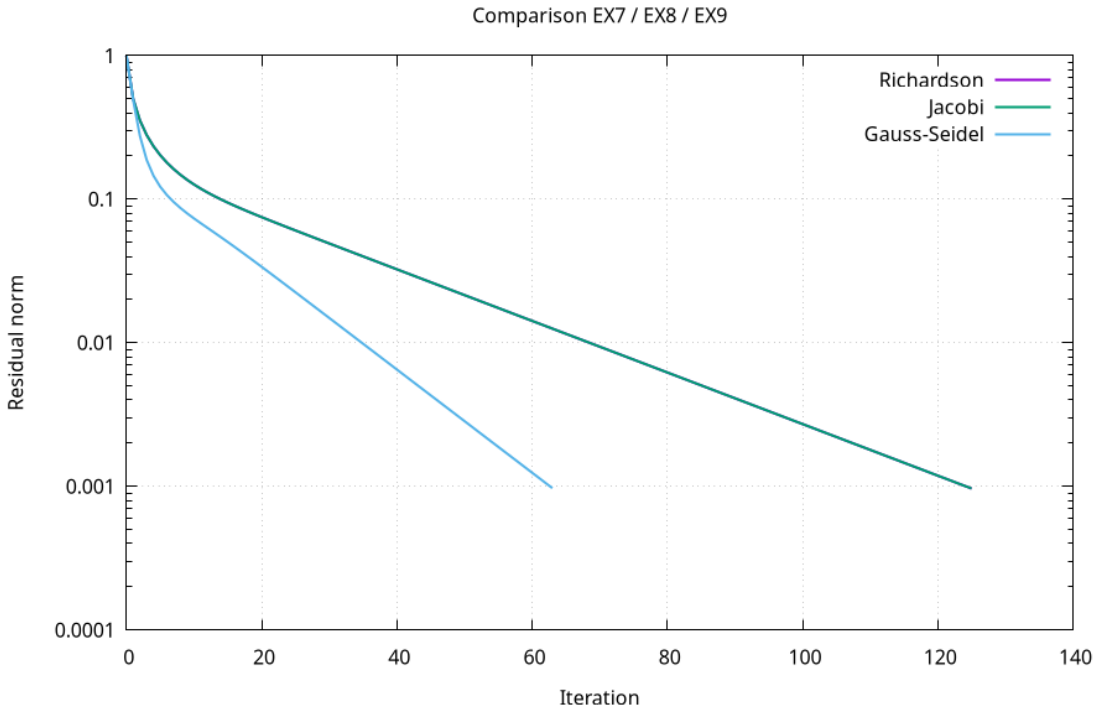

FIGURE 1 – Historique de convergence (échelle log sur l'axe y).

Pour Poisson 1D, le nombre d'éléments non nuls vaut :

$$\text{nnz} = 3n - 2. \tag{13}$$

7.1 Résultat observé

Sur l'exécution : `./tpPoisson1D_sparse` (CSR Richardson), avec $maxit = 10000$, on obtient :

— nombre d'itérations : **10000** (atteinte de $maxit$),
— dernier résidu : **7.169079e-03**.

Ce résultat indique une convergence lente : l'algorithme diminue le résidu mais n'atteint pas le seuil très strict si $maxit$ est insuffisant, ce qui est cohérent avec le conditionnement croissant du problème.

Remarque. Le format creux est surtout pertinent pour généraliser vers des problèmes 2D/3D où la matrice est très grande et où les formats denses/bandes deviennent coûteux en mémoire.

8 Conclusion

Ce TP met en évidence l'intérêt d'exploiter la structure des matrices :

— Les méthodes directes basées sur LAPACK donnent une solution très précise (erreurs proches machine) et un coût compatible avec $O(n)$ en tridiagonal.
— Les méthodes itératives dépendent fortement du préconditionnement : Gauss–Seidel réduit significativement le nombre d'itérations.
— L'extension CSR/CSC est une étape clé pour la généralisation à des problèmes de dimension supérieure.