UNIVERSITÉ VERSAILLES-SAINT-QUENTIN
PARIS-SACLAY

SPÉCIALITÉ :

Calcul Haute Performance, Simulation

MODULE :

Calcul Numérique

Rapport :

Résolution de l'équation de la chaleur 1D stationnaire
Exercices 5 et 6 – Méthodes directes

Présenté par :
Zamoum Tassadit

Encadré par :
J. Gurhem
T. Dufaud

27 Décembre 2025

Table des matières

1 Introduction

La résolution numérique de l'équation de la chaleur en régime stationnaire conduit, après discrétisation par la méthode des différences finies, à la résolution d'un système linéaire de grande taille. Dans le cas unidimensionnel étudié dans ce travail, la matrice associée au problème de Poisson 1D possède une structure tridiagonale, ce qui permet d'envisager des méthodes de résolution directe particulièrement efficaces.

Les exercices 5 et 6 ont pour objectif d'étudier et de comparer des méthodes de résolution directe adaptées à ce type de systèmes. Une première approche repose sur l'utilisation des bibliothèques BLAS et LAPACK, qui fournissent des routines générales et robustes pour la factorisation et la résolution de systèmes linéaires stockés sous forme bande. Cette méthode constitue une référence fiable, tant du point de vue de la précision numérique que de la stabilité des calculs.

Dans un second temps, une méthode de factorisation LU spécifique aux matrices tridiagonales est implémentée. En exploitant directement la structure du problème, cette approche vise à réduire le coût de calcul et à améliorer les performances par rapport à une méthode générale.

Pour chacune des méthodes, la validité numérique de la solution est vérifiée, puis une étude de performances est menée en fonction de la taille de la matrice. Les temps d'exécution sont mesurés pour différentes valeurs de n, et les résultats sont analysés à l'aide de courbes de performance et de représentations en échelle logarithmique afin d'interpréter la complexité algorithmique.

Les implémentations sont réalisées en langage C et exécutées dans un environnement Docker afin de garantir la reproductibilité des résultats.

2 Méthode directe avec LAPACK

2.1 Principe de la méthode

Le système linéaire issu de la discrétisation du problème de Poisson 1D s'écrit :

$$Au = f, \quad A \in \mathbb{R}^{n\times n}, \quad u, f \in \mathbb{R}^n. \tag{1}$$

La matrice A est tridiagonale et stockée sous forme bande générale (GB). La résolution directe est réalisée à l'aide des routines LAPACK suivantes :

— `DGBTRF` : factorisation LU de la matrice bande.

— `DGBTRS` : résolution du système linéaire à partir de la factorisation.

Ces deux appels correspondent aux étapes internes de la routine `DGBSV`, qui combine factorisation et résolution dans une interface unique. Dans ce travail, les étapes sont appelées explicitement afin de permettre une analyse plus fine des performances.

Les routines LAPACK reposent sur les bibliothèques BLAS pour l'exécution des opérations linéaires élémentaires (produits matrice–vecteur, opérations arithmétiques de base), permettant de bénéficier d'implémentations optimisées pour l'architecture matérielle.

2.2 Validation numérique

La validation de la solution est réalisée à l'aide du résidu relatif :

$$\frac{\|Au - f\|}{\|f\|} \tag{2}$$

Les calculs effectués donnent un résidu relatif égal à

$$\text{relres} = 2.69 \times 10^{-16}.$$

Ce résultat est de l'ordre de la précision machine en double précision, ce qui indique que la solution obtenue est numériquement exacte au sens de l'arithmétique flottante. La méthode directe basée sur LAPACK est donc correctement implémentée et stable sur ce cas test.

2.3 Analyse des performances

Les temps d'exécution ont été mesurés pour différentes tailles de matrices allant de $n = 100$ à $n = 10000$. La Figure 1 présente le temps d'exécution de la méthode LAPACK en fonction de la taille du problème.

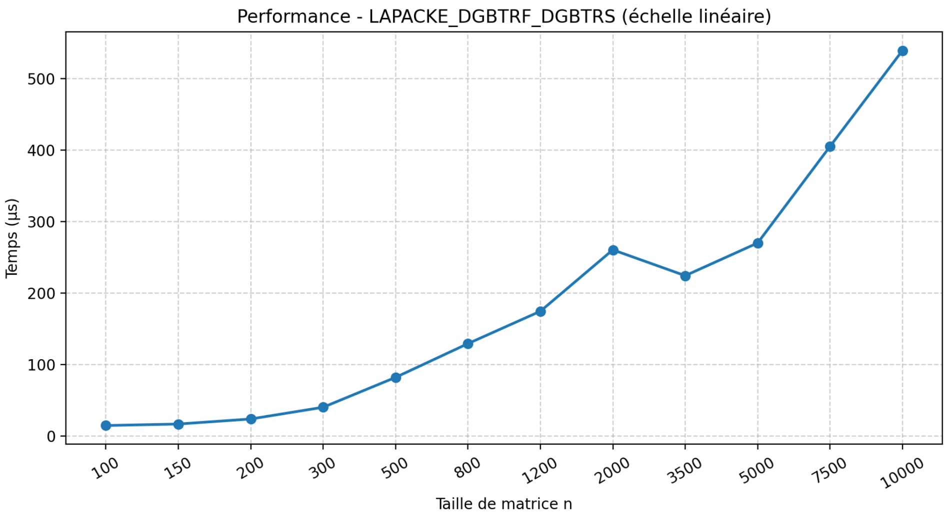

FIGURE 1 – Temps d'exécution de la méthode LAPACK (DGBTRF/DGBTRS)

2.3.1 Interprétation des résultats

On observe que le temps d'exécution de la méthode LAPACK augmente globalement avec la taille de la matrice. Pour les tailles testées, les temps mesurés passent d'environ 15 μs pour $n = 100$ à environ 540 μs pour $n = 10000$.

Les variations locales observées entre certaines tailles successives ne traduisent pas un changement de comportement algorithmique, mais sont dues aux effets du cache, à l'ordonnancement du processeur et au coût fixe des appels aux routines LAPACK, qui devient significatif lorsque les temps de calcul sont très faibles.

2.4 Analyse de complexité

Dans le cas étudié, la matrice issue de la discrétisation est tridiagonale et stockée sous forme bande, avec une largeur de bande constante (nombre de diagonales non nulles indépendant de n). Dans ce cadre, la factorisation LU et la résolution associée ont un coût proportionnel à n, donc une complexité attendue en $O(n)$. Les mesures expérimentales (temps en fonction de n) sont cohérentes avec cette tendance, malgré de légères fluctuations dues aux effets d'architecture (cache) et au coût fixe des appels aux routines.

3 Factorisation LU tridiagonale

3.1 Principe de la méthode

Dans cet exercice, une factorisation LU spécifique aux matrices tridiagonales est implémentée. Cette méthode exploite directement les trois diagonales non nulles de la matrice, ce qui permet de réduire le nombre d'opérations et les accès mémoire par rapport à une méthode générale.

3.2 Validation

La validation de la méthode LU tridiagonale est réalisée par comparaison avec la solution obtenue à l'aide de la méthode LAPACK. Les solutions numériques fournies par les deux approches coïncident à la précision machine, le résidu relatif étant égal à 2.69×10^{-16}, valeur identique à celle obtenue avec la méthode LAPACK.

Cette concordance confirme que l'implémentation de la factorisation LU tridiagonale est correcte et stable du point de vue numérique. Elle montre également que l'exploitation explicite de la structure tridiagonale n'entraîne aucune dégradation de la précision par rapport à une méthode générale.

3.3 Performances

La Figure 2 présente les performances de la factorisation LU tridiagonale.

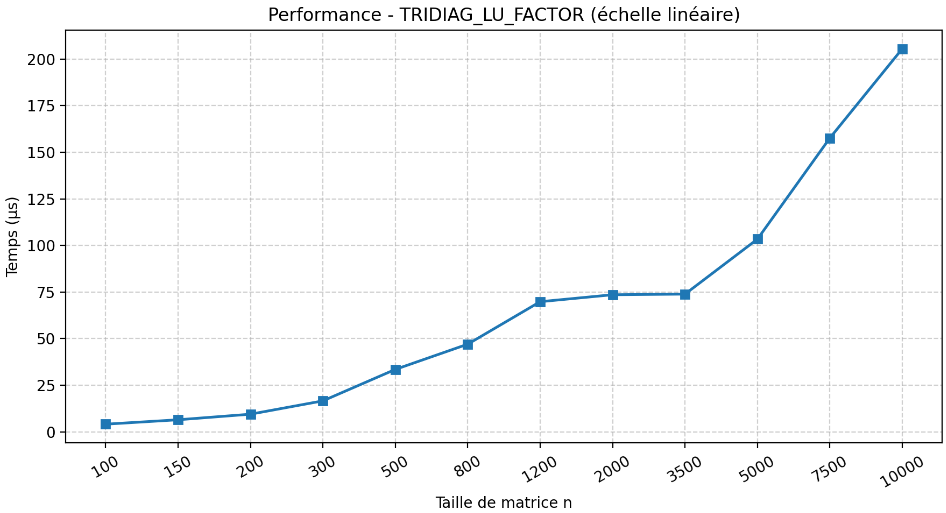

Figure 2 – Temps d'exécution de la factorisation LU tridiagonale

3.3.1 Interprétation des résultats

La courbe associée à la factorisation LU tridiagonale présente une tendance globalement linéaire du temps d'exécution en fonction de la taille du système. Toutefois, les temps mesurés sont systématiquement inférieurs à ceux obtenus avec la méthode LAPACK.

Ce comportement s'explique par le caractère spécialisé de l'algorithme, qui exploite directement la structure tridiagonale de la matrice et réduit ainsi le nombre d'opérations arithmétiques ainsi que les accès mémoire. L'absence de surcoûts liés à une implémentation générale permet d'obtenir de meilleures performances en pratique.

3.4 Analyse de complexité

La factorisation LU tridiagonale exploite directement la structure tridiagonale de la matrice, ce qui conduit à un nombre d'opérations proportionnel au nombre d'inconnues. On s'attend donc à une complexité théorique linéaire en $O(n)$, ce qui est cohérent avec la tendance observée sur les courbes de temps d'exécution.

3.5 Comparaison des méthodes

La Figure 3 compare les performances des deux méthodes étudiées.

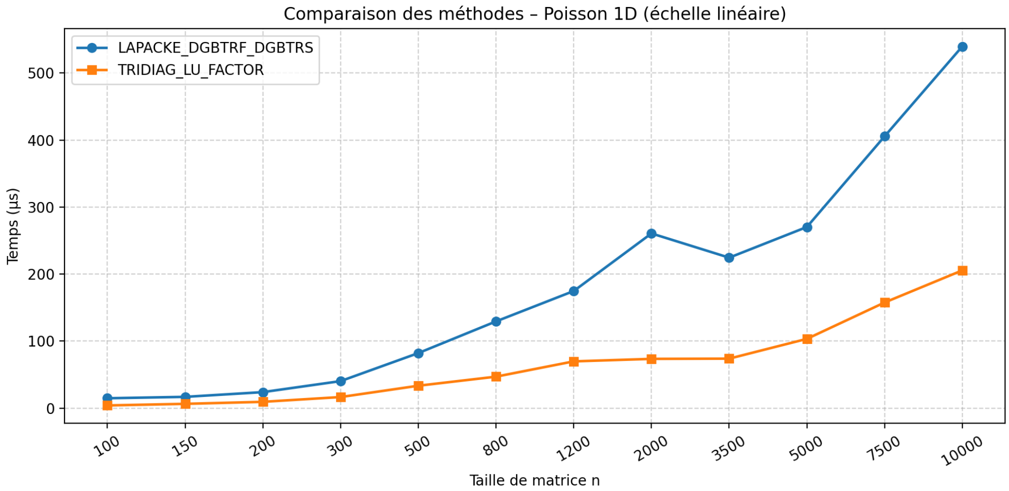

FIGURE 3 – Comparaison des méthodes LAPACK et LU tridiagonale

3.5.1 Interprétation des résultats

Pour l'ensemble des tailles testées, la factorisation LU tridiagonale est systématiquement plus rapide que la méthode LAPACK. Les mesures montrent un gain de performance compris entre un facteur 2.4 et 3.6 selon la taille du système.

Par exemple, pour $n = 10000$, le temps d'exécution est d'environ 540 μs pour la méthode LAPACK, contre 206 μs pour la factorisation LU tridiagonale. Ce gain reste du même ordre de grandeur lorsque n augmente, ce qui indique que la différence observée est principalement due à une réduction des constantes multiplicatives.

3.5.2 Comparaison quantitative

Afin de compléter la comparaison graphique, le tableau 1 présente quelques valeurs représentatives extraites du fichier `timing.csv`. Les temps sont exprimés en microsecondes. Le gain est défini comme le rapport $t_{\text{LAPACK}}/t_{\text{LU}}$.

n	LAPACK (μs)	LU tridiagonale (μs)	Gain
100	14.8	4.1	3.6
500	82.2	33.5	2.5
1200	174.8	69.8	2.5
5000	270.3	103.4	2.6
10000	539.7	205.6	2.6

TABLE 1 – Comparaison quantitative des temps d'exécution

3.6 Analyse de complexité

La Figure 4 présente l'analyse de complexité en échelle log-log.

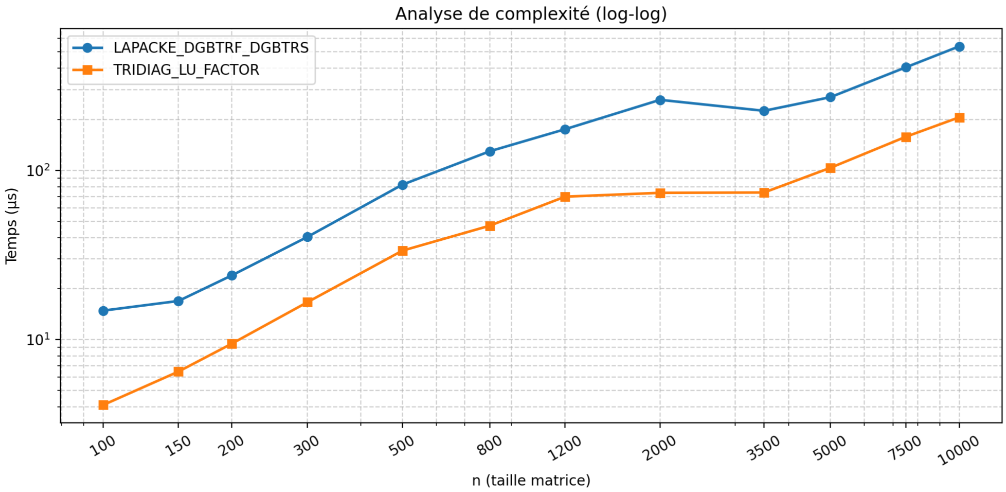

FIGURE 4 – Analyse de la complexité algorithmique (log-log)

3.6.1 Interprétation des résultats

La représentation en échelle log–log met en évidence une pente proche de 1 pour les deux méthodes, ce qui indique un comportement compatible avec une complexité linéaire $O(n)$. Cette observation confirme que les deux approches partagent le même comportement asymptotique.

L'écart vertical quasi constant entre les deux courbes traduit une différence de constantes multiplicatives, et non une différence de complexité. Cet écart est directement lié au caractère général de la méthode LAPACK, par opposition à la méthode tridiagonale, spécifiquement optimisée pour la structure du problème.

4 Conclusion

Les exercices 5 et 6 ont permis de comparer deux approches de résolution directe appliquées au problème de Poisson 1D stationnaire issu de la discrétisation par différences finies. L'objectif n'était pas uniquement d'obtenir une solution numérique, mais d'analyser l'impact du choix de l'algorithme sur la précision et les performances de calcul.

La méthode basée sur les routines LAPACK `DGBTRF` et `DGBTRS`, reposant sur les bibliothèques BLAS pour les opérations de base, constitue une solution de référence pour la résolution de systèmes linéaires stockés sous forme bande. Les résultats obtenus montrent une excellente précision numérique, le résidu relatif étant de l'ordre de la précision machine, garantissant robustesse et stabilité indépendamment de la structure fine de la matrice.

L'implémentation d'une factorisation LU spécifique aux matrices tridiagonales met en évidence l'intérêt d'exploiter explicitement la structure du problème. Bien que les deux méthodes présentent une complexité théorique linéaire en fonction de la taille du système, l'étude des performances montre que la méthode spécialisée est systématiquement plus efficace en pratique. Cette amélioration s'explique par la réduction du nombre d'opérations arithmétiques et des accès mémoire, ainsi que par l'absence de surcoûts liés à une implémentation généraliste.

Les courbes de performances et l'analyse en échelle logarithmique confirment que, dans un contexte de calcul scientifique et de calcul haute performance, les constantes multiplicatives et l'organisation mémoire jouent un rôle déterminant, en particulier lorsque les temps de calcul sont très faibles. Ce travail illustre ainsi l'importance de choisir des méthodes numériques adaptées à la structure du problème afin d'obtenir des gains significatifs tout en conservant une excellente précision numérique.