1 Introduction

L'objectif de l'exercice 4 et 5 du TP est de résoudre numériquement l'équation de la chaleur stationnaire en dimension 1. La discrétisation de cette équation par la méthode des différences finies conduit à un système linéaire de la forme $Au = f$. La matrice A obtenue possède une propriété importante : elle est tridiagonale. L'enjeu de ce travail est d'exploiter cette structure creuse pour optimiser le stockage et le temps de calcul, en utilisant d'abord la bibliothèque standard LAPACK, puis en proposant notre propre implémentation.

2 Exercice 5 : Résolution avec LAPACK (DGBTRF et DGBTRS)

Nous avons d'abord utilisé la bibliothèque LAPACK qui est la référence pour l'algèbre linéaire numérique. Afin d'économiser la mémoire, la matrice n'est pas stockée sous forme pleine mais sous le format *General Band* (GB). Pour notre problème, ce stockage ne nécessite que quelques vecteurs diagonaux, ce qui réduit considérablement l'occupation mémoire par rapport à un stockage classique.

2.1 Validation de la méthode

Pour vérifier le bon fonctionnement de l'appel aux fonctions LAPACK `dgbtrf` (factorisation LU) et `dgbtrs` (résolution), nous avons calculé l'erreur relative entre notre solution numérique et la solution exacte connue du problème.

Nous obtenons une erreur relative de l'ordre de 2.69×10^{-16}. Ce résultat correspond à la précision machine pour des nombres en double précision. Cela confirme que la résolution du système linéaire est numériquement exacte aux erreurs d'arrondi près.

2.2 Analyse des performances

Nous avons mesuré le temps d'exécution global de la méthode pour des tailles de matrices allant de $N = 100$ à $N = 1\,000\,000$.

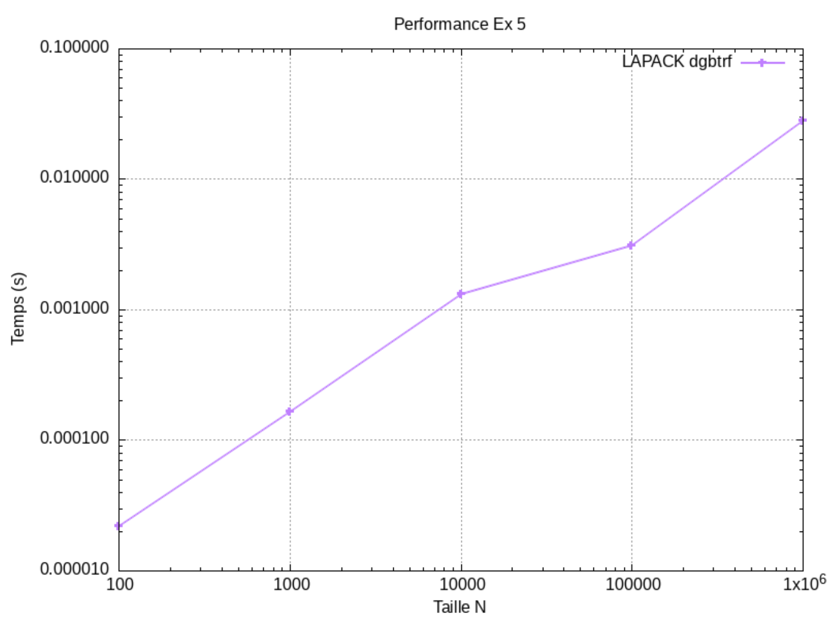

Figure 1: Temps de calcul de la méthode LAPACK

Interprétation de la complexité : Le graphique ci-dessus présente l'évolution du temps de calcul en fonction de la taille N avec des échelles logarithmiques. Nous observons une droite de pente 1. Cela signifie que lorsque la taille de la matrice est multipliée par 10, le temps de calcul est également multiplié par environ 10. Cette observation expérimentale prouve que la complexité de la méthode utilisant le stockage bande est linéaire, c'est-à-dire en $O(N)$. Cela est beaucoup plus efficace qu'une résolution standard qui serait en $O(N^3)$, car la largeur de la bande reste constante quelle que soit la taille de la matrice.

3 Exercice 6 : Factorisation LU Manuelle pour matrices tridiagonales

Dans cette partie, nous avons cherché à implémenter nous-mêmes la factorisation LU sans passer par la fonction générique de LAPACK. L'objectif était de concevoir un algorithme spécifiquement dédié aux matrices tridiagonales, c'est-à-dire avec exactement une sous-diagonale et une sur-diagonale.

3.1 Implémentation et Validation

Nous avons développé la fonction `dgbtrftridiag`. L'algorithme parcourt la matrice et élimine la sous-diagonale par une méthode de Gauss simplifiée. Cette approche évite les tests inutiles sur la largeur de bande puisque la structure de la matrice est connue et fixe.

La validation a été effectuée de la même manière que précédemment. L'erreur relative obtenue est identique à celle de LAPACK, soit environ 2.69×10^{-16}. Cela valide la justesse de notre algorithme manuel.

3.2 Analyse des performances

De la même façon que pour l'exercice précédent, nous avons tracé le temps d'exécution en fonction de la taille du problème.

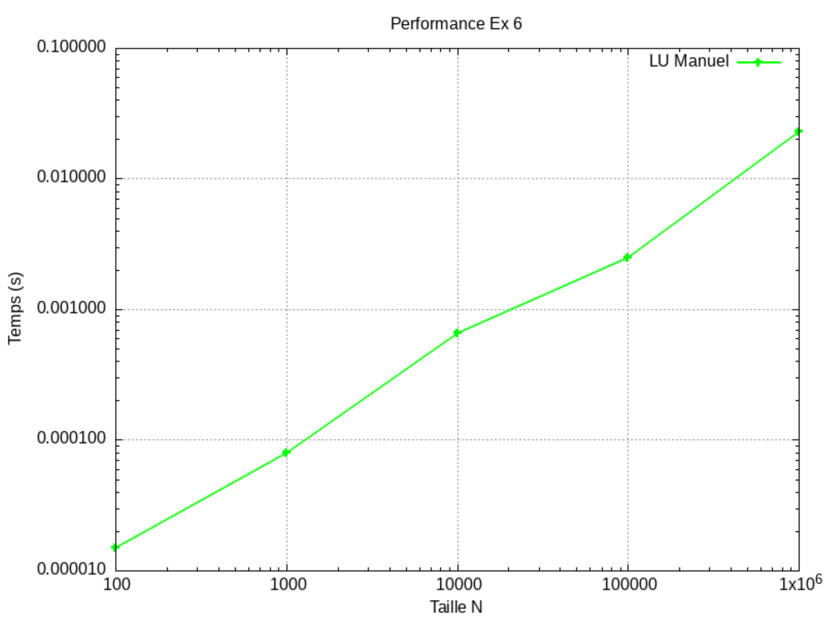

Figure 2: Temps de calcul de la méthode LU

Interprétation de la complexité : Nous retrouvons ici aussi une courbe linéaire. L'algorithme manuel conserve donc bien la complexité optimale en $O(N)$. En analysant les temps mesurés, nous remarquons que cette méthode spécifique est très rapide. Pour une matrice de taille $N = 1\,000\,000$, le système est résolu en quelques centièmes de seconde. Cela confirme que l'écriture d'un code dédié à une structure tridiagonale permet d'obtenir d'excellentes performances en évitant le surcoût lié à la gestion de cas plus généraux.

4 Conclusion

Ce travail sur les méthodes directes a mis en évidence l'importance du format de stockage. En passant d'une matrice pleine à un stockage bande, nous avons réduit la complexité algorithmique de cubique à linéaire. Les mesures réalisées confirment que pour des matrices tridiagonales, la résolution est extrêmement rapide et précise.