

MASTER 1 CHPS

Rapport de Calcul Numérique

Méthodes directes pour l'équation de la chaleur 1D

Exercices 5 et 6 : Résolution LAPACK et Factorisation LU

Étudiant :
Iram MADANI-FOUATIH

Enseignants :
M. T. DUFAUD
M. J. GURHEM

29 décembre 2025

Table des matières

1	Introduction	2
2	Exercice 5 : Résolution LAPACK (Stockage Bande)	2
2.1	Problématique du stockage	2
2.2	Implémentation du format GB	2
2.3	Appels aux routines LAPACK	2
3	Exercice 6 : Factorisation LU Spécialisée	3
3.1	Optimisation algorithmique	3
3.2	Algorithme Implémenté	3
4	Analyse des Résultats et Complexité	3
4.1	Complexité Théorique	3
4.2	Validation Numérique	4
5	Conclusion	4
	Références	5

1 Introduction

Ce TP s'inscrit dans le cadre de la résolution numérique d'équations différentielles partielles (EDP). Nous étudions l'équation de la chaleur stationnaire en 1D :

$$\begin{cases} -k \frac{\partial^2 T}{\partial x^2} = g(x), & x \in]0, 1[\\ T(0) = T_0, \quad T(1) = T_1 \end{cases} \quad (1)$$

La discrétisation de ce problème par différences finies centrées mène à un système linéaire $Au = f$ où la matrice A possède une structure très particulière : elle est **tridiagonale**, symétrique et définie positive.

L'objectif de ce rapport est de présenter l'implémentation et l'analyse des méthodes directes pour résoudre ce système. Nous nous concentrerons sur deux approches :

- **Exercice 5** : L'utilisation de la bibliothèque standard **LAPACK** avec un stockage optimisé (Bande).
- **Exercice 6** : L'implémentation manuelle d'une factorisation **LU spécialisée** pour matrices tridiagonales.

Les fichiers sources (main, librairies, headers) et l'architecture globale du projet ont été fournis. Notre travail a consisté à implémenter le cœur des algorithmes numériques et la gestion de la mémoire.

2 Exercice 5 : Résolution LAPACK (Stockage Bande)

2.1 Problématique du stockage

Pour un maillage fin (ex : $N = 1000$ points), la matrice A contient 10^6 éléments. Cependant, seuls les éléments de la diagonale principale et des deux diagonales adjacentes sont non nuls. Utiliser un stockage "dense" classique gaspillerait environ 99% de la mémoire. Pour optimiser la consommation mémoire et la performance (cache CPU), nous utilisons le format **General Band (GB)** préconisé par LAPACK.

2.2 Implémentation du format GB

Pour une matrice tridiagonale ayant $kl = 1$ sous-diagonale et $ku = 1$ sur-diagonale, LAPACK requiert un tableau 'AB' de dimension $(2kl + ku + 1) \times N$. Nous avons implémenté la fonction `set_GB_operator_colMajor_poisson1D` dans `lib_poisson1D.c`.

Le stockage se fait colonne par colonne, avec un décalage d'indices. La matrice "aplatie" en mémoire contient 4 lignes :

- **Ligne 0** : Réservee pour le remplissage (fill-in) lors de la factorisation.
- **Ligne 1** : Sur-diagonale (valeurs -1).
- **Ligne 2** : Diagonale principale (valeurs 2).
- **Ligne 3** : Sous-diagonale (valeurs -1).

2.3 Appels aux routines LAPACK

Le programme principal `tp_poisson1D_direct.c` permet de sélectionner la méthode de résolution. Nous avons activé les appels suivants :

- **dgbtrf** (Double Precision General Band Triangular Factorization) : Cette routine effectue la décomposition $A = LU$. Elle gère le pivotage partiel pour assurer la stabilité numérique.
- **dgbtrs** (Solve) : Utilise la factorisation précédente pour résoudre le système $AX = B$ par descente et remontée triangulaire.

3 Exercice 6 : Factorisation LU Spécialisée

3.1 Optimisation algorithmique

La routine LAPACK **dgbtrf** est très robuste mais généraliste (elle fonctionne pour n'importe quelle largeur de bande). Pour notre cas spécifique (matrice strictement tri-diagonale), nous pouvons écrire un algorithme plus simple et potentiellement plus rapide car : 1. Nous connaissons exactement la position des zéros. 2. La matrice est à diagonale dominante, ce qui rend le pivotage numérique souvent superflu.

3.2 Algorithme Implémenté

Nous avons codé la fonction **dgbtrftridiag** qui réalise l'élimination de Gauss sans allocation mémoire supplémentaire (in-situ).

L'algorithme parcourt la matrice une seule fois (j de 0 à $N - 2$) :

```

1 for (j = 0; j < N - 1; j++) {
2     // 1. R cup ration du pivot (Diagonale)
3     double Ujj = AB[2 + j*LDAB];
4
5     // 2. Calcul du facteur L (Sous-diagonale normalis e)
6     double Ljp1j = AB[3 + j*LDAB] / Ujj;
7     AB[3 + j*LDAB] = Ljp1j; // Stockage direct de L
8
9     // 3. Mise      jour du pivot suivant (Schur complement)
10    // A_{j+1,j+1} = A_{j+1,j+1} - L_{j+1,j} * A_{j,j+1}
11    double Uj_jp1 = AB[1 + (j+1)*LDAB];
12    AB[2 + (j+1)*LDAB] -= Ljp1j * Uj_jp1;
13 }
```

Listing 1 – Cœur de la factorisation LU manuelle

Cette fonction remplace l'appel à **dgbtrf**. La résolution finale se fait toujours avec **dgbtrs**, car nous avons respecté le format de stockage des facteurs L et U attendu par LAPACK.

4 Analyse des Résultats et Complexité

4.1 Complexité Théorique

Le tableau ci-dessous résume les gains obtenus par rapport à une approche naïve.

Méthode	Mémoire	Temps (Fact.)	Temps (Résol.)
Dense (Classique)	$O(N^2)$	$O(N^3)$	$O(N^2)$
Bande LAPACK (Ex 5)	$O(N)$	$O(N)$	$O(N)$
Tridiag Manuelle (Ex 6)	$O(N)$	$O(N)$	$O(N)$

TABLE 1 – Comparaison des complexités pour une matrice de taille N

Nos deux implémentations sont linéaires ($O(N)$), ce qui est optimal pour ce problème. L'implémentation manuelle (Ex 6) présente une constante de temps plus faible car elle évite les tests de pivotage et les appels de fonctions complexes de LAPACK.

4.2 Validation Numérique

Nous avons validé la correction de nos algorithmes en calculant l'erreur relative (norme L2) entre notre solution numérique X_{num} et la solution analytique exacte X_{exact} :

$$E = \frac{\|X_{num} - X_{exact}\|}{\|X_{exact}\|}$$

Les résultats montrent que :

1. L'erreur obtenue avec LAPACK et notre méthode manuelle est strictement identique (aux erreurs d'arrondi près, $\sim 10^{-16}$).
2. L'erreur décroît quadratiquement avec le pas du maillage ($O(h^2)$), ce qui confirme que notre solveur résout correctement le système discret issu du schéma centré d'ordre 2.

5 Conclusion

Ce travail a permis de mettre en œuvre des méthodes directes performantes pour l'équation de la chaleur 1D. L'utilisation du format **General Band** s'est avérée indispensable pour traiter efficacement la matrice tridiagonale. Si LAPACK offre une solution robuste et prête à l'emploi (Exercice 5), la réimplémentation manuelle de la factorisation LU (Exercice 6) a permis une compréhension fine de l'algorithme d'élimination de Gauss et de l'accès mémoire, tout en fournissant une solution légèrement plus optimisée pour ce cas particulier.

Références

1. **Support de cours M1 CHPS**, *Calcul Numérique et Algèbre Linéaire*, Université de Versailles Saint-Quentin-en-Yvelines.
2. **Anderson, E. et al.**, *LAPACK Users' Guide*, Third Edition, SIAM, 1999. <http://www.netlib.org/lapack/>
3. **Sujet de TP 5**, *Méthodes directes et itératives pour résoudre l'équation de la chaleur 1D*, T. Dufaud, J. Gurhem.