RAPPORT TD/TP 5 CALCUL NUMÉRIQUE

Dorian Abaul

Décembre 2024

1 Introduction

L'objectif de ce TD/TP est d'appliquer un ensemble d'algorithmes étudiés en cours et TD pour la résolution d'un système linéaire obtenu par discrétisation par la méthode des différences finies de l'équation de la chaleur 1D stationnaire. Les implémentations seront faites en C avec BLAS et LAPACK. Afin de valider vous pouvez vous reposer sur les implémentations Scilab réalisées en TD

2 Référence et utilisation de BLAS/LAPACK

1.1 Utilisation de BLAS et LAPACK

Pour utiliser BLAS et LAPACK, on peut...

1.2 Format de stockage des matrices dans LAPACK

LAPACK a été créé en Fortran, un langage qui utilise par défaut le stockage des matrices par colonnes. LAPACK a conservé ce principe. Cependant, d'autres langages comme le C utilisent par défaut un accès par lignes. La constante `LAPACK_COL_MAJOR` indique à LAPACK que les matrices sont stockées en colonnes dans la mémoire (de manière contiguë).

1.3 La dimension principale (*leading dimension*)

La dimension principale (*leading dimension*, `ld`) représente l'espacement entre le début de deux lignes ou colonnes consécutives dans une matrice. Elle est généralement utilisée pour la linéarisation d'une matrice. Elle permet de déterminer la nouvelle position dans la matrice 1D grâce à une relation proportionnelle entre la dimension principale et les coordonnées de l'élément dans la matrice initiale.

1.4 La fonction `dgbmv`

La fonction `dgbmv` permet d'effectuer le produit d'une matrice stockée en bande (*band matrix*) avec un vecteur. Elle est utilisée pour optimiser les calculs sur des matrices creuses.

1.5 La fonction `dgbtrf`

La fonction `dgbtrf` réalise la factorisation LU d'une matrice stockée en bande. Cette décomposition exprime la matrice comme un produit de deux matrices : une matrice triangulaire inférieure L, une matrice triangulaire supérieure U, et un tableau de pivots permettant de gérer le stockage en bande.

1.6 La fonction `dgbtrs`

La fonction `dgbtrs` permet de résoudre un système linéaire pour une matrice stockée en bande, qui a préalablement été factorisée à l'aide de la fonction `dgbtrf`. Elle est donc utilisée pour résoudre un système linéaire en utilisant la factorisation LU.

1.7 La fonction `dgbsv`

La fonction `dgbsv` permet de résoudre un système linéaire pour une matrice stockée en bande. Contrairement à `dgbtrs`, elle effectue directement la factorisation LU (via `dgbtrf`) et résout ensuite le système linéaire. Autrement dit, `dgbsv` = `dgbtrf` + `dgbtrs`.

1.8 Calcul de la norme du résidu relatif avec BLAS

BLAS propose des méthodes pour calculer la norme du résidu relatif grâce à la fonction `dnrm2`.

3 DGBTRF, DGBTRS, DGBSV

2.1 Écrire le stockage GB pour la matrice Poisson 1D

Afin de répondre à la question, nous avons défini la fonction `set_GB_operator_colMajor_poisson1D`. Celle-ci permet de stocker la matrice Poisson 1D au format GB.

2.2 Utilisation de la fonction BLAS `dgbmv`

En utilisant la fonction `cblas_dgbmv`, nous obtenons les résultats suivants, stockés dans les fichiers `Mvec.dat` ou `RHS.dat` :

$$[-5.000000, 0.000000, -0.000000, 0.000000, 0.000000, 0.000000, -0.000000, 5.000000]$$

2.3 Méthode de validation

Pour obtenir une méthode de validation précise, nous utilisons un produit matrice-vecteur dont le résultat est connu afin de le comparer à notre propre solution. Dans le cadre de ce projet, la solution exacte est donnée par `EX_SOL`. Nous calculons ensuite $b = A \cdot x_{exact}$ pour résoudre le système linéaire.

Poisson 1D
Solution avec LAPACK : Temps d'exécution : `0.000016`
Validation par la solution exacte :

$$\frac{\|x - x_{exact}\|}{\|x_{exact}\|} = 4.364358 \times 10^{-1}$$

Erreur relative directe :

$$relres = 6.813851 \times 10^{-1}$$

L'erreur de validation (4.36×10^{-1}) est inférieure à l'erreur relative initiale (6.81×10^{-1}), mais elle demeure relativement élevée.

3.1 Évaluation de la performance et de la compléxitée

- Pour la factorisation LU , la complexité est de l'ordre o(n . kl . ku) avec kl (bande inférieure) et ku (bande supérieure).
- Pour la résolution de dgbtrs , la complexité est similaire à celle de LU.

Pour évaluer les performances nous allons utilisés la bibliothèque time pour chronométrer le temps d'exécution afin de comparer l'appel de LAPACK et l'implémentation.Les résultats sont :

- **TRF** (`LAPACK dgbtrf`) : 18×10^{-6} s
- **TRI** (résolution triangulaire) : 1×10^{-6} s
- **SV** (`LAPACK dgbsv`) : 6×10^{-6} s

Le resultat est l'implémentation est optimisé pour les matrices tridiagonales.