

Contexte et Introduction

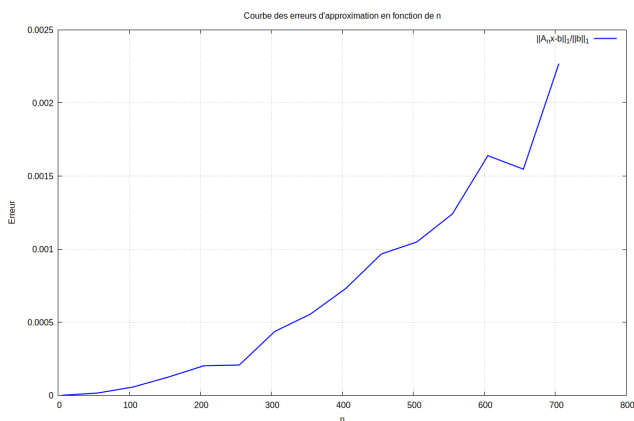
Nous souhaitons résoudre un système linéaire $Ax = b$ avec A matrice de taille $n \times n$ tridiagonale symétrique définie positive avec 2 comme valeur des éléments diagonaux et -1 pour les éléments sur-diagonaux et sous-diagonaux pour $n = 5 : 50 : 705$. Les éléments du vecteur b de taille n sont tous égaux à $h^2 = (1/(n+1))^2$.

La résolution de ce système se fera en langage C à l'aide de la fonction Fortran SGESV de la bibliothèque LAPACK.

Nous cherchons à observer les phénomènes liés à la résolution numérique de systèmes linéaires de structure identique mais de taille variable et de déterminer quelles sont les causes et leurs potentielles solutions.

Nous porterons une attention particulière aux différentes erreurs d'approximations obtenues que l'on estimera par $\|A\tilde{x} - b\|_1 / \|b\|_1$ avec \tilde{x} solution numérique du système. Nous faisons le choix volontaire de travailler en simple précision pour obtenir des résultats plus flagrants.

Observations graphiques des résultats numériques



- À mesure que la taille n du système augmente, l'erreur d'approximation du résultat augmente
- Pour $n = 5$ le résidu est proche - du même ordre - de l'erreur machine lorsque l'on travaille en simple précision
- Pour $n = 305$ l'erreur est d'ordre 10^{-4}
- Pour $n = 705$ l'erreur est d'ordre 10^{-3}

En mettant de côté le choix de travailler en simple précision, cette augmentation de l'erreur est notamment due à deux facteurs :

1. Le conditionnement de la matrice. En effet, à mesure que la taille n du système augmente, son conditionnement augmente pour passer d'une matrice bien conditionnée à une matrice mal conditionnée.
($n = 5 \rightarrow \text{cond}_2(A) = 1,39.10^1$; $n = 305 \rightarrow \text{cond}_2(A) = 3,79.10^4$; $n = 705 \rightarrow \text{cond}_2(A) = 2,02.10^5$)
2. L'augmentation de la taille du système implique une augmentation des opérations. De ce fait, les erreurs d'arrondis sont de plus en plus nombreuses et se cumulent.

Pistes d'amélioration de la mise en œuvre choisie

Notre matrice **A** est **tridiagonale symétrique définie positive**. Par conséquent, elle est composée d'un nombre élevé de zéros.

La fonction Fortran SGESV calcule la solution du système réel linéaire d'équations $Ax = b$ pour une **matrice A quelconque en simple précision**. Ainsi, sachant que la fonction traite du **cas général** elle ne prend pas en compte le nombre récurrent de zéros dans **A tridiagonale** et opère pour chaque valeur de la matrice, ceci est une perte considérable d'efficacité au vu du grand nombre de zéros.

Perspective d'amélioration : utilisation d'autres fonctions Fortran de la bibliothèque LAPACK :

- SGTSV : Matrice tridiagonale générale en simple précision
- DGTSV : Matrice tridiagonale générale en double précision
- **SPTSV : Matrice tridiagonale symétrique définie positive en simple précision** ^[1]
- DPTSV : Matrice tridiagonale symétrique définie positive en double précision

Changement de précision : passer de simple à double précision

- Changer la déclaration des variables de *float* à *double*.
 - Utiliser une autre fonction Fortran de la bibliothèque LAPACK que SGESV. Par exemple **DGESV** qui traite une matrice **A générale en double précision** ou bien **dans notre cas**, pour maximiser l'efficacité, la fonction **DPTSV**.
-

Réalisation de l'executable et execution du programme dans le terminal

Programme monolithique

```
## ===== Manuellement ===== ##
# Compilation
gcc -c mono.c
# Edition de lien et création de
l'executable
gcc mono.o -llapack -o mono.exe
# Execution
./mono.exe

## ===== ou ===== ##
# Compilation, édition de lien et
executable
gcc mono.c -o mono.exe
# Execution
./mono.exe

## ===== ##
# En utilisant le script.sh
./projet.sh
```

Deuxième version (avec appels de fonctions)

```
## ===== Manuellement ===== ##
# Compilation
gcc -c fonc.c operations.c
# Edition de lien et création de
l'executable
gcc fonc.o operations.o -llapack -o
fonc.exe
# Execution
./fonc.exe

## ===== ou ===== ##
# Compilation, édition de lien et
executable
gcc fonc.c operations.c -o fonc.exe
# Execution
./fonc.exe

## ===== ##
# En utilisant le script.sh
./projet.sh
```

22103988 KADDOURI Amyne

1. C'est la fonction qui, dans notre cas, serait la plus pertinente à utiliser ↗