École Internationale des Sciences du Traitement de l'Information

TP noté Analyse Numérique

BARRÉ Élodie
FERRIÉ Vincent

Cergy, le 12 avril 2012

Chapitre 1

Construction des matrices

1.1 expériences effectuées

Dans cette partie, on construit la matrice diagonale delta, qui nous permet de construire A. Afin de construire la matrice tridiagonale H, on doit utiliser les matrices orthogonales U et V. La fonction svd de Scilab permet de faire la décomposition en valeurs singulières d'une matrice.

Dans un premier temps, on commence par construire une matrice diagonale pour les différentes valeurs de K et de N. On prend les valeurs 10, 1000, $10^8, 10^{16} pour K et 10, 50, 100, 150 pour n$.

La matrice H est faite de telle sorte que l'on multiplie la norme 2 de A par la norme 2 de l'inverse de A. Pour calculer A, on tape [U,S,V]=svd(t) dans Scilab. Afin d'obtenir la décomposition en valeurs singulière de la matrice.

Dans notre code, nous n'avons pas testé toutes les valeurs de K et de n, mais il suffit de copier coller le petite partie de code correspondante et de remplacer les valeurs de K et n. Si nous ne l'avons pas mis, c'est parce que lorsque l'on compile cela prend beaucoup de place. Mais nous l'avons fait au moment où nous avons codé.

1.2 Analyse des résultats :

Nous n'avons pas réussi à inclure nos matrices dans ce rapport, mais en compilant notre code on peut aisèment remarquer que la méthode de Jacobi converge. Aprés avoir fait varier les valeurs de K et de N, et avoir décomposé A en valeurs singulières nous pouvons remarquer que les valeurs de la matrice A correspondent aux valeurs préalablement demandées.

Chapitre 2

Calculs à effectuer avec Scilab

2.1 expériences effectuées

Dans cette partie, on cherche à faire la résolution du système linéaire de nos matrices précédentes. Pour ce faire, on utilise la fonction Scilab linsolve qui nous permet de calculer la solution numérique x du système linéaire Ax = b. Et on évalue l'erreur de la solution obtenue. On utilise deux méthodes, celles de Jacobi et de "linsolve".

Pour commencer, on construit la fonction Jacobi afin d'utiliser la méthode itérative de Jacobi de dimension n.

2.2 Analyse des résultats :

On calculs l'erreur gràce à la méthode de "linsolve" avant d'utiliser la méthode de Jacobi. Enfin on calcul l'erreur relative en faisant la division de l'erreur par la méthode de Jacobi sur l'erreur par la méthode de "linsolve".

Après le calculs d'erreur, on trouve E=11.062873. On est relativement proche de la valeur attendue qui est de 10.

on a pris une valeur de $10^8 pour le seuil.$

L'erreur résiduelle de la solution obtenue est de 11.062873 tandis qu'avec la méthode directe, on était plus proche de 10. la méthode directe est certes plus proche, mais l'erreur rédiduelle peut être minimisée en augmentant le nombre d'itérations.

Chapitre 3

Erreur et nombre d'ittération

3.1 expériences effectuées

Dans cette partie, on veut m itérations, où m dépend du rayon spectral de la matrice Rj. Afin de calculer le rayon spectral d'une matrice, il existe sur Scilab une fonction spec pour le calculer. Les valeurs sont triées dans l'ordre décroissant.

Table des matières