Préparation à l'Agrégation de Mathématiques
Année 2006-2007

TP Scilab : méthodes directes de résolution des systèmes linéaires

Remarque : Dans toute cette fiche et en règle générale, on n'utilisera jamais le calcul de l'inverse de la matrice A, c'est-à-dire la fonction `inv` de Scilab.

1. Cas des matrices triangulaires

Programmer une fonction qui prend en entrée une matrice A triangulaire inférieure (resp. triangulaire supérieure) et un vecteur b, qui résout le système $Ax = b$ par une méthode de descente (resp. de remontée) et qui donne le vecteur solution x en sortie.

2. Méthode de l'élimination de Gauss

Programmer la méthode de Gauss sans permutation, puis la méthode de Gauss avec pivot partiel. L'appliquer à la résolution du système $Ax = b$, où $A \in \mathcal{M}_{n,n}(\mathbb{R})$ est définie par

$$\begin{cases} A_{i,i} = 2, & 1 \leq i \leq n, \\ A_{i,i+1} = A_{i+1,i} = -1, & 1 \leq i \leq n-1, \\ A_{i,j} = 0, & \text{sinon.} \end{cases}$$

et $b =^t (1, 0, \dots, 0, 1) \in \mathbb{R}^n$.

3. Factorisation LU d'une matrice

Adapter le programme précédent pour obtenir la factorisation LU de la matrice A.

4. Factorisation de Choleski d'une matrice

4.1. Ecrire l'algorithme qui permet de calculer la décomposition de Choleski. Pour cela, poser une matrice B triangulaire inférieure de coefficients $B_{i,j}$, $1 \leq i,j \leq n$ et déduire de l'égalité $A = BB^T$ les relations qui permettent de calculer les coefficients de B colonne par colonne.

4.2. Le programmer et l'appliquer à la même matrice A que précédemment.

5. Cas des matrices tridiagonales

D'après l'exercice 6 du TD, si A est une matrice tridiagonale qui vérifie l'hypothèse du théorème de la décomposition LU, alors les matrices L et U sont des matrices triangulaires bidiagonales. Si, de plus, A est définie positive, la matrice B de la décomposition de Choleski est elle aussi triangulaire bidiagonale. On peut exploiter ces informations pour réécrire les programmes dans le cas particulier des matrices tridiagonales.

5.1. Vérifier le théorème suivant : Soit $A = \begin{pmatrix} b_1 & c_1 & & & \\ a_2 & b_2 & c_2 & & (0) \\ & \ddots & \ddots & \ddots & \\ (0) & & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & a_n & b_n \end{pmatrix}$. On définit la suite $\delta_0 = 1$, $\delta_1 = b_1$ et $\delta_k = b_k\delta_{k-1} - a_k c_{k-1}\delta_{k-2}$. Alors la décomposition LU de la matrice A

est

$$LU = \begin{pmatrix} 1 & & & (0) \\ a_2\dfrac{\delta_0}{\delta_1} & 1 & & \\ & \ddots & \ddots & \\ (0) & & a_n\dfrac{\delta_{n-2}}{\delta_{n-1}} & 1 \end{pmatrix} \begin{pmatrix} \dfrac{\delta_1}{\delta_0} & c_1 & & (0) \\ & \dfrac{\delta_2}{\delta_1} & c_2 & \\ & & \ddots & \ddots \\ (0) & & & \dfrac{\delta_n}{\delta_{n-1}} \end{pmatrix}.$$

En déduire un algorithme pour calculer la décomposition LU de matrices tridiagonales. Combien d'opérations sont alors nécessaires pour la décomposition LU d'une matrice tridiagonale ? Comparer avec le nombre d'opérations nécessaires pour une matrice quelconque.

5.2. Adapter la décomposition de Choleski au cas d'une matrice tridiagonale et comparer le nombre d'opérations nécessaires avec le nombre d'opérations nécessaires en général.

6. Un exemple : entraînement aux textes..

Classement des pages web par Google
La Recherche, Octobre ou Novembre 2004

Pour classer les réponses à une requête donnée, Google utilise deux facteurs : le "score de pertinence" qui mesure l'adéquation entre la requête et le contenu de la page web et l'"indice de popularité" qui ne dépend pas de la consultation de la page par les internautes mais du nombre de liens qui pointent sur cette page à partir d'autres pages web. On s'intéresse ici au calcul de l'"indice de popularité" des pages web.

On considère qu'il y a N pages web au total ; en pratique, N est de l'ordre de 3 à 4 milliards (en 2004). On calcule la probabilité $p(A)$ de consulter la page A, en sachant qu'on arrive directement à la page A dans 20% des cas et indirectement (c'est-à-dire en utilisant un lien trouvé sur une autre page web) dans 80% des cas.

On appelle "indice de popularité"de A le nombre $IP(A) = N \times p(A)$. Il vérifie donc l'équation suivante :

$$IP(A) = 0,2 + 0,8\left(\frac{IP(B_1)}{N(B_1)} + \cdots + \frac{IP(B_{k(A)})}{N(B_{k(A)})}\right),$$

où les B_i, $1 \leq i \leq k(A)$ sont les pages qui ont un lien sortant vers A et $N(B_i)$ est le nombre de liens sortants de la page B_i.

6.1. On suppose que toutes les pages web ont au moins un lien sortant. Vérifier que la somme des indices de popularité de toutes les pages web est égale à N.

6.2. On considère le cas où la page A pointe vers la page B, la page B pointe vers A et C, la page C pointe vers B et D et la page D pointe vers C. Écrire le système linéaire 4×4 lié au cas ci-dessus. L'étendre au cas de n pages web.

6.3. Résoudre le système obtenu. Qu'en conclure ?

7. Références

Ciarlet, *Introduction à l'analyse numérique matricielle et à l'optimisation*, Dunod.
Lascaux-Théodor, *Analyse numérique matricielle appliquée à l'art de l'ingénieur*, Dunod.