M2 Agrégation - UE 8.

TP : Analyse Numérique matricielle.

Liste non exhaustive des points **à connaître :**
- *Matrices et propriétés classiques.*
- *Réduction des matrices*
- *Décomposition en valeurs singulières*
- *Valeurs propres des matrices, rayon spectral et propriétés, localisation des valeurs propres.*
- *Matrices à diagonale dominante, strictement dominantes etc...*
- *Normes matricielles.*
- *Conditionnement d'une matrice, d'un système linéaire*
- *Quelques problèmes de modélisation amenant à résoudre des sytèmes linéaires.*
- *Propriétés de convergence des suites de vecteurs ou matrices suivant le rayon spectral.*

$\sim\sim\sim$

Au niveau de la résolution proprement dite :
- *Méthodes directes de résolution : Elimination de Gauss, Factorisation LU, Cholesky, QR.*
- *Cas d'une matrice réelle symétrique définie positive, lien avec les méthodes de gradient.*
- *Notion de nombre d'opérations*
- *Méthodes itératives et leur convergence : Jacobi, Gauss Seidel, relaxation.*
- *Savoir traiter notamment l'exemple de la matrice des différences finies.*
- *Moindres carrés (cf. partie optimisation)*
- *Routines Scilab et algorithmes utilisés par ces routines*

On se donne $n \in \mathbb{N}$ et $\mathbb{K} = \mathbb{R}$ ou $\mathbb{C}$. On s'intéresse à la résolution de systèmes linéaires du type $Ax = b$ où $A \in \mathcal{M}_n(\mathbb{K})$, $x \in \mathbb{K}^n$ et $b \in \mathbb{K}^n$.

I. Routines Scilab

Ex 1. Identification

Identifier les routines Scilab permettant de résoudre des systèmes linéaires, factoriser des matrices, calculer le conditionnement... Sur quels algorithmes se basent-elles ? Les tester.

II. **Méthodes directes pour la résolution de systèmes linéaires**

II.1. Méthodes de descente et de remontée

Ex 2. Méthodes de descente et remontée : programmation...

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

II.2. Méthode d'élimination de Gauss et factorisation LU d'une matrice

Ex 3. Programmation

1) Programmer la méthode d'élimination de Gauss sans permutation, puis la méthode de Gauss avec pivot partiel.

2) L'appliquer à la résolution du système $Ax = b$, où $A \in \mathcal{M}_n(\mathbb{R})$ est définie par

$$\begin{cases} a_{ii} = 2, & 1 \leq i \leq n, \\ a_{i,i+1} = a_{i+1,i} = -1, & 1 \leq i \leq n-1, \\ a_{ij} = 0, \text{ sinon}, & \end{cases}$$

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

3) Adapter le programme précédent pour obtenir la factorisation LU d'une matrice A donnée.

II.3. Factorisation de Cholesky d'une matrice

1) Écrire l'algorithme qui permet de calculer la décomposition de Cholesky. Pour cela, si B est une matrice triangulaire inférieure de coefficients B_{ij}, $1 \leq i, j \leq n$, écrire l'égalité $A = BB^T$ et en déduire les relations qui permettent de calculer les coefficients de B colonnes par colonnes.

2) Le programmer et l'appliquer à la même matrice A que celle de l'exercice 2.

II.4. Cas des matrices tridiagonales

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 Cholesky est elle aussi triangulaire bidiagonale. On peut exploiter ces informations pour adapter les algorithmes à ces cas particuliers.

Theorem II.1. *[LT1] 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

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

Ex 4. Cas des matrice tridiagonales

1) Démontrer le théorème.

2) 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.

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

III. Méthodes itératives pour la résolution de systèmes linéaires

III.1. Programmation des méthodes classiques

Ex 5. Programmation

1) Programmer la méthode de Jacobi, la méthode de Gauss-Seidel et la méthode de relaxation de paramètre ω, pour calculer la solution du système $Ax = b$ à partir d'un vecteur initial x_0.

2) Programmer la méthode de gradient à pas constant et la méthode de gradient à pas optimal pour calculer la solution du système $Ax = b$ à partir d'un vecteur initial x_0.

3) Pour chacune des méthodes précédentes, tracer l'erreur $\|x^k - x\|$ en fonction de k où x^k est le k-ième vecteur itéré et x la solution exacte (connue). Grâce à un graphique log-log, mettre en évidence la vitesse de convergence de toutes ces méthodes. En particulier, étudier l'influence du pas choisi pour la méthode de gradient à pas constant et l'influence du paramètre ω pour la méthode de relaxation.

III.2. L'exemple de la matrice du Laplacien en différences finies

Soit A la matrice $\begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & (0) & \\ & \ddots & \ddots & \ddots & \\ (0) & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}$ de taille $n \in \mathbb{N}^*$. On considère également le vecteur $b = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}$.

La solution exacte du système $Ax = b$ est alors $x = (1, 1, \cdots, 1, 1)^T$.

1) Évaluer le nombre d'itérations nécessaires pour avoir une précision de 10^{-12} avec la méthode du gradient à pas constant, la méthode de gradient à pas optimal, la méthode de Jacobi, la méthode de Gauss-Seidel et la méthode de relaxation avec $\omega = 1.5$. Comparer ces trois méthodes pour différentes valeurs de n, par exemple $3 \leq n \leq 20$.

2) Vérifier numériquement la relation $\rho(\mathcal{L}_1) = \rho(J)^2$, valable pour les matrices tridiagonales.

3) Dans le cas où $n = 20$, trouver graphiquement le paramètre ω optimal pour la méthode de relaxation, en représentant le rayon spectral de la matrice obtenue en fonction de ω. Montrer que ce paramètre est égal à $\omega_{opt} = \dfrac{2}{1 + \sqrt{1 - \rho(J)^2}}$. Compléter la question 1) avec la méthode de relaxation avec un paramètre optimal.

Références :

[Ci] P. Ciarlet, *Introduction à l'analyse numérique matricielle et à l'optimisation,* Dunod.

[Fi] F. Filbet, *Analyse numérique,* Dunod.

[LT1] P. Lascaux, R. Théodor, *Analyse numérique matricielle appliquée à l'art de l'ingénieur, tome 1,* Dunod.

[LT2] P. Lascaux, R. Théodor, *Analyse numérique matriciellea ppliquée à l'art de l'ingénieur, tome 2,* Dunod.

[S] M. Schatzman, *Analyse Numérique,* Dunod.