Université Abou Bekr Belkaid 2016/2017
Département de Mathématiques
Module: Analyse numérique 2
Fiche de TP N°1

Résolution des systèmes linéaires
(méthodes directes)

I- Méthode d'élimination de Gauss

1- Ecrire une fonction Matlab qui résoud le système linéaire $Ax = b$ par la méthode d'élimination de Gauss sans choix du pivot sous la forme *function x=Gausselim(A,b)*

2- Déterminer la solution du problème:

$$\begin{pmatrix} -0.04 & 0.04 & 0.12 \\ 0.56 & -1.56 & 0.32 \\ -0.24 & 1.24 & -0.28 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 3 \\ 1 \\ 0 \end{pmatrix}$$

Vérifier les résultats obtenus en calculant le résidu Ax-b

3- Modifier le programme précédent pour avoir un choix de **pivot partiel**

4- Tester votre programme sur l'exemple suivant:

$$\begin{pmatrix} 7 & 35 & 1 \\ 3 & 15 & 3 \\ 3 & 20 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 10.6 \\ 4.8 \\ 5.5 \end{pmatrix}$$

II- Décomposition LU

1- Ecrire une fonction decomplu (sans stratégie de pivot) contenant une seule boucle for, qui a une matrice donnée A retourne les matrices L et U de la factorisation de A si la matrice A est carrée et un message d'erreur sinon.

Tester avec :

$$A = \begin{pmatrix} 2 & 3 & 0 & 1 \\ 3 & 5 & -2 & 7 \\ 0 & -2 & 1 & 2 \\ -1 & 7 & 2 & 0 \end{pmatrix}.$$

2-Utiliser cette fonction pour trouver le déterminant de A.

3-Ecrire une procédure descente et une procédure remontée pour résoudre respectivement $Ly = b$ et $Ux = y$

4-Appliquer ces procédures pour résoudre $LUx_i = e_i$ où e_i est le i-ème vecteur de la base canonique de $\mathbb{R}^n$.Que vaut $(x_1 \mid x_2 \mid \cdots \mid x_n)$?

5-Modifier votre fonction afin qu'elle retourne un message d'erreur en cas de pivot nul.

Tester avec:

$$A = \begin{pmatrix} 1 & -1 & 1 & 1 \\ -1 & 1 & 2 & 2 \\ 1 & 1 & 3 & 1 \\ 1 & 1 & 1 & 4 \end{pmatrix}$$

III- Décomposition de Cholesky

On rappelle l'algorithme de factorisation de Cholesky d'une matrice symétrique définie positive

pour j=1 à n

$$l_{jj} = (a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2)^{\frac{1}{2}}$$

pour i=j+1 à n

$$l_{ij} = \frac{1}{l_{jj}}(a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk})$$

1- Ecrire une fonction Matlab L=cholesky(A), qui calcule la matrice triangulaire inférieure L dans la factorisation LL^T d'une matrice symétrique définie positive A.

Vérifier que votre algorithme marche en résolvant $Ax = b$ où A est la matrice $(100, 100)$ symétrique définie positive de Lehmer telle que $A_{ij} = \begin{cases} i/j, & j \geq i \\ j/i, & j < i \end{cases}$ (utiliser la commande matlab A = gallery('lehmer',100)) et $b \in \mathbb{R}^{100}$ est choisi pour que la solution exacte soit égale à $ones(100, 1)$. Vous pouvez utiliser \ de matlab pour les substitutions avant et arrière).

2- Pour N=100:100:1000, former la matrice A = gallery('lehmer',N) et résoudre $Ax = b$ où $b = \text{rand}(N, 1)$ en utilisant votre algorithme de Cholesky et avec une factorisation LU sans pivotage. Est-ce que les temps de calcul pour les deux algorithmes se comportent comme prévu par la théorie? Proposer des raisons pourquoi sinon.