Exercice 1: Préparation pour la factorisation LU

Soit A une matrice 3×3 dont les lignes sont notées l_1, l_2 et l_3. Soient a, b et c trois réels.

a) Trouver la matrice E_1 telle que $E_1 A$ soit la matrice constituée des lignes $(l_1, al_1+l_2, bl_1+l_3)$.

b) Trouver la matrice E_2 telle que $E_2 A$ soit la matrice constituée des lignes $(l_1, l_2, cl_2 + l_3)$.

c) Calculer E_1^{-1}, E_2^{-1} puis $(E_2 E_1)^{-1}$

Exercice 2: Factorisation LU

Soit $A = \begin{pmatrix} 1 & 1 & 1 \\ 2 & 3 & 4 \\ 2 & 5 & 7 \end{pmatrix}$ et $b_1 = \begin{pmatrix} 1 \\ 3 \\ 5 \end{pmatrix}$ et $b_2 = \begin{pmatrix} 3 \\ 9 \\ 14 \end{pmatrix}$.

a) Triangulariser le système $AX = b$.

b) Ecrire la factorisation LU de la matrice A.

c) Utiliser cette factorisation pour trouver X_1 et X_2 tels que $AX_1 = b_1$ et $AX_2 = b_2$.

d) Quel est l'intérêt de l'algorithme de factorisation LU par rapport à l'algorithme de l'élimination de Gauss?

Exercice 3: Conditionnement d'un système linéaire

On considère les matrices et vecteurs:

$$A = \begin{pmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{pmatrix}, b = \begin{pmatrix} 32 \\ 23 \\ 33 \\ 31 \end{pmatrix}, \delta A = \begin{pmatrix} 0 & 0 & 0.1 & 0.2 \\ 0.08 & 0.04 & 0 & 0 \\ 0 & -0.02 & -0.11 & 0 \\ -0.01 & -0.01 & 0 & -0.02 \end{pmatrix} \text{ et } \delta b = \begin{pmatrix} 0.01 \\ -0.01 \\ 0.01 \\ -0.01 \end{pmatrix}.$$

a) Calculer grâce à Scilab le conditionnement de A pour la norme 2.

b) Résoudre grâce à Scilab les systèmes $AX_1 = b$, $(A + \delta A)X_2 = b$ et $AX_3 = b + \delta b$.

c) Calculer $\|\delta b\|_2/\|b\|_2$,$\|\delta A\|_2/\|A\|_2$, $\|X_1 - X_2\|_2/\|X_1\|_2$ et $\|X_1 - X_3\|_2/\|X_1\|_2$.

d) Conclure.

Exercice 4: Elimination de Gauss

a) Programmer l'algorithme de remontée pour résoudre $Uz = b$ où U est une matrice triangulaire supérieure. On utilisera la même vecteur pour stocker b et z. Valider le programme.

b) Programmer l'algorithme de descente pour résoudre $Lx = z$ où L est une matrice triangulaire inférieure. On utilisera la même vecteur pour stocker z et x. Valider le programme.

c) Programmer la méthode de Gauss sans pivot. Valider le programme.

d) Programmer la méthode de Gauss avec pivot. On utilisera la fonction *max*. Valider le programme.

Exercice 5: Mise en évidence de l'importance du pivotage

a) Construire une matrice aléatoire de taille $n \times n$ et un vecteur x aléatoire de taille n pour un n donné. Calculer $b = Ax$.

b) Pour un n donné, résoudre le système $Ax = b$ précédent (dont on connait la solution) grâce aux méthodes de Gauss avec et sans pivot.

c) Faire une boucle sur les n, et pour chacun de ces n effectuer la procédure **a)-b)** 20 fois. Représenter le logarithme de l'erreur entre la solution 'exacte' et la solution numérique en fonction de n.

d) Tester sur :

i) $A =$ une matrice aléatoire.

ii) $A =$ la matrice de Hilbert. Observer son conditionnement.

iii) $A =$ une matrice orthogonale. Calculer son conditionnement en norme 2.

iv) $A =$ la matrice du Laplacien.

Exercice 6: Factorisation LU

a) Calculer (sur le papier) $\rho = \frac{\max_{i,j} |U_{i,j}|}{\max_{i,j} |A_{i,j}|}$, le taux d'accroissement de la matrice:

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 & 1 \\ -1 & -1 & 1 & 0 & 1 \\ -1 & -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 \end{pmatrix}.$$

b) Tracer en fonction de n le taux d'accroissement d'une matrice aléatoire $n \times n$. Comparer avec la courbe $n \to \sqrt{n}$. On utilisera les fonctions "élimination de Gauss avec pivot" (puisque l'on n'a besoin que de la matrice U).

c) Etudier les fonctions qui existent en Scilab pour calculer la factorisation LU.

d) Programmer la factorisation LU (sans pivotage).

e) Ecrire en pseudo-code la version 'ligne' de la factorisation LU.

Exercice 7: Factorisation de Cholesky

a) Programmer la factorisation de Cholesky.

b) Etudier les fonctions qui existent en Scilab pour calculer la factorisation de Cholesky.