Faculté des sciences et ingénierie (Toulouse III) **Année universitaire**
Département de mathématiques – L3 ESR **2018-2019**
U.E. Analyse Numérique

Feuille de TP2
Algèbre linéaire : Méthodes directes, Méthodes itératives

Exercice 1. *(Résolution de systèmes triangulaires, décomposition LU)*

1. *Programmer deux fonctions, dont chacune prend en entrée une matrice A triangulaire inférieure (resp. triangulaire supérieure) et un vecteur b et qui résoud le système $Ax = b$ par descente (resp. par remontée) et renvoie le vecteur solution x en sortie. Tester-le sur une matrice triangulaire (inversible bien sur) choisie au hasard.*
2. *Vérifier l'hypothèse des mineurs principaux pour la matrice* $A = \begin{pmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{pmatrix}$.

   Que peut-on en déduire ?
3. *Utiliser la décomposition LU pré-programmée en Python (commande* **lu** *de la bibliothèque scipy.linalg). Qu'obtient-on ? Utiliser cette décomposition et les fonctions de remontée et de descente programmées à la question précédente pour résoudre le système de matrice A et de second membre le vecteur ne contenant que des 1.*

Exercice 2. *(Décomposition de Cholesky)*

1. *Écrire l'algorithme qui permet de calculer la décomposition de Cholesky d'une matrice A. Pour cela, si L est une matrice triangulaire inférieure de coefficients b_{ij}, $1 \leq i, j \leq n$, écrire l'égalité $A = LL^T$ entraine que les coeffs de L sont donnés par :*

$$\begin{cases} \ell_{1,1} = \sqrt{a_{1,1}},\ \forall j \geq 2,\ \ell_{j,1} = \frac{a_{j,1}}{\ell_{1,1}}, \\ \forall j \geq 2, \forall i \geq j+1, \quad \ell_{i,j} = \frac{a_{i,j} - \sum_{k=1}^{j-1} \ell_{i,k}\overline{\ell_{j,k}}}{\ell_{j,j}} \end{cases}$$

2. *Le programmer et l'appliquer à la matrice* $A = \begin{pmatrix} 1 & 1 & -1 & 1 \\ 1 & 3 & -1 & 5 \\ -1 & -1 & 2 & -2 \\ 1 & 5 & -2 & 13 \end{pmatrix}$. *Pour cela on commencera par vérifier que A est bien symétrique définie positive (on pourrait pour cela utiliser la commande* **eig** *de la bibliothèque numpy.linalg). Que calcule la commande* **chol** *de numpy.linalg ?*

Exercice 3. *(Étude comparée de méthodes itératives)*
On reprend ici l'exercice 1 du TD 3 sur les méthodes itératives.

1. *Soit*
$$A = \begin{pmatrix} 1 & 2 & -2 \\ 1 & 1 & 1 \\ 2 & 2 & 1 \end{pmatrix}.$$
Calculer grâce à Scilab $\rho(J)$ *et* $\rho(G)$ *et vérifier que* $\rho(J) < 1 < \rho(G)$. *Que peut-on en déduire pour les méthodes de Jacobi et Gauss Seidel ?*
2. *Soit*
$$A = \begin{pmatrix} 2 & -1 & 1 \\ 2 & 2 & 2 \\ -1 & -1 & 2 \end{pmatrix}.$$
Calculer grâce à Scilab $\rho(J)$ *et* $\rho(G)$. *Que peut-on en déduire pour les méthodes de Jacobi et Gauss Seidel ?*
3. *Implémenter une fonction Jacobi qui prend en entrée la matrice* A, *le second membre* b *et une tolérance* ε *et renvoie* x *est le vecteur construit par la méthode de Jacobi et qui approche la solution de l'équation "à tol près" en choisissant un critère d'arrêt de résidu. Appliquer cette méthode au cas précédent qui vous paraît le plus judicieux.*