Analyse Numérique Licence L3 de Mathématique 2015-2016, TP7

Soit $A \in \mathcal{M}_n(\mathbb{R})$ et $b \in \mathbb{R}^n$. On va programmer l'algorithme de factorisation $PA = LU$ avec pivotage ainsi que la descente remontée pour résoudre le système linéaire $Ax = b$.

(1) Compléter la fonction *[A,P]=LUAvecPivotage(A,n)* mettant en oeuvre l'algorithme de factorisation $PA = LU$ avec pivotage avec stockage dans A suivant :

Initialisation : $P = (1, \cdots, n)$
For $k = 1, \cdots, n-1$ (boucle sur les pivots)
$i_k = \mathrm{argmax}_{i=k,\cdots,n}|A_{i,k}|$ (choix du pivot) transposition : $\tau = (k, i_k)$, $i_k \geq k$
$A \leftarrow \tau A =$ permutation des lignes k et i_k
$P \leftarrow \tau P$ mise à jour de la permutation $= P(k) \leftarrow P(i_k)$ et $P(i_k) \leftarrow P(k)$
For $i = k+1, \cdots, n$
$$A_{i,k} \leftarrow \frac{A_{i,k}}{A_{k,k}}$$
End For
For $i, j = k+1, \cdots, n$
$$A_{i,j} \leftarrow A_{i,j} - A_{i,k}A_{k,j}$$
End For
End For

(2) Compléter la fonction *x=SolLUAvecPivotage(A,P,n,b)* mettant en oeuvre l'algorithme suivant de descente remontée avec A contenant la factorisation LU avec pivotage précédente et P la matrice de permutation.

Descente : $Lz = Pb$

$z = 0$
For $i = 1, \cdots, n$
$z_i = b_{P(i)} - \sum_{j=1}^{i-1} A_{i,j}z_j$
End For
Remontée : $Ux = z$

$x = 0$
For $i = n, \cdots, 1$;
$$x_i = \frac{z_i - \sum_{j=i+1}^{n} A_{i,j}x_j}{A_{i,i}}$$
End For

(3) Appliquer l'algorithme avec pivotage pour la résolution de $Ax = b$ sur la matrice

$$A = \begin{pmatrix} \delta & 1 \\ 1 & 1 \end{pmatrix}$$

avec $\delta = 10^{-14}$ et le second membre

$$b = \begin{pmatrix} \delta + 1 \\ 2 \end{pmatrix}$$

La solution exacte est

$$y = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Calculer l'erreur $\|x - y\|$. Comparer avec ce que l'on obtient dans le cas de l'algorithme sans pivotage.
Appliquer l'algorithme avec pivotage à la matrice

$$A = \begin{pmatrix} 1 & -2 & 1 \\ 3 & -6 & 2 \\ 4 & 1 & 5 \end{pmatrix}$$

et

$$b = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}$$

Comparer avec ce que l'on obtient dans le cas de l'algorithme sans pivotage.

(4) Créer la fonction *[A,Q]=LUAvecPivotageCol(A,n)* mettant en oeuvre l'algorithme de factorisation $AP = LU$, $Q = P^{-1}$ avec pivotage des colonnes et avec stockage dans A suivant :

Initialisation : $Q = (1, \cdots, n)$
For $k = 1, \cdots, n-1$ (boucle sur les pivots)
$j_k = \text{argmax}_{j=k,\cdots,n}|A_{k,j}|$ (choix du pivot) transposition : $\tau = (k, j_k)$, $j_k \geq k$
$A \leftarrow A\tau =$ permutation des colonnes k et j_k
$Q \leftarrow \tau Q$ mise à jour de la permutation $= Q(k) \leftarrow Q(j_k)$ et $Q(j_k) \leftarrow Q(k)$
For $i = k+1, \cdots, n$
$A_{i,k} \leftarrow \dfrac{A_{i,k}}{A_{k,k}}$
End For
For $i, j = k+1, \cdots, n$
$A_{i,j} \leftarrow A_{i,j} - A_{i,k}A_{k,j}$
End For
End For

(5) Créer la fonction *x=SolLUAvecPivotageCol(A,Q,n,b)* mettant en oeuvre l'algorithme suivant de descente remontée avec A contenant la factorisation LU avec pivotage des colonnes précédente et Q l'inverse de la permutation P.

Descente : $Lz = b$

$z = 0$
For $i = 1, \cdots, n$

$z_i = b_i - \sum_{j=1}^{i-1} A_{i,j} z_j$
End For
Remontée : $Uy = z$

$y = 0$
For $i = n, \cdots, 1$;

$$y_i = \frac{z_i - \sum_{j=i+1}^{n} A_{i,j} y_j}{A_{i,i}}$$

End For
Résolution de $Qx = y$

For $i = 1, \cdots, n$
$x_{Q(i)} = y_i$
End For

(6) Reprendre les tests précédents avec la factorisation avec pivotage des colonnes.