TD 3 - Décomposition LU et résolution de systèmes linéaires

1 Introduction

Le but de l'exercice est de programmer une résolution de système linéaire par décomposition LU. On se placera dans les conditions les plus favorables en limitant les tests et cas particuliers au traitement d'un pivot de Gauss simple par inversion de lignes.

Le système s'écrit :

$$A\,X = B$$

où A est une matrice $n \times n$ est B un vecteur de taille n. On suppose que $\det A \neq 0$, mais ça ne veut pas dire qu'elle est bien conditionnée...

On rappelle ci-dessous les principales équations.

2 Décomposition

Dans un première étape, il faut trouver une matrice triangulaire supérieure U et une matrice triangulaire inférieure L telles que :

$$PA = LU$$

où P est une matrice de permutation. En pratique, on tiendra compte des permutations de lignes en initialisant un vecteur S de taille n tel que :

$$s_i = i$$

et en échangeant deux composantes de S, s_i et s_j à chaque fois qu'on échange les lignes i et j.

L et U obéissent aux contraintes suivantes :

$$l_{ij} = 0 \quad j > i$$

$$u_{ij} = 0 \quad j < i$$

On choisi de normaliser L et U en imposant :

$$u_{ii} = 1.0$$

Il y a donc n^2 inconnues à déterminer, et n^2 équations :

$$a_{ij} = \sum_{k=1}^{n} l_{ik} \cdot u_{kj}$$

On commence par la première colonne de L qui donne :

$$a_{i1} = l_{i1} \Rightarrow l_{i1} = a_{i1}$$

Puis la première ligne de U qui donne (pour $j = 2$ à n) :

$$a_{1j} = l_{11} \cdot u_{1j} \Rightarrow u_{1j} = \frac{a_{1j}}{l_{11}}$$

La deuxième colonne de L donne (pour $i = 2$ à n) :

$$a_{i2} = l_{i1} \cdot u_{12} + l_{i2} \Rightarrow l_{i2} = a_{i2} - l_{i1} \cdot u_{12}$$

La deuxième ligne de U donne (pour $j = 3$ à n) :

$$a_{2j} = l_{21} \cdot u_{1j} + l_{22} \cdot u_{2j} \Rightarrow u_{2j} = \frac{1}{l_{22}} \left(a_{2j} - l_{21} \cdot u_{1j}\right)$$

Plus généralement, la colonne j de L donne (pour $i = j$ à n) :

$$a_{ij} = \sum_{k=1}^{j-1} l_{ik} \cdot u_{kj} + l_{ij} \Rightarrow l_{ij} = a_{ij} - \sum_{k=1}^{j-1} l_{ik} \cdot u_{kj}$$

et la ligne i de U donne (pour $j = i + 1$ à n) :

$$a_{ij} = \sum_{k=1}^{i} l_{ik} \cdot u_{kj} \Rightarrow u_{ij} = \frac{1}{l_{ii}} \left(a_{ij} - \sum_{k=1}^{i-1} l_{ik} \cdot u_{kj}\right)$$

3 Substitution

Une fois les matrices U et L obtenues, on peut passer à la substitution :

$$y_i = \frac{1}{l_{ii}} \left(b_i - \sum_{j=1}^{i-1} l_{ij} \cdot y_j \right)$$

Puis :

$$x_i = y_i - \sum_{j=i+1}^{n} u_{ij} \cdot x_j$$

4 Pivot

On voit que le calcul des u_{ij} nécessite une division par l_{ii}. Ceci exige évidemment que celui-ci soit différent de 0. Avant de commencer le calcul d'une nouvelle colonne de l_{ij} à j fixé (i variant de j à n), on va donc échanger la ligne numéro j avec une ligne k telle que $|a_{kj}| = \sup_{i \geqslant j} |a_{ij}|$. Il faut bien sûr échanger **TOUTE** la ligne ! Et pas seulement les coefficients pas encore calculés. on inverse aussi les deux éléments concernés de S afin de pouvoir échanger les coefficients de B lors de l'opération de substitution.

Note Dans la mise au point du programme, on commencera par une première version (partielle) sans s'occuper de cette substitution. Elle est simple à ajouter une fois que le reste fonctionne. En revanche, le debugage est beaucoup plus compliqué si on cherche à tout faire à la fois. Il faut juste veiller à faire les tests avec une matrice qui ne demande pas de pivot.

5 Exemples

Testez votre programme sur les systèmes suivants :

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 3 \\ 7 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \\ 3 \end{pmatrix}$$

Prévoyez d'écrire une routine qui imprime une matrice à l'écran dans un format commode.