MM031 - TP5
Méthodes directes de résolution d'un système linéaire (2)

Maxime Chupin : `chupin@ann.jussieu.fr`

02 février 2016

Le but de ce TP est d'implémenter les décompositions LU et de Cholesky pour la résolution de système linéaire par méthode directe. Nous utiliserons les classes écrites au cours des TPs précédents.

Exercice 1 : Décomposition LU

La décomposition LU est principalement une décomposition de Gauss-Jordan modifiée. On transforme une matrice carrée A en une matrice triangulaire supérieure en éliminant les valeurs en dessous de la diagonale. L'algorithme qu'on présente ici dans ce TP effectue l'élimination colonne par colonne en commençant à gauche, en multipliant à gauche la matrice A par en matrice triangulaire inférieure.

Cet algorithme produit une unique matrice triangulaire inférieure unitaire L et une matrice triangulaire supérieure U.

Description de l'algorithme. Soit A notre matrice carrée de taille $N \times N$. On note par a_{ij}, $1 \leqslant i, j \leqslant N$, les éléments de la matrice.

Initialisation : On définit $A^{(0)} = A$.

Boucle sur les colonnes : On élimine les éléments de la n-ème colonne de $A^{(n-1)}$ en dessous de la diagonale en ajoutant à la i-ème ligne de cette matrice, la n-ème ligne multipliée par

$$l_{i,n} = -\frac{a_{i,n}^{(n-1)}}{a_{n,n}^{(n-1)}},$$

i allant de $n+1$ à N. Ceci peut être effectué par la multiplication à gauche de $A^{(n-1)}$ par

$$L_n = \begin{pmatrix} 1 & & & & & 0 \\ & \ddots & & & & \\ & & 1 & & & \\ & & l_{n+1,n} & \ddots & & \\ & & \vdots & & \ddots & \\ 0 & & l_{N,n} & & & 1 \end{pmatrix}.$$

On définit alors $A^{(n)} = L_n A^{(n-1)}$.

Conclusion : Après $N-1$ itération, on a obtenu une matrice triangulaire supérieure $A^{(N-1)}$, et on a :

$$A = L_1^{-1} \cdots L_{N-1}^{-1} A^{(N-1)},$$

et parce que l'inverse et le produit de matrices triangulaires inférieures sont des matrices triangulaires inférieures, on obtient la décomposition $A = LU$ où

$$U = A^{N-1} \text{ et } L = \begin{pmatrix} 1 & & & & & 0 \\ -l_{2,1} & \ddots & & & & \\ & & 1 & & & \\ \vdots & & -l_{n+1,n} & \ddots & & \\ & & \vdots & & 1 & \\ -l_{N,1} & & -l_{N,n} & & -l_{N,N-1} & 1 \end{pmatrix}.$$

Bien entendu, pour appliquer un tel algorithme, il est nécessaire que les éléments $a_{n,n}^{n-1}$ soient différents de zéro à chaque étapes de l'algorithme. Si ça n'était pas le cas, puisque la matrice considérée est inversible, il existe une ligne j pour laquelle l'élément $a_{j,n}^{(n-1)} \neq 0$. Il faudra donc interchanger les deux lignes.

Conseil : Pour implémenter l'algorithme, on écrira que

$$\forall (i,j) \in \{1,\ldots,N\}^2,\ a_{ij} = \sum_{k=1}^{??} l_{ik} u_{kj},$$

où il faut déterminer ce que vaut ??.

1. Écrire une procédure

```
void decomp_LU(const matrice & M, matrice & L, matrice & U);
```

qui construit dans L et U la décomposition LU de M.

2. Résoudre

$$\begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix} \cdot y = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

3. Résoudre le système $Hy = b$ où H est une matrice de Hilbert : c'est-à-dire une matrice carrée de taille $N \times N$ de terme générale

$$H_{i,j} := \frac{1}{i+j-1}$$

et b le vecteur de taille N où $b_i = 1$. (On fera en sorte de pouvoir essayer ces résolutions pour différentes valeurs de N).

Exercice 2 : Décomposition de Cholesky

Lorsque la matrice du système linéaire que l'on souhaite résoudre est *symétrique définie positive*, alors il existe une matrice réelle triangulaire inférieure L telle que :

$$A = LL^T.$$

On peut également imposer que les éléments diagonaux de la matrice L soient tous positifs, alors la factorisation correspondante est unique.

Légère description de l'algorithme. On cherche la matrice :

$$L = \begin{pmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{pmatrix}$$

En utilisant la relation :

$$\forall 1 \leqslant i,j \leqslant N, \quad a_{ij} = \sum_{k=1}^{N} l_{ik} l_{jk} = \sum_{ij}^{??} l_{ik} l_{jk},$$

où ?? est à déterminer, on va calculer les éléments de L colonne par colonne.

1. Écrire une fonction

```
matrice Cholesky(const matrice & M);
```

   qui renvoie la matrice de Cholesky de M.

2. Résoudre

$$\begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix} \cdot y = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

3. Résoudre le système $Hy = b$ où H est une matrice de Hilbert : c'est-à-dire une matrice carrée de taille $N \times N$ de terme générale

$$H_{i,j} := \frac{1}{i+j-1}$$

   et b le vecteur de taille N où $b_i = 1$. (On fera en sorte de pouvoir essayer ces résolutions pour différentes valeurs de N).

4. En utilisant les fonctions `clock()` de la librairie standard `<ctime>`, comparer les temps d'exécution des deux méthodes. Voici un exemple d'utilisation :

```
#include<ctime>
int main(){
   clock_t start, end;
   double msecs;

   start = clock();
   /* any stuff here ... */
   end = clock();
   msecs = ((double) (end - start)) / CLOCKS_PER_SEC;
   return 0;
}
```