Université Paris I, Panthéon - Sorbonne

Deuxième Année Licence M.I.A.S.H.S. 2016 – 2017

TP de Méthodes Numériques n^0 9 : Résolution de systèmes linéaires, calcul de matrices inverses et de déterminants

On s'intéresse dans ce TP aux applications possible avec un logiciel numérique en algèbre linéaire. Nous avons déjà utilisé certaines fonctionnalités (commandes `solve`, `det` par exemple), nous allons voir d'un peu plus près ce qui se cache dessous...

L'algorithme de décomposition LU

On commence par mettre en place un algorithme permettant de résoudre un système d'équations linéaires. On se place dans le cadre simple d'une matrice A inversible et dont tous les pivots vont être supposés non nuls. On met en place un programme avec en entrée la matrice A et on doit donner en sortie les matrices triangulaires inférieure et supérieure L et U telles que $A = L\,U$, avec en plus l'exigence que sur la diagonale de L il n'y ait que des 1. Une proposition d'un tel programme est la suivante:

```
AlgoLU <-function(A)
{n=nrow(A)
L=diag(1,n,n)
U=diag(0,n,n)
for (k in c(1:(n-1)))
{for (i in c((k+1):n))
{A[i,k]=A[i,k]/A[k,k]
j=c((k+1):n)
A[i,j]=A[i,j]-A[i,k]*A[k,j]}}
U[1,c(1:n)]=A[1,c(1:n)]
for (i in c(2:n))
{L[i,c(1:(i-1))]=A[i,c(1:(i-1))]
U[i,c(i:n)]=A[i,c(i:n)]}
return(U)
return(L)}
if (interactive()) AlgoLU()
```

Comprendre ce qui a été fait. Pour cela, supposer que $n = 3$, écrire les équations donnant les composantes de L et de U, notées en (ℓ_{ij}) et (u_{ij}) en fonction de celles de A, notées a_{ij}. Vérifiant que l'algorithme donne exactement ce que l'on désire.
Appliquer l'algorithme à des matrices A de tailles $n = 2$, $n = 3$, $n = 10$ et $n = 1000$ en vérifiant à chaque fois que $L\,U = A$.
Ecrire les algorithmes dits de remontée qui permettent de résoudre $A\,X = B$, où B est un vecteur colonne quelconque donné, en résolvant $L\,Y = B$ puis $U\,X = Y$ (ce qui revient à inverser L et U). Dans les 2 cas, écrire les équations puis les coder en R. Appliquer ces algorithmes aux différentes matrices A considérées ci-dessus. Utiliser également la commande `solve` pour résoudre ces systèmes.
On désire maintenant traiter des cas où $A[k,k] = 0$ dans le programme précédent (algorithe de décomposition LU), ce qui peut advenir lorsque toutes les matrices principales extraites de A ne sont pas inversibles. Pour cela, à chaque étape k, commencer par chercher le plus grand coefficient $|a_{ik}|$ où $i \geq k$, par exemple noté $|a_{i_0k}|$, puis échanger les lignes k et i_0. Appliquer à nouveau votre algorithme sur les matrices précédentes puis sur la matrice de Hilbert définie dans le TP3. Mémoriser dans votre algorithme les permutations effectuées, ce qui permettra ensuite de résoudre des systèmes ou inverser la matrice.

Inverse de matrice et déterminant

De ce qui précède, il est aisé de proposer une méthode permettant d'obtenir l'inverse de la matrice A. Cela peut par exemple être fait en utilisant les matrices L^{-1} et U^{-1} obtenus dans les algorithmes de remontée.

Le calcul du déterminant de A est facilement obtenu en écrivant $\det(A) = \det(L)\det(U) = \prod_{i=1}^{n} u_{ii}$ lorsque la décomposition de A ne nécessite pas de permutation de lignes. Mettre en place un

Exercices

Exercice 1. Pour $n \in \mathbf{N}^*$, on considère une matrice de Vandermonde, c'est-à-dire que l'on choisit $(\alpha_1, \cdots, \alpha_n)$, n réels distincts, tels que $A = (\alpha_i^{j-1})_{1\leq i,j\leq n}$. Démontrer que $\det(A) = \prod_{1\leq i<j\leq n}(\alpha_j - \alpha_i)$. Déterminer numériquement ce déterminant quand $\alpha_i = 1/i$ pour différentes valeurs de n.

Exercice 2. On considère les systèmes linéaires $A_i x_i = b_i$, pour $i = 1, 2, 3$, avec

$$A_1 = \begin{pmatrix} 15 & 6 & 8 & 11 \\ 6 & 6 & 5 & 3 \\ 8 & 5 & 7 & 6 \\ 11 & 3 & 6 & 9 \end{pmatrix}$$

$A_i = A_1^i$ pour $i = 2, 3$ et déterminer les b_i tels que la solution des systèmes soient toujours $x_i = {}^t(1, 1, 1, 1)$. Résoudre ensuite le système (comme si les x_i étaient inconnus). Commenter les résultats obtenus.

Exercice 3. On considère le système linéaire $A\,x = b$ avec

$$A = \begin{pmatrix} 2 & -2 & 0 \\ \varepsilon - 2 & 2 & 0 \\ 0 & -1 & 3 \end{pmatrix}$$

où b est tel que la solution du système soit $x = {}^t(1, 1, 1)$ et $\varepsilon \geq 0$. Effectuer la résolution du système, en faisant comme si x était inconnu, pour différentes valeurs de ε tendant vers 0. Vérifier numériquement que le terme ℓ_{32} devient très grand. Le démontrer théoriquement. Vérifier que la solution n'est pas affectée par des erreurs d'arrondi quand $\varepsilon = 10^{-10}$. Et si ε est encore plus petit?