Université René Descartes Paris 5 — Licence de mathématiques
UFR de mathématiques et informatique — Licence MASS

Méthodes numériques

Partiel du 4 avril 2003

Durée 2h
Documents autorisés : documents distribués et notes personnelles de cours
Calculettes et téléphones portables interdits

I

Calculer l'inverse de la matrice $\begin{bmatrix} 2 & 2 & 3 \\ 1 & 1 & 3 \\ 2 & 3 & 2 \end{bmatrix}$ par la méthode de Gauss-Jordan. On appliquera l'algorithme vu en cours utilisant en mémoire une seule matrice 3×3.

II

SCILAB exécute les instructions suivantes. Donner l'affichage obtenu à l'écran.

```
u=[2,3,2]
A=u*u'
B=u'*u
u=[u,5,2]
C=[u;u;u;u]
D=C; D(2,:)=D(2,:)/D(2,3)
for j=1:5, D(3,j)=D(3,j)/D(3,3); end
for j=5:-1:1, D(4,j)=D(4,j)/D(4,3); end
D
for i=1:4, for j=1:4, E(i,j)=i+j; end, end,E
F=E; F(2:3,2:3)=zeros(2,2)
G=F; G(3:5,3:6)= ones(3,4)
for k=1:3:9, H(k:k+2,k:k+2)=k*ones(3,3); end,H
K=H; for i=[1,3:5,8:9], K(i,:)=-K(i,:); end,K
L=H; for j=2:2:9, L(:,j)=-L(:,j);  end,L
M=[4*ones(2,2),zeros(2,2),[5*[1:2];5*[2:-
1:1]];zeros(2,1),6*ones(2,4),zeros(2,1)]
```

III

L'algorithme de résolution de système linéaire par la méthode de Gauss vu en cours, dont le code est rappelé ci-après, ne traite que le cas où la matrice A est carrée et régulière.

```
function X = gauss0(A,B)
n=size(A,'c')
  if ~ size(A,'r')==n  then printf('matrice non carree'), end
  A(:,n+1) = B // ou A=[A,B]
  print(6,A),

  for k = 1 : n ,   // etape k
    if A(k,k)==0 then // recherche d'un pivot non nul
        printf('pivot nul a l''etape %d',k),
        if k<n then i=k+1,
                    while(A(i,k)==0 & i<n ), i=i+1,
                    end,
               else i=k,
        end,
        if A(i,k)==0 then X='matrice singuliere', return
                     else printf('on echange les lignes %d et %d',k,i),
                          A([k,i],:)=A([i,k],:),
                          print(6,A),
        end,
    end,
    printf('pivot=%f',A(k,k)),
```

```
    // for i = k+1 : n ,   // nouvelle ligne i
    //   A(i,k:n+1) = A(i,k:n+1) - A(i,k)*A(k,k:n+1)/A(k,k)
    // end,
    //// mieux
    A(k+1:n,k:n+1) = A(k+1:n,k:n+1) - A(k+1:n,k)*A(k,k:n+1)/A(k,k)
    printf('etape %d',k),
    print(6,A),
end,

X(n) = A(n,n+1)/A(n,n),
for i=[n-1:-1:1],   // remontee : calcul des xi
    X(i) = (A(i,n+1) - A(i,i+1:n)*X(i+1:n))/A(i,i),
end,
printf('solution'),
```

1. Pourquoi est-il inutile de vérifier si la matrice est régulière avant de lancer le programme ?
Que se passe-t-il si elle est singulière ?

2. On sait que, si la matrice est singulière, le système est impossible ou indéterminé.
Comment reconnaître ces deux cas dans l'algorithme ?

3. Compléter l'algorithme pour traiter ces deux cas de la façon suivante :
- s'il y a impossibilité, on écrit un message et on s'arrête au moment où on détecte l'impossibilité ;
- si une variable est indéterminée, on lui donne une valeur arbitraire (par exemple 0 ou 10000 ou une valeur aléatoire), on le signale par un message et on continue le calcul.

4. Compléter le programme SCILAB. On appellera `gauss0modifiee` la nouvelle fonction.

5. Application : exécuter les instructions suivantes (on affichera ce qui serait affiché par SCILAB) :

```
A= [2 2 3 ; 1 1 3 ; 2 2 1]
B1= [1;2;-1], B2=[1;2;2]
X1=gauss0modifiee(A,B1); X2=gauss0modifiee(A,B2) ;
X0=gauss0modifiee(A,zeros(3,1));
```

Donner les valeur de `X1, X2` et `X0` données par ce programme.

Donner, sans calcul, la valeur de `A*(X1+567898*X0)` et `A*(X2+567898*X0)` (si possible).

IV

On appelle matrice en croix une matrice carrée de dimension `n` dont tous les éléments sont nuls sauf ceux des deux diagonales.

Exemple : $\begin{bmatrix} 3 & 0 & 0 & 0 & 8 \\ 0 & 2 & 0 & 7 & 0 \\ 0 & 0 & 7 & 0 & 0 \\ 0 & 6 & 0 & 4 & 0 \\ 7 & 0 & 0 & 0 & 3 \end{bmatrix}$

1. Le produit de deux matrices en croix est-il une matrice en croix ?

2. On considère un algorithme de multiplication de deux matrices en croix qui tient compte de l'existence de 0 (pas de multiplication par 0 ou d'addition de 0).

2.1. Programmer cet algorithme en SCILAB.

2.2. Calculer la complexité de cet algorithme.

3. Même question pour l'algorithme de résolution d'un système linéaire par la méthode de Gauss pour une matrice en croix.