Université de Picardie Jules Verne
Master 2 AAM
Calcul Scientifique
2019-2020

Méthodes directes

1 Commandes Scilab

- Partie triangulaire supérieure : `triu(A)`
- Partie triangulaire inférieure : `tril(A)`
- Factorisation de Cholesky : `[R]=chol(A)`
- Factorisation de LU : `[R]=lu(A)`
- Factorisation QR : `[Q,R]=qr(A)`
- Moindres carrés : `lsq(A)`
- Calcul du temps de calcul

```
tic
⋮
instructions
⋮
toc
```

2 Méthodes classiques

Exercice 1 (Factorisation LU et de Cholesky)

1. Programmer la méthode de factorisation LU, telle que présentée en cours
2. Produire un nouvelle version du programme en essayant d'utiliser un maximum de produits scalaires.
3. Pour une matrice A SDP donnée, calculer les temps de calculs respectifs. Conclusion ?
4. Mêmes questions que précédemment mais pour la méthode de Cholesky.

Exercice 2 (Factorisation LU d'une matrice tridiagonale)
On s'intéresse au cas particulier d'une matrice tridiagonale

$$A = \begin{pmatrix} a_1 & c_1 & & 0 \\ b_2 & a_2 & \ddots & \\ & \ddots & & c_{n-1} \\ 0 & & b_n & a_n \end{pmatrix}$$

Les matrices L et U sont de la forme

$$L = \begin{pmatrix} 1 & & & 0 \\ \beta_2 & 1 & \ddots & \\ & \ddots & \ddots & \\ 0 & & \beta_n & 1 \end{pmatrix} \text{ et } U = \begin{pmatrix} \alpha_1 & c_1 & & 0 \\ & \alpha_2 & \ddots & \\ & \ddots & & c_{n-1} \\ 0 & & & \alpha_n \end{pmatrix}$$

1. Montrer que les coefficients α_i et β_i satisfont les relations :

$$\boxed{\alpha_1 = a_1,\ \beta_i = \frac{b_i}{\alpha_{i-1}},\ \alpha_i = a_i - \beta_i c_{i-1},\ i = 2, \cdots, n}$$

2. Programmer la méthode sur la matrice du schéma à trois points.

Exercice 3 (Factorisation LU par blocs et applications)
Soient A, B, C et D des matrices à coefficients réels, de tailles respectives $n \times n$, $n \times p$, $p \times n$ et $p \times p$. On considère la matrice construite M par blocs comme suit :

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix}.$$

1. On suppose que A est inversible. Déterminer les matrices X, Y, Z et V, de tailles respectives $p \times n$, $n \times n$, $p \times p$ et $n \times p$ telles que

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix} = \begin{pmatrix} I_n & 0_{n,p} \\ X & I_p \end{pmatrix} \begin{pmatrix} Y & V \\ 0_{p,n} & Z \end{pmatrix} = LU,$$

   où I_m est la matrice identité de taille m (pour $m = p, n$) et $0_{k,m}$ est la matrice nulle de taille $k \times m$, avec $k = n, p$ et $m = n, p$.

2. Sous quelles conditions portant sur A, B, C et D, la matrice M est-elle inversible ?
3. Programmer la méthode en Scilab.

3 Méthodes particulières

Exercice 4 (Algorithme de Thomas)
Soit A la matrice diagonale :

$$A = \begin{pmatrix} b_1 & c_1 & & & \\ a_2 & b_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & a_n & b_n \end{pmatrix}$$

On considère le système linéaire $Au = f$. L'idée de la méthode de Thomas consiste à appliquer l'élimination de Gauss au système. On pose

$$c'_i = \begin{cases} \dfrac{c_i}{b_i} & i = 1 \\ \dfrac{c_i}{b_i - a_i c'_{i-1}} & i = 2, \cdots, n \end{cases} \quad \text{et } f'_i = \begin{cases} \dfrac{f_i}{b_i} & i = 1 \\ \dfrac{f_i - a_i f'_i}{b_i - a_i c'_{i-1}} & i = 2, \cdots, n \end{cases}$$

La solution se calcule alors par remontée

$$\begin{aligned} & x_n = f'_n, \\ & u_i = f'_i - c'_i x_{i+1}, i = n-1, \cdots, 1. \end{aligned}$$

1. Programmer cette méthode.
2. Comparer le résultat obtenu avec une autre méthode directe de votre choix.

Exercice 5 (Réduction cyclique)
Considérons le système $Au = f$ où A est la matrice $N \times N$:

$$A = \begin{pmatrix} \alpha & -1 & 0 & . & . & 0 \\ -1 & \alpha & -1 & 0 & . & . \\ 0 & -1 & \alpha & -1 & . & . \\ . & . & . & . & . & 0 \\ . & . & 0 & -1 & \alpha & -1 \\ 0 & . & . & 0 & -1 & \alpha \end{pmatrix}.$$

On suppose que $N = 2^p - 1$. Le système est équivalent à

$$\begin{cases} -u_{i-1} + \alpha u_i - u_{i+1} = f_i, \quad 1 \leq i \leq N \\ u_0 = f_0, \; u_N = f_{N+1} \end{cases}$$

Considérons les trois équations successives du système :

$$\begin{cases} -u_{2i-2} + \alpha u_{2i-1} - u_{2i} = f_{2i-1} \\ -u_{2i-1} + \alpha u_{2i} - u_{2i+1} = f_{2i}, \\ -u_{2i} + \alpha u_{2i+1} - u_{2i+2} = f_{2i+1} \end{cases}$$

Multiplions la seconde équation par α et sommons alor les trois équations :

$$\begin{cases} -u_{2i-2} + \alpha^{(1)} u_{2i} - u_{2i+2} = f_2^{(1)} i, \; i = 1, 2, \cdots 2^{p-1} - 1 \\ u_0 = f_0, \; u_{N+1} = u_{2^p} = f_{N+1} \end{cases}$$

où l'on a posé

$$\left\{ \begin{array}{l} \alpha^{(1)} = \left(\alpha^{(0)}\right)^2 - 2, \\ f_2^{(1)} i = f_{2i-1} + \alpha^{(0)} f_{2i} + f_{2i+1} \end{array} \right.$$

A ce point, la résolution du système de départ peut se décomposer en deux temps :

i Résolution du système pour les indices pairs

ii Calcul explicite des inconnues d'indice impair par

$$u_{2i+1} = \frac{1}{\alpha^{(0)}} \left(f_{2i+1}^{(0)} + u_{2i} + u_{2i+2} \right)$$

L'étape [i] peut se résoudre suivant un plan similaire au plan global, (i.e.) récursivement, ce qui simplifie davantage la résolution du système. On résume la procédure comme suit :

- **Initialisation et construction des suites**

$$\alpha^{(0)} = \alpha, \; f_i^{(0)} = f_i, \; i = 1, \cdots N$$

$$\begin{array}{l} \text{pour} k = 1, ..p-1 \\ \alpha^{(k)} = \left(\alpha^{(k-1)}\right)^2 - 2, \\ f_i^{(k)} = f_{i-2^{k-1}}^{(k-1)} + \alpha^{(k-1)} f_i^{(k-1)} + f_{i+2^{k-1}}^{(k-1)}, \; i = 2^k, 2^{k+1}, \cdots, N - 2^k \end{array}$$

- **Résolution des systèmes**

  - **Initialisation : résolution du système le plus profond**

$$u_{2^{p-1}} = f_{2^{p-1}-2^{p-2}}^{(p-1)} + \alpha^{(p-1)} f_{2^{p-1}}^{(p-1)} + f_{2^{p-1}+2^{p-2}}^{(p-1)}$$

  - **résolution explicite des systèmes**
    Pour $k = p-1, \cdots, 1$

$$u_i = \frac{1}{\alpha^{(k-1)}} \left(f_i^{(k-1)} + u_{i-2^{k-1}} + u_{i+2^{k-1}} \right), \; i = 2^{k-1}, 3.2^{k-1}, 5.2^{k-1}, \cdots, N - 2^{k-1}.$$

1. Programmer la méthode pour $p = 1$.

Exercice 6 (Méthode de Bordage)
Soit A une matrice inversible de taille $(n-1) \times (n-1)$. On se donne

- b et c deux vecteurs colonne de $\mathbb{R}^{n-1}$
- a un scalaire

On considère alors la matrice M définie par

$$M = \begin{pmatrix} A & b \\ c^T & a \end{pmatrix}.$$

1. On considère le système linéaire

$$M\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}$$

avec x et f_1 dans $\mathbb{R}^{n-1}$, y et f_2 scalaires.

(a) Exprimer x et y en fonction de f_1 et de f_2. En déduire une condition nécéssaire et suffisante pour que M soit inversible.

(b) En déduire une méthode pour résoudre le système linéaire de proche en proche.

(c) Calcul de l'inverse de M. Exprimer l'inverse de M sous la forme

$$W = \begin{pmatrix} R & q \\ p^T & z \end{pmatrix}.$$

Ici R est une matrice $(n-1)\times(n-1)$, p et q deux vecteurs colonne de $\mathbb{R}^{n-1}$ et z un scalaire.

(d) En déduire une méthode pour construire une inverse de proche en proche.

(e) Application au calcul de l'inverse de A.

4 Annexe

4.1 Optimisation des calculs

La manipulation des matrices et des vecteurs est optimisée en Scilab. Exécuter le programme suivant :

```
// ----------------------------------------
// JPC - M2AAM -Calcul Scientifique 2019/2020
// ----------------------------------------
clear all
n=10; A=rand(n,n);
B=rand(n,n);
tic,C=A*B;
toc
tic
for i=1:n
for j=1:n
s=0;
for k=1:n
s=s+A(i,k)*B(k,j);
end
C(i,j)=s;
end
end
toc
```