Maîtrise de Mathématiques Fondamentales et Applications
Grands Systèmes linéaires
feuille de TP 4

1^er^ octobre 2009

Exercice - 1 *Méthode de résolution par la factorisation Cholesky*

1- Décomposition Cholesky et complexité.

a- Ecrire une fonction MATLAB de prototype `function [L,Nop] = decompCholesky(A)` qui effectue la décomposition Cholesky d'une matrice carrée symétrique définie positive et retourne le nombre `Nop` de multiplications et de divisions effectuées.

b- Générer aléatoirement des matrices symétriques définies positives (*voir* TP 1) de taille $n = 5, \dots 30$ et représenter en échelle logarithmique, la courbe $n \longmapsto$ Nop(n), où Nop(n) est le nombre d'opérations fournies par la décomposition Cholesky précédente.
En déduire la complexité de la factorisation Cholesky ; c'est-à-dire les constantes C et α, telles que $Nop(n) \approx Cn^\alpha$. Comparer cette complexité à celle obtenue par la factorisation LU (*voir* TP 3).

2- Inversion du système linéaire par la factorisation Cholesky.

a- Ecrire une fonction MATLAB de prototype `function [x] = inverseCholesky(L,b)` qui resoud le système $LL^Tx = b$. Tester cette fontion sur quelques exemples.

Exercice - 2 *Discrétisation par différences finies du Laplacien en une et deux dimensions*

1- Matrice du Laplacien en une dimension d'espace

On considère le problème
chercher $u \in c^2([0,1])$ telle que $-u^{''}(x) = f(x)$ dans$]0,1[$, $\quad u(0) = \alpha, u(1) = \beta$; où $f \in C([0,1])$ et α, β sont deux nombres réels donnés.

On se propose de déterminer une solution approchée de cette équation par la méthode de différences finies.

a- Rappeler les différentes étapes à suivre pour discrétiser cette équation par différences finies.
Donner l'expression de la matrice A et du second membre b de la formulation matricielle du problème discrétisé.

b- Déterminer analytiquement les valeurs propres de la matrice A. Calculer le rapport entre la plus grande valeur propre et la plus petite valeur propre. Que devient ce rapport lorsque la taille N de la matrice tend vers l'infini. Conclure.

c- Ecrire deux fonctions MATLAB de prototype `[A] = laplace1D(n)` et `[b] = Smlaplace1D(n,f,alpha,beta)` retournant respectivement la matrice et le second membre de la discrétisation. Et montrer qu'on peut résoudre le système $Au = b$ par la factorisation Cholesky.

d- Choisir f, α, β de sorte que la solution exacte soit $u_e(x) = sin(\pi x)$. Représenter sur un même graphique la solution exacte et la solution du système discrétisé.

e- Pour les mêmes expressions de f, α, β, ci-dessus, faites varier le pas de discrétisation $h = 1/(n+1)$ de façon dyadique ($h_i = h \times 2^{-i}, i = 1, 2, 3, 4$) et représenter à l'échelle logarithmique, la courbe $h_i \longmapsto \|u_e - u_{h_i}\|_\infty$, où u_{h_i} désigne la solution approchée obtenue par différences finies pour le pas h_i. Déterminer la pente de la droite (approchée) obtenue et déduire l'ordre de la discrétisation par différences finies adoptée.

2- Matrice du Laplacien en deux dimensions d'espace
On pose $\Omega =]0,1[\times]0,1[$, on se donne deux fonctions f, g continues sur $\bar{\Omega}$, et on cherche une fonction $u \in C^2(\Omega)$ telle que : $-\Delta u(x,y) = f(x,y)$ dans Ω, $\quad u(x,y) = g(x,y)$ sur $\partial\Omega$, où $\Delta u(x,y) = \frac{\partial^2 u(x,y)}{\partial^2 x} + \frac{\partial^2 u(x,y)}{\partial^2 y}$.

a- En suivant les étapes de discrétisation par différences finies vues en cours, donner l'expression de la matrice A et le second membre b de la formulation matricielle du problème discret.

b- Ecrire deux fonctions MATLAB de prototype `[A] = laplace2D(n,m)` et `[b] = Smlaplace2D(n,m,f,g)` retournant la matrice A et le second membre b obtenue ci-dessus, pour les pas de discrétisation $h_x = \frac{1}{n+1}, h_y = \frac{1}{m+1}$. Décrire la matrice obtenue.

Exercice - 3 *Stockage pour les méthodes directes d'inversion de systèmes linéaires*

Le but de cet exercice est d'exhiber un stockage de matrice favorable à l'utilisation des méthodes directes d'inversion des systèmes linéaires.
1- Matrices bandes (stokage et factorisation)
a- A l'aide de la commande `spy`, représenter graphiquement la matrice A de la discrétisation 1D du Laplacien. Représenter aussi les matrices de sa factorisation LU et Cholesky (*où on a stocké dans une matrice de même taille que A les factorisations LU et Cholesky*).
Comparer les largeurs de bande des différentes matrices et conclure.
.

b- Ecrire une fonction MATLAB de prototype `[AA, lbg, lbd, n ] = StockBand(A)` qui calcule les demi-largeurs de bande inférieure `lbg` et supérieure `lbd` d'une matrice pleine A et la stocke sous forme bande dans un vecteur AA de taille $(lbg + lbd + 1) \times n$.

Ecrire aussi une fonction MATLAB `[A] = Band2full(AA, lbg, lbd, n)` qui transforme une matrice stockée sous format bande en une matrice pleine.

c- Ecrire une fonction MATLAB de prototype `[AA, lbg, lbd, n , Nop] = decompLUBand(AA,lbg, lbd, n)`, qui effectue une factorisation LU d'une matrice stockée bande (*voir algorithme donné en annexe*).

Faites varier l'entier n et générer la matrice du laplacien 1D. Recupérer les nombres d'opérations donnés par les factorisations `decompLUBand` et `decompLU_nop` (*voir TP 3*). Représenter sur un même graphique à l'échelle logarithmique ces nombres d'opérations en fonction de la taille n. Calculer les pentes des droites obtenues et conclure.

d- Ecrire une fonction MATLAB de prototype `[x] = inverseLUBand(AA,lbg, lbd, n, b)`, qui resoud le système linéaire $Ax = b$ pour lequel la matrice A factorisée LU est stockée sous forme bande.

e- Ecrire une fonction MATLAB de prototype `[AA, lb, n ] = StockBandSym(A)` qui calcule la demi-largeur de bande `lb` d'une matrice pleine symétrique A et stocke sa partie triangulaire inférieure sous forme bande dans un vecteur AA de taille $(lb + 1) \times n$.

f- Ecrire une fonction MATLAB de prototype `[AA, lb, n] = decompCholBand(AA,lb, n)`, qui effectue une factorisation Cholesky d'une matrice symétrique stockée bande (*voir algorithme donné en annexe*)

g- Ecrire des fonctions MATLAB de prototype `[x] = inverseCholBand(AA,lb, n, b)`, qui resoud le système linéaire $Ax = b$ pour lequel la matrice A factorisée Cholesky est sctokée sous forme bande.

2- Réduction de largeur de bande des matrices
Cette question a pour but de montrer qu'on peut ramener, via une renumérotation, une matrice donnée avec beaucoup de zéros, en une matrice de faible lageur de bande.

Il est fourni, pour cette question, une fonction `[B,p]=rcm(A,i)` à travers un fichier `rcm.m`. Pour son appel, il faut apporter une matrice A pleine de taille n et un indice $1 \leq i \leq n$. On a en retour un vecteur p de permutation de l'ensemble $\{1, \ldots n\}$, et une matrice pleine $B_{ij} = A_{p(i)p(j)}, 1 \leq i, j \leq n$.
a- Tester cette fonction sur certaines matrices (`laplace2D(n,n)`) en affichant, commande `spy`, les matrices entrées et sorties. Commenter les observations.

Pour différentes valeurs de n, exécuter `A=laplace2D(n,n) ; [B,p]=rcm(A,1) ; AA=decompLU_eco(A) ; BB=decompLU_eco(B) ; length(find(AA)), length(find(BB))`. Conclure.
(*La fonction* `decompLU_eco` *est celle du* TP 3).

b- Générer pour différentes valeurs de n la matrice du laplacien en dimension deux. Pour un vecteur b donné de taille n, comparer les temps d'exécution des commandes :
- `A = laplace2D(n,n) ; x= A\b ;`
- `A = laplace2D(n,n) ; [B,p]=rcm(A,i) ; bb=b(p) ; y= B\bb ; x(p)=yy ;`

Conclure.