Examen de Modélisation et calcul scientifique

Jocelyne Erhel
INSA, Rennes

Février 2012

Durée 2 heures. Tous les documents sont autorisés. Les questions ne sont pas par ordre de difficulté et peuvent être traitées dans un ordre quelconque. Il est conseillé de lire tout l'énoncé avant de commencer.

On considère un problème de diffusion (par exemple un écoulement d'eau). Après discrétisation, on obtient un système linéaire d'ordre N, avec une matrice A creuse, symétrique définie positive. On veut effectuer une factorisation de Cholesky, $A = LL^T$. La matrice L est creuse, on définit $\text{Struct}(L_{*k}) = \{j \text{ tel que } j > k \text{ et } l_{jk} \neq 0\}$.

On rappelle ci-dessous l'algorithme de Cholesky, version fan-out.

Algorithme Cholesky fan-out, boucles kji
L=A
for $k = 1$ to N
  cdiv(k)
  for $j \in$Struct(L_{*k})
    cmod(j, k)
  endfor
endfor

L'opération cdiv(k) est la boucle
$l_{kk} = \sqrt{l_{kk}}$
for $i \in$ Struct(L_{*k})
  $l_{ik} = l_{ik}/l_{kk}$
endfor

L'opération cmod(j, k) est la boucle
for $i \in$ Struct(L_{*k}) et $i \geqslant j$
  $l_{ij} = l_{ij} - l_{ik} * l_{jk}$
endfor

1 Cholesky pour un problème 1D

On utilise un maillage régulier dans un domaine 1D et une méthode d'éléments finis mixtes. La matrice A a une structure tridiagonale.

Autrement dit, $a_{i,j} \neq 0 \Leftrightarrow |i-j| \leq 1$ (donc $i = j-1$ ou $i = j$ ou $i = j+1$).

1. Montrer par récurrence que la matrice L est bidiagonale ($l_{i,j} \neq 0 \Leftrightarrow i = j-1$ ou $i = j$). En déduire que $\text{Struct}(L_{*k}) = \{\min(k+1, N)\}$.

2. Ecrire la version fan-out de l'algorithme, en détaillant les fonctions cdiv(k) et cmod(j,k). Préciser les indices des boucles sur j et i, autrement dit les éléments dans $\text{Struct}(L_{*k})$.

3. En déduire que le nombre d'opérations de la factorisation est environ $4N$.

2 Cholesky pour un problème 2D

On utilise un maillage régulier dans un domaine 2D carré, avec n mailles dans chaque direction, soit un total de $N = n^2$ mailles. La matrice A a une structure régulière, avec les éléments non nuls répartis sur cinq diagonales: $i - j = -n, -1, 0, 1, +n$.

On n'effectue pas de renumérotation. On suppose que la matrice L a une structure bande: les éléments vérifient $l_{i,j} \neq 0 \Rightarrow |i-j| \leq n$. On suppose que $|i-j| \leq n \Rightarrow l_{i,j} \neq 0$. Donc $\text{Struct}(L_{*k}) = \{j \text{ tel que } k < j \leq \min(k+n, N)\}$.

1. Vérifier que le nombre d'éléments non nuls de A est environ $nz(A) = 5N$.

2. Vérifier que le nombre d'éléments non nuls de L est environ $N^{3/2}$.

3. Ecrire la version fan-out de l'algorithme de Cholesky, en détaillant les fonctions cdiv(k) et cmod(j,k). Préciser les indices de boucles j et i, autrement dit les éléments dans $\text{Struct}(L_{*k})$.

4. Calculer le nombre d'opérations approximatif de cdiv(k).

5. Calculer le nombre d'opérations approximatif de cmod(j,k).

6. En déduire que le nombre d'opérations de la factorisation est environ N^2.

7. Analyser le remplissage lors de la factorisation et retrouver le résultat que la structure de L est bande.

8. Quels algorithmes de renumérotation pourrait-on utiliser pour réduire le remplissage ?

3 Gradient Conjugué Préconditionné

On rappelle ci-dessous l'algorithme du Gradient Conjugué Préconditionné, PCG, pour résoudre le système $Ax = b$ (avec x_0 donné), où A est une matrice creuse, d'ordre N, symétrique définie positive:

Algorithme PCG
$r_0 = b - Ax_0; z_0 = M^{-1}r_0; p_0 = z_0; \gamma_0 = r_0^T z_0$
for $k = 0$ to $niter$ until convergence
  $q_k = Ap_k$
  $\delta_k = p_k^T q_k$
  $\alpha_k = \frac{\gamma_k}{\delta_k}$
  $x_{k+1} = x_k + \alpha_k p_k$
  $r_{k+1} = r_k - \alpha_k q_k$
  $z_{k+1} = M^{-1} r_{k+1}$
  $\gamma_{k+1} = r_{k+1}^T z_{k+1}$
  $\beta_k = \frac{\gamma_{k+1}}{\gamma_k}$
  $p_{k+1} = z_{k+1} + \beta_k p_k$
endfor

On choisit un préconditionnement par factorisation de Cholesky incomplète. On a $M = CC^T$, où C est une matrice triangulaire inférieure inversible, de même structure creuse que A, autrement dit telle que Struct(C_{*k}) =Struct(A_{*k}).

Pour calculer $z = M^{-1}r$, on calcule $u = C^{-1}r$ puis $z = C^{-T}u$, où C^{-T} désigne l'inverse de C^T.

On utilise l'algorithme suivant pour calculer u:

Algorithme de résolution d'un système triangulaire $Cu = r$
$u = r$
for $j = 1$ to N
  $u_j = \frac{1}{c_{jj}} u_j$
  for $i \in$ Struct(C_{*j})
    $u_i = u_i - c_{ij} * u_j$
  endfor
endfor

On note NZ le nombre d'éléments non nuls dans la partie strictement triangulaire inférieure de A.

1. Montrer que le nombre d'éléments non nuls dans A est $nz(A) = 2NZ + N$.
2. Montrer que le nombre d'éléments non nuls dans C est $nz(C) = NZ + N$.
3. Montrer que le nombre d'opérations du produit $q = Ap$ est $2nz(A)$.
4. Montrer que le nombre d'opérations de la résolution $u = C^{-1}r$ est $nz(A)$.
5. Ecrire l'algorithme de la résolution $z = C^{-T}r$.

6. Montrer que le nombre d'opérations de la résolution $z = C^{-T}r$ est $nz(A)$.
7. Montrer que le nombre d'opérations pour les calculs vectoriels (produits scalaires δ_k et γ_k, calculs de x_k, r_k, p_k) est $10N$.
8. Calculer le nombre d'opérations dans une itération du gradient conjugué préconditionné, noté $nop(pcg)$.
9. Calculer le nombre d'opérations dans une itération du gradient conjugué non préconditionné, noté $nop(cg)$.
10. On considère la matrice du problème 2D précédent. Calculer $nop(pcg)$ et $nop(cg)$ en fonction de N.
11. Sans préconditionnement, le gradient conjugué, appliqué à cette matrice, converge en 100 itérations. Quel est le nombre maximum d'itérations dans le gradient conjugué préconditionné pour qu'il nécessite moins d'opérations au total ?