Un peu d'analyse numérique matricielle

mots clefs Scilab: *diag, chfact, chsolve, chol, cond, eye, inv, linsolve, lu, lufact, luget, lusolve, norm, plot, plot2d, qr, schur, sparse, spec, sqroot, ...*

1 Quelques systèmes linéaires associés au laplacien

$$-u''(x) + c(x)u(x) = f(x), \qquad 0 < x < 1, \qquad u(0) = u(1) = 0, \tag{1}$$

où $c(.)$ est une fonction positive donnée : $c(.) \geq 0$.
On pose $h = \frac{1}{N+1}$, $x_j = jh$, $f_j = f(x_j)$, $c_j = c(x_j)$, $u_j \simeq u(x_j)$, $j = 0, \cdots, N+1$, u_h le vecteur ayant pour composante $(u_j)_{1 \leq j \leq N}$, f_h le vecteur ayant pour composante $(f_j)_{1 \leq j \leq N}$. On cherche à résoudre $Au_h = h^2 f_h$:

$$\begin{pmatrix} 2+h^2c_1 & -1 & 0 & \cdots & 0 \\ -1 & 2+h^2c_2 & -1 & 0 & 0 \\ 0 & -1 & 2+h^2c_3 & -1 & 0 \\ \vdots & & & & \vdots \\ 0 & 0 & -1 & 2+h^2c_{N-1} & -1 \\ 0 & \cdots & 0 & -1 & 2+h^2c_N \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ \vdots \\ u_{N-1} \\ u_N \end{pmatrix} = h^2 \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ \vdots \\ f_{N-1} \\ f_N \end{pmatrix} \tag{2}$$

On va utilser plusieurs méthodes pour résoudre (2).
On commencera par traiter les cas suivants : $c \equiv 0$ ou 1, et, $f \equiv 0$ ou 1. Comparer les résultats obtenus avec la solution exacte.

1. Résoudre le système linéaire (2) à l'aide d'une méthode directe : Choleski , Gauss.

2. Résoudre le système linéaire (2) à l'aide d'une méthode itérative,
   de la forme $Bu^{k+1} = Cu^k + b$ où $b = h^2 f_h$, $A = B - C$, $B \in GL_N(I\!R)$, u^0 est à choisir,
   ($u^0 = 0$ ou b par exemple), et $\lim_{k \to \infty} u^k = u_h$.
   Pour définir B et C on posera $A = D - E - F$ où :
   $D_{ij} = A_{ij}\delta_{ij}$, "diagonale de A"
   $(-E)_{ij} = A_{ij}$ si $i > j$, 0 sinon, " partie triangulaire inférieure stricte de A"
   $(-F)_{ij} = A_{ij}$ si $i < j$, 0 sinon, " partie triangulaire supérieure stricte de A".
   On prendra soin à minimiser le nombre de calculs. Deux écriture mathématiques équivalentes peuvent correspondre à deux coûts de caculs trés différents !

   (a) Utiliser la méthode de Jacobi : $B = D$, $C = E + F$.

   (b) Utiliser la méthode de Gauss-Seidel : $B = D - E$, $C = F$.

   (c) Utiliser la méthode de relaxation de paramètre $\omega \neq 0$: $B = \frac{1}{\omega}D - E$, $C = \frac{1-\omega}{\omega}D + F$.

3. Appliquer la méthode du gradient conjugué. (Ciarlet p. 199)
   On cherche à minimiser $J(v) = \dfrac{1}{2}\langle Av, v\rangle - \langle b, v\rangle$ en effet, $Au_h = b \iff \nabla J(u_h) = 0$.
   $u^{k+1} = u^k - r_k d^k, d^k = \nabla J(u^k) + \dfrac{\|\nabla J(u^k)\|^2}{\|\nabla J(u^{k-1})\|^2} d^{k-1}, \quad d^{-1} = 0, r_k = \dfrac{\langle \nabla J(u^k), d^k\rangle}{\langle Ad^k, d^k\rangle}.$

2 Valeurs propres

On cherche les valeurs de λ telles que le problème suivant :

$$-u''(x) + c(x)u(x) = \lambda u(x), \qquad 0 < x < 1, \qquad u(0) = u(1) = 0, \tag{3}$$

admette une solution u non identiquement nulle. Après discrétisation de ce problème, on se ramène à chercher31 les valeurs propres de $\frac{1}{h^2}A$. Puisque A est une matrice symétrique définie positive, il existe $O \in O_n(I\!R)$ et Λ une matrice diagonale telles que $O^tAO = \Lambda$. Trouver les valeurs propres de A revient à trouver Λ. Trouver les vecteurs propres de A revient à trouver O.

1. La méthode de la puissance itérée : Soit $u^0 \in I\!R^N$, $l \in (I\!R^N)^*$, on pose $u^{k+1/2} = Au^k$, $u^{k+1} = \frac{1}{l(u^{k+1/2})}u^{k+1/2}$. Pour presque tout x_0 et l, $(u^k)_{k\geq 0}$ converge vers un vecteur propre de A, u^∞, associé à sa plus grande valeur propre, λ_∞ (son rayon spectral).

   (a) La méthode de la puisance modifiée : $u^{k+1/2} = Au^k$, $u^{k+1} = \frac{1}{\|u^{k+1/2}\|}u^{k+1/2}$.

   (b) En projetant (à chaque étape !) sur l'orthogonal de u^∞, obtenir numériquement la deuxième plus grande valeur propre de A.

   (c) Obtenir ainsi, toutes les valeurs propres de A et sa base de vecteurs propres.

2. Par la factorisation de Choleski itérée: $A_1 = A$, $A_k = C_k^tC_k$, $A_{k+1} = C_kC_k^t$. Vérifier que $A_k \to \Lambda$.
   Faire de même avec la factorisation QR .

3. La méthode de Jacobi. (Ciarlet p.113)
   Soit $(e_j)_{1\leq j\leq N}$ la base canonique de $I\!R^N$, $1 \leq p < q \leq N$, $\theta \in I\!R$, on note $\Omega_{pq}(\theta)$, ou, plus simplement Ω la matrice de la rotation d'angle θ dans le plan (e_p, e_q) : $\Omega_{pp} = \Omega_{qq} = \cos\theta$, $\Omega_{pq} = -\Omega_{qp} = \sin\theta$, $\Omega_{ii} = 1$, $p \neq i \neq q$, $\Omega_{ij} = 0$, $i \neq j$, $p \neq i \neq q$,$p \neq j \neq q$,
   Cette méthode itérative consiste à "éliminer" (temporairement) le terme $a^k_{pq} \neq 0$ de A. Soit $\theta \in]-\pi/4, \pi/4]$ donné par $cotg(2\theta) = \dfrac{a^k_{qq} - a^k_{pp}}{2a^k_{pq}}$, on pose $\Omega_k = \Omega_{pq}(\theta)$, $A_{k+1} = \Omega_k^tA_k\Omega_k$, $A_1 = A$, $P_{k+1} = P_k\Omega_k$, $P_1 = Id$. On a $A_k \to \Lambda$ et $P_k \to O$.
   Classiquement, on cherche à "éliminer" le terme le plus gros à chaque étape : on choisit (p, q) tel que $|a^k_{pq}| \max\limits_{i\neq j} |a^k_{ij}|$.

   Cycliquement, on balaye tout les termes non nuls (surdiagonaux) de la matrice A. (On peut aussi omettre "d'éliminer" le terme $|a^k_{pq}|$ inférieur à un seuil donné.)

Références :

- Ciarlet, Introduction à l'analyse numérique matricielle et à l'optimisation,
- Lascaux & Théodor: Analyse numérique matricielle appliquée à l'art de l'ingénieur.
- Schatzman, analyse numérique : cours et exercices pour la licence,