Analyse Numérique Licence L3 de Mathématique 2015-2016, TP4

Soit $A \in \mathcal{GL}_n(\mathbb{R})$ une matrice inversible telle que $A_{i,i} \neq 0$ pour tous $i = 1, \cdots, n$. On considère $b \in \mathbb{R}^n$ et le système linéaire $Ax = b$ de solution $x \in \mathbb{R}^n$.

$$\text{On pose } e^{(k)} = x - x^{(k)},\ r^{(k)} = Ae^{(k)} = b - Ax^{(k)},$$

Soit $C \in \mathcal{GL}_n(\mathbb{R})$ un préconditionnement. On considère l'algorithme de Richardon de pas 1 préconditionné par C pour résoudre le système $Ax = b$: $x^{(1)}$ donné, $r^{(1)} = b - Ax^{(1)}$ et pour $k \in \mathbb{N}$

$$\left\{\begin{aligned} q^{(k)} &= Ap^{(k)}, \\ p^{(k)} &= C^{-1}r^{(k)}, \\ x^{(k+1)} &= x^{(k)} + p^{(k)}, \\ r^{(k+1)} &= r^{(k)} - q^{(k)}. \end{aligned}\right.$$

On considère dans ce qui suit les préconditionnement de Jacobi

$$C = D,$$

le préconditionnement de Gauss Seidel

$$C = D - E$$

et le préconditionnement SOR (Successive Over Relaxation) :

$$C = (\frac{D}{\omega} - E),$$

avec $\omega \in]0, 2[$ et où D est la partie diagonale de A, $-E$ est la partie triangulaire inférieure stricte de A.

(1) Soient $n_x \in \mathbb{N}^*$, $n_y \in \mathbb{N}^*$, $n = n_x n_y$, $\Delta x = \frac{1}{n_x+1}$, $\Delta y = \frac{1}{n_y+1}$. Programmer dans scilab la matrice pentadiagonale $A \in \mathcal{M}_n(\mathbb{R})$ telle que $A_{i,i} = \frac{2}{\Delta x^2} + \frac{2}{\Delta y^2}$ pour $i = 1, \cdots, n$, $A_{i,i+1} = -\frac{1}{\Delta x^2}$ pour $i = 1, \cdots, n-1$, $A_{i,i-1} = -\frac{1}{\Delta x^2}$ pour $i = 2, \cdots, n$, $A_{i,i+n_x} = -\frac{1}{\Delta y^2}$ pour $i = 1, \cdots, n-n_x$, $A_{i,i-n_x} = -\frac{1}{\Delta y^2}$ pour $i = n_x + 1, \cdots, n$, et $A_{i,j} = 0$ pour tous les autres indices.
On utilisera la commande $diag(u)$ et $diag(u_x, +-1)$, $diag(u_y, +-n_x)$ où u et u_x, u_y sont des vecteurs.

(2) Programmer dans scilab le vecteur colonne $y \in \mathbb{R}^n$ tel que $y(i_x + (i_y - 1)n_x) = \sin(100 i_x \Delta x) + \sin(10 i_y \Delta y)$, $i_x = 1, \cdots, n_x$, $i_y = 1, \cdots, n_y$ et calculer $b = Ay$. Par la suite on va résoudre le système $Ax = b$ de solution y par l'algorithme du gradient conjugué.

(3) Programmer les trois fonctions scilab qui prennent en entrée n, $A \in \mathcal{M}_n(\mathbb{R})$, $r \in \mathbb{R}^n$ et qui calculent la solution $p \in \mathbb{R}^n$ du système

$$Cp = r,$$

dans les trois cas $C = D, D - E, \frac{D}{\omega} - E$. On s'attachera à utiliser autant que possible des produits sous matrice sous vecteur au lieu de boucles for.

(4) Programmer dans scilab l'algorithme de Richardson préconditionné ci-dessus La forme programmée dans scilab ne stokera que l'itéré courant de p, q, x, r. On utilisera la boucle *while condition* ; *end*, le produit scalaire $(x, y) = x' * y$. Un seul produit matrice vecteur et un seul appel au préconditionnement sont autorisés par itération.

(5) Tests numériques :

Tracer la courbe du log de la norme du résidu relatif $log(\frac{nr^{(k)}}{nr^0})$ fonction de k. Tester différentes valeur du paramètre de relaxation ω. Trouver numériquement la valeur optimale de ω. Comparer les convergences obtenues avec les trois algorithmes pour différentes valeurs de $n_x = n_y$.