Analyse Numérique Licence L3 de Mathématique 2015-2016, TP5

Soit $A \in \mathcal{M}_n(\mathbb{R})$ une matrice SDP (Symétrique Définie Positive). 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 $C \in \mathcal{M}_n(\mathbb{R})$ un préconditionnement SDP. On considère l'algorithme itératif du gradient conjugué préconditionné par C pour résoudre le système $Ax = b$: $x^{(1)}$ donné, $r^{(1)} = b - Ax^{(1)}$, $p^{(1)} = C^{-1}r^{(1)}$ et pour $k \in \mathbb{N}$

$$\left\{\begin{array}{l} x^{(k+1)} = x^{(k)} + \alpha^{(k)}\ p^{(k)}, \\ r^{(k+1)} = r^{(k)} - \alpha^{(k)}\ Ap^{(k)}, \\ p^{(k+1)} = C^{-1}r^{(k+1)} + \beta^{(k)}\ p^{(k)}, \\ \alpha^{(k)} = \frac{(C^{-1}r^{(k)}, r^{(k)})}{(p^{(k)}, Ap^{(k)})}, \\ \beta^{(k)} = \frac{(C^{-1}r^{(k+1)}, r^{(k+1)})}{(C^{-1}r^{(k)}, r^{(k)})}. \end{array}\right.$$

On considère dans ce qui suit le préconditionnement SSOR (Symmetric Successive Over Relaxation) :

$$C = (\frac{D}{\omega} - E)D^{-1}(\frac{D}{\omega} - F)$$

avec $\omega \in [1, 2[$ et où D est la partie diagonale de A, $-E$ est la partie triangulaire inférieure stricte de A et $-F$ est la partie triangulaire supérieure stricte de A.

(1) Programmer la fonction scilab $z = SSOR(n, A, w, r)$ qui calcule la solution $z \in \mathbb{R}^n$ du système

$$(\frac{D}{\omega} - E)D^{-1}(\frac{D}{\omega} - F)z = r.$$

On résoudra ce système en faisant une descente

$$(\frac{D}{\omega} - E)z_1 = r,$$

suivie d'une remontée

$$(\frac{D}{\omega} - F)z = Dz_1.$$

(2) Programmer dans scilab l'algorithme du gradient conjugué préconditionné suivant en utilisant la fonction $SSOR$ précédente :

— Choix de la précision $\epsilon = 10^{-10}$ sur le résidu relatif

— Initialisation : $x^{(1)} = 0$ (dans $\mathbb{R}^n$), $r^{(1)} = b - Ax^{(1)}$, $p^{(1)} = C^{-1}r^{(1)}$, $z^{(1)} = p^{(1)}$ $nr^{(1)} = \|r^{(1)}\|_2$, $k = 1$

— Tant Que $\frac{nr^{(k)}}{nr^{(1)}} > \epsilon$ Faire

$q^{(k)} = Ap^{(k)}$
$\alpha^{(k)} = \frac{(z^{(k)}, r^{(k)})}{(p^{(k)}, q^{(k)})}$
$x^{(k+1)} = x^{(k)} + \alpha^{(k)} r^{(k)}$
$r^{(k+1)} = r^{(k)} - \alpha^{(k)} q^{(k)}$

$z^{(k+1)} = C^{-1} r^{(k+1)}$
$\beta^{(k)} = \frac{(z^{(k+1)}, r^{(k+1)})}{(z^{(k)}, r^{(k)})}$
$p^{(k+1)} = z^{(k+1)} + \beta^{(k)} p^{(k)}$
$nr^{(k+1)} = \|r^{(k+1)}\|_2$
$k = k + 1$

— End Tant Que
— $nit = k$

La forme programmée dans scilab ne stokera que l'itéré courant de p, q, z, 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.

(3) Tests numériques :
Tracer la courbe de la norme du résidu $nr^{(k)}$ fonction de k. Tester différentes valeur du paramètre de relaxation ω. Comparer la convergence obtenue avec celle de l'algorithme sans préconditionnement ie celui obtenue avec $z = r$.