EISTI - DÉPARTEMENT MATHÉMATIQUES

ANALYSE NUMÉRIQUE - TP N° 2

7 avril 2008

ÉTUDE DE MÉTHODES DE RÉSOLUTION DE SYSTÈMES LINÉAIRES

Considérons le système linéaire

$$\mathbf{Ax} = \mathbf{b}; \quad \mathbf{A} \in \mathbf{R}^{n\times n}, \quad \mathbf{x}, \mathbf{b} \in \mathbf{R}^n$$

avec $\mathbf{A} = [a_{ij}]$ matrice telle que

$$a_{ij} = \begin{cases} 2, & \text{si } i = j \\ -1, & \text{si } j = i+1 \text{ ou } i = j+1 \\ 0, & \text{sinon} \end{cases}$$

et $\mathbf{b} = [1, 0, \ldots, 0, 1]^\top$.

Nous allons résoudre ce système par trois méthodes : Décomposition LU, Jacobi et Jacobi relaxée.

TRAVAIL À FAIRE

(1) Décomposition LU : Il faut utiliser le programme Scilab `lu` (pour les détails de ce programme, voir l'aide de Scilab) pour obtenir la décomposition de la matrice $\mathbf{A}$ en deux matrices triangulaires $\mathbf{L}$ (triangulaire inférieure) et $\mathbf{U}$ (triangulaire supérieure). Ensuite il faut résoudre les deux systèmes triangulaires $\mathbf{Ly} = \mathbf{b}$ et $\mathbf{Ux} = \mathbf{y}$ pour calculer $\mathbf{x}$.

   (a) On fera une étude de la précision numérique du résultat, en utilisant comme critère la norme $\|\mathbf{Ax} - \mathbf{b}\|_\infty$, en fonction de la taille de la matrice $\mathbf{A}$. On prendra $n = 10, 20, 40, 80$.

   (b) On évaluera le nombre d'opérations nécessaires pour obtenir le résultat dans chaque cas.

   (c) En utilisant la fonction Scilab `timer` (pour les détails de cette fonction, voir l'aide de Scilab), évaluer le temps CPU pour faire seulement les calculs de la décomposition LU et les calculs de la résolution de deux systèmes triangulaires.

(2) Méthode itérative de Jacobi : Il faut programmer la méthode de Jacobi (pour les formules se reporter au poly du cours). On peut prendre $\mathbf{x}(0) = [0, \ldots, 0]$.

   (a) On fera une étude de la précision numérique du résultat, en utilisant comme critère la norme $\|\mathbf{Ax} - \mathbf{b}\|_\infty$, en fonction de la taille de la matrice $\mathbf{A}$. On prendra $n = 10, 20, 40, 80$.

   (b) On évaluera le nombre d'itérations nécessaires pour obtenir le résultat dans chaque cas.

(c) En utilisant la fonction Scilab `timer`, évaluer le temps CPU nécessaire pour l'application de la méthode de Jacobi dans chaque cas.

(3) Méthode itérative relaxée de Jacobi. Il faut programmer la méthode relaxée de Jacobi (pour les formules se reporter au poly du cours). On peut prendre $\mathbf{x}(0) = [0,\dots,0]$ et on choisira le coefficient de relaxation μ dans l'intervalle $]0,2]$.

(a) En utilisant plusieurs valeurs pour le coefficient de relaxation μ, trouver la valeur optimale de μ.

(b) On fera une étude de la précision numérique du résultat, en utilisant comme critère la norme $\|\mathbf{Ax}-\mathbf{b}\|_\infty$, en fonction de la taille de la matrice $\mathbf{A}$ et en utilisant la valeur optimale de μ trouvée précédement. On prendra $n = 10,20,40,80$.

(c) On évaluera le nombre d'itérations nécessaires pour obtenir le résultat dans chaque cas de la question précedente.

(d) En utilisant la fonction Scilab `timer`, évaluer le temps CPU nécessaire pour l'application de la méthode de Jacobi dans chacun de cas précédents

(4) Évaluer le nombre d'itérations nécessaires pour avoir une précision numérique de 10^{-12} avec la méthode de Jacobi et la méthode de Jacobi relaxée.

(5) Comparer les trois méthodes ci-dessus en termes

(a) de précision des résultats ;

(b) d'occupation mémoire, et

(c) de temps de calcul.

Rapport et programmes à rendre jusqu'au 18 avril, minuit, en utilisant le bouton approprié du TP 2 sur le site du cours **`http ://sifoci.eisti.fr`** .