

Examen sur machine du 8 janvier 2008
(9h30-12h30)

Les notes de cours et de TD/TP sont autorisées.

Il est demandé de tester vos fonctions sur des exemples simples pour vérification.

En cas d'erreur(s), lire attentivement les messages donnés par Scilab.

Appeler les surveillants pour tout problème non mathématique.

Recommandations : Il faut taper toutes les commandes, fonctions, et autres dans un seul fichier que vous nommerez exam-nom.sce, où "nom" est à remplacer par votre nom de famille (ou votre login). On séparera les réponses aux différentes questions par de larges espaces avec si possible, un commentaire indiquant le numéro de la question. Dans les questions qui demandent une courte réponse, vous la taperez en commentaire dans le fichier. La syntaxe des commentaires est la suivante :

```
\\ On place les commentaires après un \\, Scilab ne lira pas ce qui suit,  
\\ jusqu'à la fin de la ligne.  
\\ Pour changer de ligne, on commence une nouvelle ligne avec un double anti-slash \\  

```

```
\\ Question 4  
.... mes instructions Scilab pour la réponse.
```

```
\\ question 5
```

Une matrice réelle (ou complexe) carrée de taille n est dite à diagonale strictement dominante si et seulement si :

$$\forall i \leq n, \quad |a_{i,i}| > \sum_{j \neq i} |a_{i,j}|$$

1) Définir une fonction `somme_ligne`, qui quand on lui donne une matrice A et un indice de ligne i , calcule et renvoie la somme $\sum_{j \neq i} |a_{i,j}|$.

Vérifier cette fonction avec les matrices

$$A = \begin{pmatrix} -1 & 1 & 1 \\ 2 & -2 & 1 \\ 3 & 1 & -3 \end{pmatrix}, \quad B = \begin{pmatrix} -7 & 3 & 1 \\ 2 & 5 & 1 \\ 4 & -1 & -8 \end{pmatrix}$$

2) Définir une fonction `ligne_OK`, qui quand on lui donne une matrice A et un indice de ligne i , renvoie 1 si $|a_{i,i}| > \sum_{j \neq i} |a_{i,j}|$, et 0 sinon.

Vérifier votre fonction avec les matrices A et B .

3) Définir une fonction `diag_strict_dom`, qui quand on lui donne une matrice A , renvoie 1 si la matrice est à diagonale strictement dominante, et 0 sinon.

Vérifier votre fonction avec les matrices A et B .

On sait que l'algorithme de Gauss-Seidel pour résoudre le système $Ax = b$ converge si la matrice A est à diagonale strictement dominante.

4) Définir une fonction `Gauss_Seidel`, qui quand on lui donne une matrice A , un vecteur b , une précision e et un nombre d'itération maximum `max_iter`, calcule une solution du système $Ax = b$ par la méthode de Gauss-Seidel. Cette fonction devra d'abord vérifier si la matrice A est à diagonale strictement dominante et si c'est le cas, calculer une solution approchée à e près, en faisant au plus `maxiter` itérations. Elle devra retourner la solution x et le nombre d'itérations effectuées `nbiter`.

Vérifier votre fonction avec les matrices A et B , et le vecteur $b = (1, 1, 1)'$ (n'oublier pas la transposée).

On définit la matrice de Toeplitz (c-à-d constante sur toutes les diagonales) 100×100 suivante :

$$C = \begin{pmatrix} 10000 & 99 & 98 & \dots & 2 & 1 \\ 99 & 10000 & 99 & 98 & \dots & 2 \\ 98 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 98 \\ 2 & \dots & 98 & 99 & 10000 & 99 \\ 1 & 2 & \dots & 98 & 99 & 10000 \end{pmatrix}$$

5) a) Définir la matrice C sous Scilab et vérifier qu'elle est à diagonale strictement dominante.

b) Calculer avec la fonction `Gauss_Seidel`, une solution du système $Cx = b$, avec $b = (10000, 10000, \dots, 10000)$. Noter le nombre d'itération nécessaire.

6) En analyse numérique matricielle, on s'intéresse souvent au nombre d'opérations élémentaires (additions, multiplications,...) effectuées, et plus particulièrement à l'ordre du nombre d'opérations en fonctions de la taille n de la matrice. C'est-à-dire qu'on annonce souvent un nombre d'opération de l'ordre de n, n^2, n^3, n^k , plutôt qu'une valeur exacte difficile à calculer (Exemple : Pour $5n$ opérations, on dira que le nombre opération est d'ordre n).

a) Quel est l'ordre du nombre d'opérations nécessaires pour inverser une matrice triangulaire inférieure ?

b) Quel est l'ordre du nombre d'opérations effectuées dans une itération de la boucle à l'intérieur de la fonction `Gauss_Seidel` ?

c) Donner un ordre du nombre d'itérations effectuées par la fonction `Gauss_Seidel` en fonction de la taille de la matrice n et du nombre d'itération (on admettra que Scilab calcule les inverses de matrice triangulaire en un nombre minimal d'opérations).

7) a) Utiliser les fonctions déjà programmées dans Scilab pour calculer une factorisation LU de C (n'oublier pas de d'utiliser l'aide pour les problèmes de syntaxe).

b) Donner l'ordre du nombre d'opérations effectuées lors d'une factorisation LU.

c) Quelle méthode conseilleriez-vous pour la résolution du système $Cx = b$?

d) Avez-vous observé si une opération, entre la factorisation LU et l'utilisation de la fonction `Gauss-Seidel`, a demandé plus de temps à Scilab ? Dans tous les cas, donnez une explication rapide.