CMI, Université de Provence **Licence 3ème année**
Année 2010-2011

TP2 : Méthodes directes pour les systèmes linéaires

1 Méthode de Gauss

Dans cette rubrique on programme le pivot de Gauss dans un premier temps sans faire de recherche de pivot, puis avec la méthode du pivot partiel.

1.1 Méthode du pivot simple

Vous pouvez

- *Soit vous lancer directement dans la programmation du pivot de Gauss avec l'algorithme étudié en cours et la tester pour résoudre les systèmes suivants.*

$$\begin{cases} 6x_1 + x_2 + x_3 + x_4 = \sqrt{2} \\ x_1 - x_2 + 2x_3 + 3x_4 = 5.1 \\ -x_1 + 2x_2 + x_3 - x_4 = 3 \\ x_1 + 3x_2 - x_3 + x_4 = 1 \end{cases} \qquad \begin{cases} x_1 + 2x_2 + 4x_3 + 8x_4 + 16x_5 = 31 \\ 2x_1 + 4x_2 + 8x_3 + 16x_4 + 31x_5 = 61 \\ 4x_1 + 8x_2 + 16x_3 + 31x_4 + 61x_5 = 120 \\ 8x_1 + 16x_2 + 31x_3 + 61x_4 + 120x_5 = 236 \\ 16x_1 + 31x_2 + 61x_3 + 120x_4 + 236x_5 = 464 \end{cases}$$

  Tester ensuite ce que vous donne la commande \ de Scilab. Que s'est-il passé ?

- *Soit suivre les instructions suivantes qui vous proposent de détailler un certain nombre de fonctions intermédiaires avant d'obtenir le programme final à tester sur les mêmes matrices.*

A vous de choisir !!

1. Soit A une matrice carrée et b un vecteur colonne ayant le même nombre de lignes que A. A quoi correspond `[A b]` dans Scilab ?

2. Fabriquer une fonction *operationlin* qui fait la somme d'une ligne donnée avec une autre multipliée par une constante donnée.

3. Fabriquer une fonction *resoltriang* qui résout un système triangulaire $Tx = c$ où T est une matrice triangulaire supérieure et c un vecteur qui a le même nombre de lignes que T sans utiliser la commande \ de Scilab.

4. En déduire le programme qui vous permet de résoudre un système linéaire à l'aide de la méthode de Gauss et l'appliquer aux deux systèmes proposés dans l'introduction.

1.2 Méthode du pivot partiel

De même que précédemment, vous pouvez

- *Soit programmer la méthode du pivot partiel directement (on recherche dans la colonne sur laquelle on travaille l'élément maximal non nul en dessous de la diagonale et on permute la ligne correspondante pour la placer comme ligne de pivot).*

- *Soit fabriquer les programmes intermédiaires qui sont détaillés ci-dessous.*

1. Fabriquer une fonction *nonzero* qui permet de rechercher l'élément de valeur absolue maximale et non nul sous la diagonale d'une colonne donnée et qui renvoie la ligne à laquelle il appartient.

2. Fabriquer une fonction *permute* qui permet d'échanger deux lignes dans une matrice donnée.

3. En déduire le programme pour le pivot partiel.

2 Systèmes linéaires et factorisation LU

La commande `lu` de Scilab permet d'effectuer la factorisation LU d'une matrice.

1. Programmer une fonction qui permet de résoudre le système linéaire $Ax = b$ à l'aide de la factorisation LU d'une matrice.

2. L'appliquer pour résoudre le système

$$\begin{cases} 3x_1 - 7x_2 - 2x_3 + 2x_4 = -9 \\ -3x_1 + 5x_2 + x_3 = 5 \\ 6x_1 - 4x_2 - 5x_4 = 7 \\ -9x_1 + 5x_2 - 5x_3 + 12x_4 = 11 \end{cases}$$

3. Comment pourriez-vous adapter votre programme sur le pivot de Gauss pour obtenir aussi la décomposition LU d'une matrice A ?

3 Factorisation de Cholesky

1. Programmer l'algorithme écrit lors de la résolution de l'exercice 9 du Td 1 pour la factorisation de Cholesky d'une matrice tridiagonale en ajoutant un compteur qui vous permet à la fin d'obtenir le nombre d'opérations que vous avez effectué lors de la factorisation.

2. Pouvez vous résoudre des systèmes linéaires en utilisant la factorisation de Cholesky d'une matrice symétrique définie positive ?

3. Le tester sur des matrices tridiagonales $A = (a_{i,j})$ telle que $a_{i,i} = 2$ et $a_{i,i+1} = a_{i,i-1} = -1$ de taille $n \times n$. Tracer la courbe du nombre d'opérations en fonction de n.