Université Ferhat ABBAS – Sétif 1
Faculté des Sciences
Département d'Informatique

2ième année Licence
Matière: Méthode numérique / Travaux Pratiques

Année 2020/2021

TP N°2 Résolution numérique des systèmes linéaires $Ax = b$

méthodes directes.

Objectif: Le but de ce TP est de résoudre le système $Ax = b$ par des méthodes numériques directes .

On considère le système d'équations linéaires $\boldsymbol{Ax = b}$; Où $\boldsymbol{A} \in \boldsymbol{\mathcal{M}_n}(\mathbb{R})$, $\boldsymbol{b} \in \mathbb{R}^{\boldsymbol{n}}$ et $\boldsymbol{x} \in \mathbb{R}^{\boldsymbol{n}}$.
Le système $\boldsymbol{Ax = b}$ admet une solution unique si et seulement si $det(A) \neq 0$

1. Cas particuliers:

1.1 Système triangulaire supérieure : est un système linéaire $Ux = b$, où U est une matrice inversible triangulaire supérieure d'ordre n.

Algorithme de résolution:

$$\begin{cases} x_{n=\frac{b_n}{U_{nn}}} \\ pour\ i = n-1,1 \\ x_i = \dfrac{1}{U_{ii}}\left[b_i - \displaystyle\sum_{j=i+1}^{n} U_{ij}x_j\right] \end{cases} \qquad (1.1)$$

1.2 Système triangulaire inférieure : est un système linéaire $Lx = b$, où L est une matrice inversible triangulaire inférieure d'ordre n.

Algorithme de résolution:

$$\begin{cases} y_{1=\frac{b_1}{L_{11}}} \\ pour\ i = 2,n \\ y_i = \dfrac{1}{L_{ii}}\left[b_i - \displaystyle\sum_{j=1}^{i-1} L_{ij}y_j\right] \end{cases} \qquad (1.2)$$

2. Méthodes directes de résolution du système $\boldsymbol{Ax = b}$

2.1. Méthode de Gauss:

Principe: Transformer le système $Ax = b$ à un système équivalent $\bar{A}x = \bar{b}$, avec $\bar{A}$ triangulaire supérieure. L'algorithme pour rendre une matrice A d'ordre n triangulaire supérieure est donné par :

Algorithme de transformation

$$\begin{array}{ll} \text{Pour} & i = 1, n-1 \\ \text{Pour} & k = i+1, n \\ & M = A_{ki}/A_{ii} \\ & b_k = b_k - M \times b_i \\ \text{Pour} & j = i, n \\ & A_{kj} = A_{kj} - M \times A_{ij} \end{array} \qquad (2.1)$$

Manipulation:
Ecrire un programme script Scilab qui permet de lire une matrice A et un vecteur b, puis appelle une fonction **Gauss** pour la triangularisation de A en utilisant l'algorithme (2.1).
Le programme appelle ensuite une autre fonction **triansup** pour la résolution du système triangulaire obtenus en utilisant l'algorithme (1.1).

2.2 Méthode de Crout LU

Principe : La méthode de Crout consiste à décomposer la matrice A en $A = L \times U$,

où : $L \in \mathcal{M}_n(\mathbb{R})$ est une matrice triangulaire inférieure.

$U \in \mathcal{M}_n(\mathbb{R})$ est une matrice triangulaire supérieure avec $U_{ii} = 1, \forall\, i$.

Algorithme de décomposition :

$$\begin{cases} \text{Pour } i = 1, n & \mathrm{U_{ii}} = 1 \\ \text{Pour } r = 1, n \,.\, \text{Pour } i = r, n & L_{ir} = A_{ir} - \sum_{k=1}^{r-1} L_{ik} \times U_{kr} \\ \text{Pour } j = r+1, n & U_{rj} = \frac{1}{L_{rr}} \left[A_{rj} - \sum_{k=1}^{r-1} L_{rk} \times U_{kj} \right] \end{cases} \qquad (2.2)$$

La résolution du système $\mathrm{Ax} = \mathrm{b} \Leftrightarrow \mathrm{LUx} = \mathrm{b} \Leftrightarrow \begin{cases} \mathrm{Ly} = \mathrm{b} \\ \mathrm{Ux} = \mathrm{y} \end{cases}$

Manipulation

Ecrire un programme script Scilab qui permet de lire une matrice A et un vecteur b puis appelle la fonction **Crout** pour la décomposition de *A* en utilisant l'algorithme (2.2). Le programme appelle ensuite une fonction **trianinf** pour la résolution du système $\mathrm{Ly} = \mathrm{b}$ en utilisant l'algorithme (1.2)**,** et une autre fonction **triansup** pour la résolution du système $\mathrm{Ux} = \mathrm{y}$ en utilisant l'algorithme (1.1).

Application: $\mathrm{A} = \begin{bmatrix} 5 & 1 & 1 \\ 1 & 7 & 1 \\ 1 & 1 & 3 \end{bmatrix}$, $\mathrm{b} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$