ESPRIT : Ecole Sup Privée d'Ingénierie et de Technologies

Méthodes Directes de Résolution des Systèmes Linéaires

Khlifi Med Naceur

Fezzani Med Amine

Encadrant : Imen BenGharbia

Rapport de stage d'été
Diplôme d'ingénieur en Informatique

Introduction

L'analyse numérique est une branche à l'interface des mathématiques et de l'informatique. Elle s'intéresse par l'étude des algorithmes permettant de résoudre numériquement les problèmes de mathématiques continues.

Cela signifie qu'elle s'occupe principalement de répondre de façon numérique à des questions à variable réelle ou complexe, la recherche de solution numérique d'équations différentielles et d'autres problèmes liés survenant dans les sciences physiques et l'ingénierie. Branche des mathématiques appliquées, son développement est étroitement lié à celui des outils informatiques.

L'un des problèmes fondamentaux de l'analyse numérique matricielle c'est le calcul des solutions des systèmes linéaires, de nombreux efforts sont été consacrés pour ce propos. Les méthodes standards incluent l'élimination de Gauss, la décomposition LU...

Table des matières

Chapitre 1

Rappel et compléments sur les matrices

Dans ce chapitre on va faire un briève rappel sur les système linéaires, les matrices, ses propriétés et quelques methodes directes pour résoudre les systèmes linéaires qui sont fréquemment utilisés dans des larges domaines comme les sciences sociales et l'économie.

1.1 Matrices et vecteurs

Une matrice est un tableau de valeurs, généralement, numériques sous la forme :

$$A_{m\times n} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

On peut utiliser 3 opérations pour manipuler les matrices :
(1) Multiplier les lignes par une constante $\lambda \neq 0$. $(\lambda E_i) \rightarrow (\lambda E_j)$
(2) Appliquer des combinaisons linéaires. $(E_i + \lambda E_j) \rightarrow (E_i)$
(3) Echange entre E_i et E_j. $(E_i) \rightarrow (E_j)$

1.1.1 Inversions des matrices

Une matrice $A \in M_n(\Re)$ est inversible s'il existe une matrice de $M_n(\Re)A^{-1}$ telle que $A^{-1}A = AA^{-1} = I_n$

1.1.2 Déterminant de matrice

On définit le déterminant de A comme suit : $\det(A) = \sum_{\sigma\in G_n} \epsilon(\sigma) \prod_{i=1}^{n} X_{\sigma(j),(j)}$
avec G_n désigne l'ensemble des permutations de $\{1, \cdots, n\}$ et $\varepsilon(\sigma)$ la signature de la permutation σ (1 pour une permutation paire, -1 pour une permutation impaire).

1.2 Décompositions matricielles

1.2.1 Elimination de Gauss

Definition

Soit $A \in M_n(\Re)$ inversible ou non. Il existe au moins une matrice inversible M telle que MA soit triangulaire supérieure.

Résolution

On considère le système suivant : Ax = b, où A est triangulaire inversible. 1. Si A triangulaire inférieure, alors on a n équations sous la forme :

$$\begin{cases} a_{1,1}x_1 + 0 + \cdots + 0 & = b1 \\ a_{2,1}x1 + a_{2,2}x_2 + \cdots + 0 & = b2 \\ \vdots + \cdots + \cdots + \cdots + \ddots & = \vdots \\ a_{n,1}x_1 + a_{n,2}x_2 + \cdots + a_{n,n}x_n & = b_n \end{cases}$$

On aura donc :

$$x_1 = \frac{b1}{a_{1,1}}$$

et pour $i = 2, 3, \cdots, n$

$$x_i = \frac{1}{a_{i,i}}(b_i - \sum_{j=1}^{i-1} a_{i,j}x_j)$$

Cet algorithme est appelé méthode de descente.

2. Si A est triangulaire supérieure, alors on a n équations sous la forme :

$$\begin{cases} a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,n}x_n & = b1 \\ 0 + a_{2,2}x_2 + \cdots + a_{2,n}x_n & = b2 \\ \vdots + \cdots + \cdots + \ddots & = \vdots \\ 0 + 0 + \cdots + \cdots + a_{n,n}x_n & = b_n \end{cases}$$

On aura donc :

$$x_1 = \frac{bn}{a_{n,n}}$$

et pour $i = n-1, \cdots, 2, 1$

$$x_i = \frac{1}{a_{i,i}}(b_i - \sum_{j=i+1}^{n} a_{i,j}x_j)$$

Cet algorithme est appelé méthode de remontée.
Ces deux algorithmes nécessitent le meme nombre d'opérations :
* $\frac{n(n-1)}{2}$ additions.
* $\frac{n(n-1)}{2}$ multiplications.
* n divisions.

1.2.2 Décomposition LU

Le principe de la méthode de décomposition LU est de se ramener à deux systèmes triangulaires. Prenons le système Ax = b, tel que A une matrice dont les mineurs principaux sont non nuls (Critère de Sylvester). Avant la résolution, il faut vérifier l'existence et l'unicité du couple :

L'unicité

Supposons qu'il existe deux décompositions LU de A : $A = L_1U_1 = L_2U_2$ avec $L_i(i = 1, 2)$ triangulaire inférieure, et Ui (i = 1, 2) triangulaire supérieure. Alors, $L_1U_1 = L_2U_2 \iff U_1U_2^{-1} = L_1^{-1}L_2$ qui est à la fois triangulaire supérieure (à gauche) et triangulaire inférieure (à droite) à Ceci prouve que $U_1U_2^{-1}$ et $L_1^{-1}L_2$ sont des matrices diagonales, et puisque $L_i, i = 1$, on a : $L_1 = L_2$ $U_1 = U_2$ D'où l'unicité.

L'existence : Par récurrence sur n

Pour n = 1, le résultat est vrai.
On suppose que le résultat est aussi vrai pour n-1 On décompose la matrice A par blocs sous la forme suivante :

$$A = \begin{pmatrix} A_{n-1} & a \\ a^t & a_{n,n} \end{pmatrix}$$

avec $a = (a_{1,n}, a_{2,n}, \cdots, a_{n-1,n})^t$ et A_{n-1} : matrice carrée d'ordre $n-1$ formée de $n-1$ premières lignes et colonnes de A.
On a A_{n-1} vérifie l'hypothèse de récurrence à $A_{n-1} = L_{n-1}U_{n-1}$
Cherchons L et U décomposées par blocs sous la forme suivante :

$$L = \begin{pmatrix} L_{n-1} & 0 \\ L^t & 1 \end{pmatrix} U = \begin{pmatrix} U_{n-1} & u \\ 0 & U_{n,n} \end{pmatrix}$$

En effectuant le produit par bloc et par identification on obtient le système d'équations :

$$\begin{cases} A_{n-1} = L_{n-1}U_{n-1} \\ b^t = L^tU_{n-1} \\ c = L_{n-1}u \\ a_{n,n} = L_tu + u_{n,n} \\ L^t = b^t(U_{n-1})^{-1} \\ u = (L_{n-1})^{-1}c \\ u_{n,n} = a_{n,n} - b^t(A_{n-1})^{-1}c \end{cases}$$

1.2.3 Méthode de Cholesky

Le principe de la méthode de Cholesky est de se ramener aussi à deux systèmes triangulaires.

Considérons le système linéaire Ax = b, avec A une matrice symétrique définie positive. Il faut, comme dans la méthode de décomposition LU, vérifier l'unicité et l'existence d'une telle factorisation :

L'unicité

Supposons que $A = LL^t = MM^t$ tels que $L_{i,i} > 0$ et $M_{i,i} > 0; \forall i \in 1, \cdots, n$. La matrice $M^{-1}L = M^tL^{-t}$ est à la fois triangulaire inférieure (à gauche) et triangulaire supérieure (à droite) c'est une matrice diagonale donc, $m_{i,i}^{-1}l_i, i = m_{i,i}l_{i,i-1}$
La condition de positivité implique $\rightarrow m_{i,i}^{-1}Xl_{i,i} = mi, iXl_{i,i}^{-1} = 1$
Ainsi, $M^{-1}L = M^tL^{-t} = I_n$ c'est à dire $L = M$
D'où l'unicité.

L'existence : Par récurrence sur n

Pour $n = 1$, on prend : $L = \sqrt{a_{1,1}}$
Supposons que le décomposition de Cholesky existe pour toute matrice définie positive d'ordre $n - 1$:

$$A = \begin{pmatrix} A_{n-1} & a \\ a^t & a_{n,n} \end{pmatrix}$$

avec $a = (a_{1,n}, a_{2,n}, \cdots, a_{n-1,n})^t$
Notons : $A_{n-1} = L_{n-1}L_{n-1}^t$ la décomposition de Cholesky de A_{n-1}, on a :

$$L = \begin{pmatrix} L_{n-1} & 0 \\ u^t & \alpha \end{pmatrix} \times \begin{pmatrix} U_{n-1} & u \\ 0 & \beta \end{pmatrix}$$

En prenant $L_{n-1}u = a$ et $\|u\|^2 + \alpha\beta = a_{n,n}$ on obtiendra la décomposition de A si l'on prend $\alpha = \beta$ ce qui sera impossible si $\alpha\beta < 0$ L'égalité ci-dessus prouve que : $\det(A) = \det(L_{n-1})\alpha \det(L_{n-1}^t)\beta$. Comme, par hypothèse $\det(A) > 0$ et $(\det(L_{n-1}))^2 > 0$, donc on a bien $\alpha\beta > 0$.

Remarque

La décomposition de Cholesky permet de calculer facilement le déterminant de A, qui est égal au carré du produit des éléments diagonaux de la matrice L, puisque :

$$det(A) = \det(LL^t) = (\det(L))^2 = (\prod_{i-1}^{n} l_{i,i})^2$$

1.2.4 Méthode de Householder ou QR

La méthode QR se base sur décomposer une matrice inversible $A \in M_n(\Re)$ en deux matrices : l'une orthogonale Q et l'autre triangulaire supérieure R à éléments diagonaux strictement positifs. donc la factorisation est unique. On doit vérifier l'unicité et l'existence.

L'unicité

Supposons que la matrice A possède deux décompositions $A = Q_1R_1$ et $A = Q_2R_2$, les coefficients diagonaux respectifs de R1 et de R2 ayant le même signe.
Alors $Q_1R_1 = Q_2R_2 \iff Q_2^{-1}Q1 = R_2R^{-1}\ 1$. La matrice $M = R_2R_1^{-1}$ est triangulaire supérieure, ses coefficients diagonaux sont les quotients deux à deux de ceux de R2 et R1 et

sont donc strictement positifs. La matrice $M = Q^{-1}$ 2. Q1 est orthogonale. La matrice M^{-1} est donc à la fois triangulaire supérieure et triangulaire inférieure. Les matrices M et M^{-1} sont donc diagonales . Du fait de l'égalité $M^{-1} = M^t$, chaque coefficient diagonal de M est égal à son propre inverse et vaut 1 ou -1, mais on sait que ces coefficients sont positifs : ils sont donc égaux à 1. On en déduit que $M = I_n$ (c-à-d $Q_1 = Q_2$ et $R_1 = R_2$).
D'où l'unicité.

L'existence :

Soit $A = Q^+R^+$ la décomposition de A (les coefficients diagonaux de R étant strictement positifs).
Soit X un sous-ensemble quelconque de $\{1, \cdots, n\}$.
Soit J la matrice diagonale dont les coefficients diagonaux successifs λi valent 1 si $i \in X$ et -1 sinon.
J est donc une matrice orthogonale qui vérifie $J_2 = I_n$. L'égalité $A = Q^+R^+$ s'écrit donc aussi $A = QR$ avec $Q = Q^+J$ et $R = JR^+$. La matrice Q est orthogonale (elle se déduit de Q^+ en inversant les signes dans les colonnes d'indice j n'appartient pas à X) et R est triangulaire supérieure (elle se déduit de R^+ en inversant les signes dans les lignes d'indice i n'appartient pas à X).
Alors on a formé une décomposition.

Chapitre 2

Algorithmes de différentes solutions des systèmes linéaires

2.1 Algorithme de l'élimination de Gauss

2.1.1 Le principe

Soit le système linéaire $Ax = b$ avec A une matrice carrée d'ordre n qui est inversible. L'idée de la méthode de Gauss : Si A n'est pas triangulaire, on est amené à trouver une matrice inversible M telle que MA soit triangulaire supérieure.

La méthode d'élimination de Gauss se décompose en trois étapes :
– Étape 1 : trouver une matrice inversible M telle que la matrice MA soit triangulaire supérieure (élimination successive des inconnues) ;
– Étape 2 : calcul simultané du vecteur Mb ;
– Étape 3 : résolution du système triangulaire supérieure $MAx = Mb$ (remontée). Pour la méthode d'élimination de Gauss on distingue deux cas :

• Sans pivotement : on ne permet pas la permutation de deux lignes ; dans certain cas on ne pourra pas obtenir de matrice triangulaire supérieure.

• Avec pivotement : on permet la permutation des lignes ; dans ce cas on obtiendra toujours une matrice triangulaire supérieure si la matrice est inversible.

2.1.2 L'algorithme

Algorithme d'élimination de Gauss dans le cas d'un pivot nul :

Entrées : une matrice A d'ordre n et un vecteur b

```
Initialisation : AA = [A, b]
pour k = 1 à n faire
    p ← AA[k, k]
    l ← k
    pour i = k à n faire
        si |AA[i, k]| > 0 alors
            p ← AA[i, k]
            l ← i
        fin si
    fin pour
    si l ≠ k alors
        pour j = k à n + 1 faire
            temp ← AA[k, j]
            AA[l, j] ← AA[l, j]
            AA[l, j] ← temp
        fin pour
    fin si
    pour i = k + 1 à n faire
        q ← AA[i, k]
        AA[i, k] ← 0
        pour j = k + 1 à n + 1 faire
            AA[i, j] = AA[i, j] − (q/p) AA[k, j]
        fin pour
    fin pour
fin pour
```

Sortie : AA la matrice triangularisée

Figure 2.1 : *Algorithme d'élimination de Gauss*

2.2 Algorithme de la décomposition LU

2.2.1 Le principe

Le principe de la méthode est de se ramener à deux systèmes triangulaires. Soit le système linéaire suivant : $Ax = b$, où A est une matrice dont tous les mineurs principaux sont non nuls, alors pour la résolution on procède par les étapes suivantes :
1. Décomposition de A : $A = LU$ avec L est triangulaire inférieure à diagonale unité et U triangulaire supérieure ;
2. Résolution du système triangulaire inférieur : $Ly = b$ (descente) ;
3. Résolution du système triangulaire supérieur : $Ux = y$ (remontée).

$$Ax = b \longrightarrow LUx = b \rightarrow \begin{cases} Ly = b \\ Ux = y \end{cases}$$

2.2.2 L'algorithme

Algorithme de factorisation LU :
Entrée : une matrice carrée A d'ordre n dont tous les mineurs principaux sont non nuls

Initialisation : $U = A$ et $L = I_n$
pour $k = 1$ à $n - 1$ faire
    $p = U[k, k]$
    pour $i = k + 1$ à n faire
        $q = U[i, k]$
        $U[i, k] = 0$
        $L[i, k] = \frac{q}{p}$
        pour $j = k + 1$ à n faire
            $U[i, j] = U[i, j] - \frac{q}{p} U[k, j]$
        fin pour
    fin pour
fin pour

Sorties : la matrice triangulaire inférieure L et la matrice triangulaire supérieure U

FIGURE 2.2 : *Algorithme de décomposition LU*

Algorithme de résolution du système triangulaire inférieur :
Entrées : la matrice triangulaire inférieure L et le vecteur b
Initialisation : $y[1:n]=0$ $y[1]=b[1]$ pour $i=2$ à n faire $\quad y[i]=b[i]-\sum_{j=1}^{i-1} L[i,j]y[j]$ fin pour
Sortie : le vecteur y

FIGURE 2.3 : *Algorithme de résolution du système triangulaire inférieur*

Algorithme de résolution du système triangulaire supérieur :
Entrées : la matrice triangulaire supérieure U et le vecteur y
Initialisation : $x[1:n]=0$ $x[n]=\dfrac{y[n]}{U[n,n]}$ pour $i=n-1$ à 1 faire $\quad x[i]=\dfrac{1}{U[i,i]}\left(y[i]-\sum_{j=i+1}^{n} U[i,j]x[j]\right)$ fin pour
Sortie : le vecteur x

FIGURE 2.4 : *Algorithme de résolution du système triangulaire supérieur*

2.3 Algorithme de la décomposition de Cholesky

2.3.1 Le principe

Soit le système linéaire suivant : Ax = b, où A est une matrice symétrique définie positive, alors pour la résolution on procède par les étapes suivantes :
1. Décomposition de A : $A = LL^t$ avec L est triangulaire inférieure à éléments diagonaux positifs ;
2. Résolution du système triangulaire inférieur : $Ly = b$ (descente) ;
3. Résolution du système triangulaire supérieur : $LTx = y$ (remontée).

$$Ax = b \longrightarrow LL^t x = b \rightarrow \begin{cases} Ly = b \\ L^t x = y \end{cases}$$

2.3.2 L'algorithme

Algorithme de décomposition de Cholesky :

Entrée : une matrice symétrique définie positive A d'ordre n

Initialisation : $L[1:n, 1:n] = 0$

$L[1,1] = \sqrt{A[1,1]}$

pour $i = 2$ à n faire

$\quad L[i,1] = \dfrac{A[i,1]}{L[1,1]}$

fin pour

pour $j = 2$ à n faire

$\quad L[j,j] = \sqrt{A[j,j] - \sum_{k=1}^{j-1} L[j,k]^2}$

$\quad$ pour $i = j$ à n faire

$\quad\quad L[i,j] = \dfrac{1}{L[j,j]} \left(A[i,j] - \sum_{k=1}^{j-1} L[i,k]L[j,k] \right)$

$\quad$ fin pour

fin pour

Sortie : la matrice triangulaire inférieure L

FIGURE 2.5 : *Algorithme de décomposition de Cholesky*

Algorithme de résolution du système triangulaire inférieur :
Entrées : la matrice triangulaire inférieure L et le vecteur b
Initialisation : $y[1:n]=0$ $y[1]=\dfrac{b[1]}{L[1,1]}$ pour $i=2$ à n faire $\quad y[i]=\dfrac{1}{L[i,i]}\left(b[i]-\sum_{j=1}^{i-1}L[i,j]y[j]\right)$ fin pour
Sortie : le vecteur y

FIGURE 2.6 : *Algorithme de résolution du système triangulaire inférieur*

Algorithme de résolution du système triangulaire supérieur :
Entrées : la matrice triangulaire inférieure L et le vecteur y
Initialisation : $x[1:n]=0$ et $U=L^T$ $x[n]=\dfrac{y[n]}{U[n,n]}$ pour $i=n-1$ à 1 faire $\quad x[i]=\dfrac{1}{U[i,i]}\left(y[i]-\sum_{j=i+1}^{n}U[i,j]x[j]\right)$ fin pour
Sortie : le vecteur x

FIGURE 2.7 : *Algorithme de résolution du système triangulaire supérieur*

2.4 Algorithme de la décomposition de Householder ou Méthode QR

2.4.1 Le principe

Soit le système linéaire suivant : $Ax=b$, où A est une matrice inversible, alors pour la résolution on procède par les étapes suivantes :

1. Décomposition de A : $A=QR$ avec Q est orthogonale et R triangulaire supérieure à éléments diagonaux positifs ;

2. Résolution du système : $Qy=b$;

3. Résolution du système triangulaire supérieur : $Rx = y$ (remontée).

$$Ax = b \longrightarrow QRx = b \rightarrow \begin{cases} Qy = b \\ Rx = y \end{cases}$$

La résolution du système à matrice orthogonale $Qy = b$:

$$Qy = b \Longleftrightarrow y = Q^1 b = Q^t b$$

donc la résolution de ce système se fait par un produit d'une matrice par un vecteur colonne.

2.4.2 L'algorithme

Algorithme de décomposition de Householder :

Entrée : une matrice carrée A d'ordre n

```
Initialisation : H = I_n
pour k = 1 à n − 1 faire
    α ← sqrt( Σ_{i=k}^{n} A[i,k]^2 )
    β ← α^2 − αA[k,k]
    v[k] ← A[k,k] − α
    pour i = k + 1 à n faire
        v[i] ← A[i,k]
    fin pour
    pour j = k à n faire
        c ← (1/β) Σ_{i=k}^{n} v[i]A[i,j]
        pour i = k à n faire
            A[i,j] ← A[i,j] − cv[i]
        fin pour
    fin pour
    pour j = 1 à n faire
        c ← (1/β) Σ_{i=k}^{n} v[i]H[i,j]
        pour i = k à n faire
            H[i,j] ← H[i,j] − cv[i]
        fin pour
    fin pour
fin pour
Q ← H^T et R ← A
```

Sorties : une matrice orthogonale Q et une matrice triangulaire supérieure R.

FIGURE 2.8 : *Algorithme de décomposition de Householder*

Algorithme de résolution du système à matrice orthogonale :
Entrées : la matrice orthogonale Q et le vecteur b
Initialisation : $y[1:n]=0$ pour $i=1$ à n faire $\quad y[i]=\sum_{j=1}^{n} Q[j,i]b[j]$ fin pour
Sortie : le vecteur y

FIGURE 2.9 : *Algorithme de résolution du système à matrice orthogonale*

Algorithme de résolution du système triangulaire supérieur :
Entrées : la matrice triangulaire supérieure R et le vecteur y
Initialisation : $x[1:n]=0$ $x[n]=\dfrac{y[n]}{R[n,n]}$ pour $i=n-1$ à 1 faire $\quad x[i]=\dfrac{1}{R[i,i]}\left(y[i]-\sum_{j=i+1}^{n} R[i,j]x[j]\right)$ fin pour
Sortie : le vecteur x

FIGURE 2.10 : *Algorithme de résolution du système triangulaire supérieur*

Chapitre 3

Discussion

3.1 Résultats Théoriques

Malgré les différences de calcul entre les méthodes, elles partagent une complexité algorithmique cubique de la forme $O(\alpha n^3), \alpha \in \Re$.

3.1.1 Élimination de Gauss : $\frac{2n^3}{3}$

Avantages

Résolution d'un système linéaire unique
Utilisation des opérations élémentaires (additions et multiplications

inconvénients

Résolution d'un système linéaire unique
Utilisation des opérations élémentaires (additions et multiplications)
Changement de la structure de A
Blocage dans le cas d'un pivot nul sans pivotement
Erreurs d'arrondi sur la machine

3.1.2 Décomposition LU : n^3

Avantages

Utile lors de la résolution d'une suite de systèmes linéaires de même matrice A où le vecteur b varie
Utilisation des opérations élémentaires (additions et multiplications)

inconvénients

Résolution de deux sous-systèmes linéaires triangulaires de matrices L et U
Particularité de la matrice carrée A (mineurs principaux non nuls)

3.1.3 Décomposition de Cholesky : $\frac{n^3}{6}$

Avantages

Utile lors de la résolution d'une suite de systèmes linéaires de même matrice A où le vecteur b varie
Calcul d'une seule matrice L
Moins de ressources machine consommées par rapport à la factorisation LU (stockage d'une matrice unique L)

inconvénients

Particularité de la matrice A (symétrique et définie positive) Résolution de deux sous-systèmes linéaires triangulaires

$$\begin{cases} Ly = b \\ L^t x = y \end{cases}$$

pour résoudre le système linéaires $Ax = b$

3.1.4 Décomposition QR ou Householder $\frac{4n^3}{3}$

Avantages

Utile lors de la résolution d'une suite de systèmes linéaires de même matrice A où le vecteur b varie
Résolution facile du système $Qy = b$

inconvénients

Résolution de deux sous-systèmes linéaires triangulaires de matrices Q et R
Gaspillage des ressources machine

3.2 Résultats Pratiques

Temps d'exécution sur JupyterLab (ms)

Méthode	Numéro d'essai	1	2	3	4	5	6	7	8	9	10
Gauss		2,87	4	5,07	5,93	4	3,52	3,79	4	4	5,63
LU		4	4,11	4	4,41	8,04	6,58	4,16	4	5,93	4,12
Cholesky		4	2,36	4	5,04	4,22	6,08	3,3	8,94	5,02	6,91
QR		5,55	9,3	4	5,64	6,21	6,16	8	8	4	8

Figure 3.1 : *Les temps d'éxecutions des differents algorithmes avec Python*

	Moyenne temps d'exécution (ms)
Gauss	4,3
LU	4,9
Cholesky	5
QR	6,5

FIGURE 3.2 : *Les moyennes des temps d'éxecutions des differents algorithmes avec Python*

Chapitre 4

Conclusion

Chaque méthode directe de résolution des systèmes linéaires est utile et meilleure dans un contexte spécifique. Ces algorithmes ont aidé l'humanité à résoudre beaucoup de problèmes rencontré en des domaines différents tel que l'électricité, les mathématiques appliqués à l'économie et à l'informatique.
De même pour les méthodes indirectes.
Quels sont les propriétés des méthodes indirectes de résolution des systèmes linéaires ? Et quels sont leurs algorithmes ?

Bibliographie

Numerical Methods in Engineering with python 3 by Jaan Kiusalaas,2013.
Méthodes directes de la résolution des systèmes linéaires, Ali TAMOUSSIT.

Cliquer sur ce texte pour accéder aux exécutables en Python