

**République Algérienne Démocratique  
et Populaire**  
**Ministère de l'Enseignement Supérieur  
et de la Recherche Scientifique**

**Université des Sciences et de la Technologie d 'Oran  
Mohamed Boudiaf USTO-MB**

**Faculté des Mathématiques et Informatique**  
**Département Informatique**



# Méthodes numériques

**Polycopié pédagogique  
de Cours et Travaux Pratiques**

**Dr. Redouane TLEMSANI**

Objectifs de l'enseignement :

Cette matière permettra aux étudiants d'investir le domaine des méthodes numériques nécessaires à la résolution des problèmes



## Remercîments

**J**e remercie tout *enseignant* m'a donné de la connaissance, de la motivation et de la science.

**J**e remercie tout *étudiant* a suivi mes cours avec passion et curiosité.

**J**e remercie toute *personne* m'a critiqué et m'a encouragé tout au long de ma mission d'enseignement à ce jour-là.

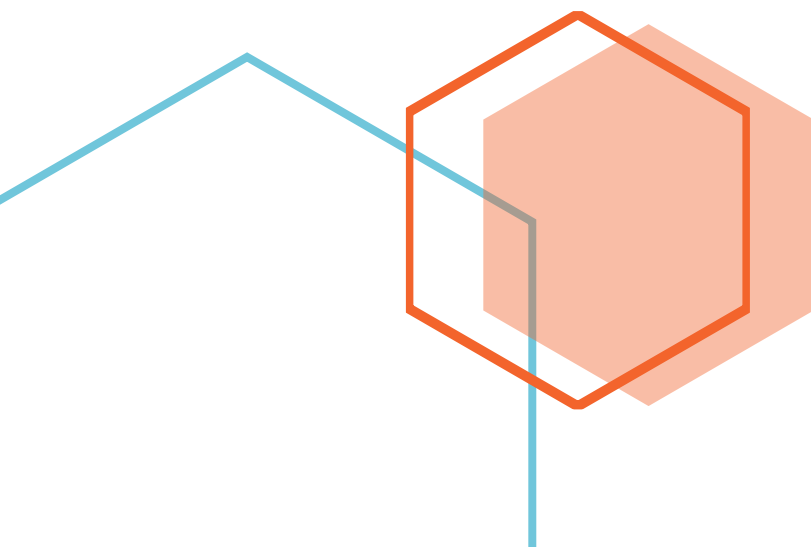


## Avant-propos

Ce polycopié se trouve être un support pédagogique de cours et travaux pratiques, destiné aux étudiants de 2<sup>ème</sup> Année des Licences LMD assurées au département d'informatique, faculté des mathématiques et d'informatique de l'USTO-MB pour le 3<sup>ème</sup> semestre comme unité d'enseignement méthodologique avec le nombre de crédits égal à 4 et le coefficient égal à 2.

Ce polycopié regroupe un certain nombre de méthodes étudiées dans les différents chapitres du cours de méthodes numériques.

L'objectif des TP est d'implémenter sous MATLAB les différentes méthodes pour les comparer et mieux les comprendre.





## *Table des matières*

<b>Avant-propos</b> .....	0
<b>Introduction générale</b> .....	5
<b>Chapitre I : Généralités sur l'analyse numérique et le calcul scientifique</b> .....	6
<b>I.1 Introduction</b> .....	6
<b>I.2 Erreurs absolue et relative</b> .....	6
<b>I.2.1 Erreur absolue</b> .....	7
<b>I.2.2 Erreur relative</b> .....	7
<b>I.2.3 Majorants des erreurs absolue et relative</b> .....	8
<i>I.2.3.2 Remarque :</i> .....	8
<b>I.3. Propagation des erreurs</b> .....	10
<b>I.3.1 Cas de l'addition</b> .....	10
<i>I.3.1.1 Preuve</i> .....	10
<b>I.3.2 Cas de la soustraction</b> .....	10
<i>I.3.2.1 Preuve</i> .....	10
<i>I.3.2.3 Exemples</i> .....	11
<b>I.3.3 Cas de la multiplication</b> .....	12
<i>I.3.3.1 Preuve</i> .....	12
<b>I.3.4 Cas de la division</b> .....	12
<i>I.3.4.1 Preuve</i> .....	12
<b>I.4 Représentation décimale des nombres approchés</b> .....	13
<b>I.4.2 Définition</b> .....	14
<i>I.4.2.1 Exemples</i> .....	14
<b>I.5 Chiffres exactes d'un nombre décimal approché</b> .....	14
<i>I.5.1.1 Exemples</i> .....	15
<b>I.5.2 Propriétés</b> .....	15



1.5.2.1 Remarque.....	15
<b>I.6 Troncature et arrondissement d'un nombre .....</b>	<b>15</b>
<b>I.6.1 Règle d'arrondissement .....</b>	<b>15</b>
1.6.1.1 Exemples.....	16
<b>I.7 Conclusion.....</b>	<b>17</b>
<b>Chapitre II : Méthodes directes de résolution des systèmes d'équations linéaires .....</b>	<b>18</b>
<b>II.1. Introduction .....</b>	<b>18</b>
<b>II.2. Méthodes directes .....</b>	<b>19</b>
II.2.1 Méthodes d'élimination de Gauss .....	19
II.2.1.1 Résolution d'un système triangulaire.....	19
II.2.1.2 Méthode de Gauss .....	19
II.2.1.3 Algorithme de Gauss .....	21
<b>II.3.2 Méthode de décomposition LU .....</b>	<b>22</b>
II.2.2.1 Introduction.....	22
II.3.2.2 Principe .....	22
II.2.2.3 Algorithme de factorisation LU.....	24
II. 2.2.4 Utilité de la factorisation LU .....	25
<b>II.3.3 Méthode de Cholesky .....</b>	<b>25</b>
II.2.3.1 Rappel .....	25
II.2.3.2 Théorème : (Cholesky) .....	26
II.2.3.3 Algorithme de Cholesky .....	28
<b>II.3. Calcul du déterminant.....</b>	<b>29</b>
<b>II.4 Conclusion .....</b>	<b>29</b>
<b>Chapitre III : Méthodes itératives de résolution des systèmes d'équations linéaires .....</b>	<b>30</b>
<b>III.1 Introduction.....</b>	<b>30</b>
<b>III.2 Principe .....</b>	<b>30</b>
III.2.1 Théorème : .....	32
III.2.2 Décomposition de la matrice A.....	32
<b>III.3 Méthode de Jacobi .....</b>	<b>33</b>
III.3.1 Principe.....	33
III.3.2 Test d'arrêt .....	34
III.3.3 Algorithme de Jacobi .....	34
III.3.4 Conditions de convergence .....	35
III.3.5 Exemple : .....	36
<b>III.4 Méthode de Gauss-Seidel.....</b>	<b>37</b>
III.4.1 Principe.....	37
III.4.2 Algorithme de Gauss-Seidel.....	38



III.4.3 Convergence de la méthode de Gauss-Seidel .....	38
III.4.4 Exemple.....	39
<b>III.5 Méthode de Relaxation .....</b>	<b>41</b>
III.5.1. Introduction .....	41
III.5.2. Description .....	41
III.5.3. Méthodes SOR et SSOR et Jacobi.....	42
III.5.4 Résolution Matricielle.....	45
<b>III.6 Convergence des méthodes itératives .....</b>	<b>46</b>
III.6.1. Conditions Suffisantes.....	46
III.6.2. Condition nécessaire & Suffisantes .....	46
<b>III.7 Conclusion.....</b>	<b>54</b>
<b>Chapitre IV : Calcul de valeurs et de vecteurs propres .....</b>	<b>55</b>
<b>IV.1 Introduction .....</b>	<b>55</b>
<b>IV.2 Localisation des valeurs propres.....</b>	<b>55</b>
IV.2.1. Définition : .....	55
IV.2.2. Calcul analytique des valeurs propres et des vecteurs propres.....	56
IV.2.3. La localisation des valeurs propres : .....	57
<b>IV.3 Méthode de la puissance .....</b>	<b>60</b>
<b>IV.3 Conclusion.....</b>	<b>65</b>
<b>Chapitre V : Analyse matricielle .....</b>	<b>66</b>
<b>V.1 Introduction.....</b>	<b>66</b>
<b>V.2 Espaces vectoriels.....</b>	<b>66</b>
<b>V.3 Matrices .....</b>	<b>67</b>
V.3.1 Opérations sur les matrices.....	68
V.3.2 Liens entre applications linéaires et matrices.....	69
V.3.3 Inverse d'une matrice .....	70
V.3.4 Trace et déterminant d'une matrice.....	70
V.3.5 Valeurs et vecteurs propres .....	72
V.3.6 Matrices semblables .....	72
V.3.7 Quelques matrices particulières.....	73
<b>V.4 Normes et produits scalaires.....</b>	<b>74</b>
V.4.1 Définitions .....	74
V.4.2 Produits scalaires et normes vectoriels.....	74
V.4.3 Normes de matrices .....	75
<b>V.5 Conclusion .....</b>	<b>76</b>
<b>Conclusion générale.....</b>	<b>77</b>
<b>Fiche de TP 1.....</b>	<b>78</b>



<b>Solution de la fiche de TP 1</b> .....	80
<b>Fiche de TP 2</b> .....	82
<b>Solution de la fiche de TP 2</b> .....	83
<b>Fiche de TP 3</b> .....	85
<b>Solution de la fiche de TP 3</b> .....	87
<b>Fiche de TP 4</b> .....	90
<b>Solution de la fiche de TP 4</b> .....	90
<i>Rappel</i> .....	90
<i>Théorème : (Cholesky)</i> .....	90
<i>Algorithme de Cholesky</i> .....	92
<b>Fiche de TP 5</b> .....	95
<i>Valeurs propres et vecteurs propres</i> .....	95
Références.....	96
<b>Annexe</b> .....	97
<b>Introduction à Matlab</b> .....	97
<i>Les Commandes Matlab</i> .....	97
<b>Espace E-learning</b> .....	101



## Introduction générale

La mise en œuvre d'une méthode numérique sur une machine amène un certain nombre de difficultés **supplémentaires, liées à une nécessaire représentation approchée des nombres.**

Un algorithme peut à son tour contenir des sous-algorithmes et doit pouvoir s'achever après un nombre fini d'opérations élémentaires afin de pouvoir être utilisé dans un programme informatique.

Le but de ce polycopie est de présenter plusieurs méthodes numériques de base (directes ou itératives) utilisées pour la résolution des systèmes linéaires, calcul et localisation des valeurs propres ainsi que l'analyse matricielle notamment les notions de convergence, de précision et de stabilité des méthodes enseignées.

La présentation et l'analyse des méthodes sont suivies d'une implémentation et d'applications réalisées sous formes de travaux pratiques par les étudiants avec les logiciels Matlab.



# Chapitre I : Généralités sur l'analyse numérique et le calcul scientifique

## I.1 Introduction

En analyse numérique, l'erreur d'approximation de certaines données est la différence entre une valeur exacte et une certaine valeur approchée ou approximation de celle-ci. Une erreur d'approximation peut se produire lorsque :

- La mesure des données n'est pas précise (en raison des instruments) ;
- L'emploi de valeurs approchées au lieu des valeurs exactes (par exemple, 3,14 au lieu de  $\pi$ ).

Parmi les types d'erreurs, on peut citer :

- ❖ Les erreurs initiales
- ❖ Les erreurs de discrétisation
- ❖ Les erreurs de troncature
- ❖ Les erreurs d'arrondi

## I.2 Erreurs absolue et relative

Quantité exacte :  $1, \frac{1}{2}, \sqrt{2}, \pi, \dots$

Quantité approchée (approximative) :  $e \cong 2.718, \pi \cong 3.14, \dots$

Soient  $x$  une quantité exacte et  $x^*$  une valeur approchée de  $x$  :

- Si  $x^* > x$ ,  $x^*$  est dite valeur approchée par excès.
- Si  $x^* < x$ ,  $x^*$  est dite valeur approchée par défaut.

**Exemple :** pour le nombre  $\sqrt{5}$ , la valeur 2.23 est une valeur approchée par défaut et 2.24 est une valeur approchée par excès. On peut donc écrire l'encadrement  $2.23 < \sqrt{5} < 2.24$ .



## I.2.1 Erreur absolue

*I.2.1.1 Définition :* On appelle erreur absolue de  $x^*$  (sur  $x$ ), la quantité  $E = |x - x^*|$ .

*I.2.1.2 Remarque :* Plus l'erreur absolue de  $x^*$  est petite, plus  $x^*$  est précise.

*I.2.1.3 Exemple :* pour la valeur exacte  $x=2/3$ , la valeur approchée  $x_1^* = 0.666667$  est mille fois plus précise que  $x_2^* = 0.667$ .

En effet :

$$E_1 = |x - x_1^*| = \left| \frac{2}{3} - 0.666667 \right| = \left| \frac{2000000 - 2000001}{3 \cdot 10^6} \right| = \frac{1}{3} 10^{-6}$$

$$E_2 = |x - x_2^*| = \left| \frac{2}{3} - 0.667 \right| = \left| \frac{2000 - 2001}{3 \cdot 10^3} \right| = \frac{1}{3} 10^{-3}$$

Il vient que :  $E_2 = 10^3 \cdot E_1$ ; ceci signifie que  $E_2$  est 1000 fois plus grande que  $E_1$  et ainsi,  $x_1^*$  est 1000 fois plus précise que  $x_2^*$ .

## I.2.2 Erreur relative

*I.2.2.1 Définition :* On appelle erreur relative à  $x^*$  la quantité  $E_r = \frac{|x-x^*|}{|x|} = \frac{E}{|x|}$ .

*I.2.2.2 Remarques*

- On exprime souvent l'erreur relative en pourcentage.
- Dans les développements mathématiques, on utilise la plupart du temps l'erreur absolue. L'erreur relative est donnée par les mesures expérimentales et est utilisée surtout dans ce contexte.
- Il est évident qu'il faut toujours manipuler le plus complètement possible les expressions mathématiques (qui sont exactes) avant de passer à l'ordinateur ou à la calculatrice qui n'utilisent que des quantités approchées et des opérations arithmétiques entraînant les erreurs d'arrondi et de troncature.



*I.2.2.3 Exemple :* Pour les valeurs exactes  $x = 2/3$  et  $y = 1/15$ , on considère les valeurs approchées respectives  $x^* = 0.67$  et  $y^* = 0.07$ .

Les erreurs absolues respectives sont :

$$E_1 = |x - x^*| = \left| \frac{2}{3} - 0.67 \right| = \left| \frac{200 - 201}{3 \cdot 10^2} \right| = \frac{1}{3} 10^{-2}$$

$$E_2 = |y - y^*| = \left| \frac{1}{15} - 0.07 \right| = \left| \frac{100 - 105}{15 \cdot 10^2} \right| = \frac{1}{3} 10^{-2}$$

Les erreurs relatives correspondantes sont :

$$E_{r1} = \frac{E_1}{|x|} = 0.5 \cdot 10^{-2} = 0.5\%$$

$$E_{r2} = \frac{E_2}{|y|} = 5 \cdot 10^{-2} = 5\%$$

Ainsi, bien que les erreurs absolues soient égales,  $x^*$  est une approximation 10 fois plus précise pour  $x$  que  $y^*$  l'est pour  $y$ .

### I.2.3 Majorants des erreurs absolue et relative

Si la valeur exacte est connue, on peut déterminer les erreurs absolue et relative. Mais dans la majorité des cas, elle ne l'est pas. Les erreurs absolue et relative deviennent alors inconnues, et pour les apprécier, on introduit la notion de majorant de l'erreur.

*I.2.3.1 Définition :* On appelle majorant de l'erreur absolue d'une valeur approchée  $x^*$ , tout nombre réel positif  $\Delta x$  qui vérifie :  $E = |x - x^*| \leq \Delta x$

Ce qui est équivalent à :  $x^* - \Delta x \leq x \leq x^* + \Delta x$

*I.2.3.2 Remarque :*

- ✚ Plus  $\Delta x$  est petite, plus l'approximation  $x^*$  est précise. On prend en pratique le plus petit  $\Delta x$  possible.



- ✚ On écrit :  $x = x^* \mp \Delta x$  ou encore  $x \cong x^*(\mp \Delta x)$ , qui veut dire :  $x \in [x^* - \Delta x, x^* + \Delta x]$
- ✚ A défaut de l'erreur absolue effective,  $\Delta x$  est appelé par abus de langage, erreur absolue de  $x^*$ .
- ✚ Si  $x_1$  et  $x_2$  sont tels que :  $x_1 \leq x \leq x_2$  alors  $x^* = \frac{x_1 + x_2}{2}$  est une approximation de  $x$  avec une erreur absolue  $\Delta x = \frac{x_2 - x_1}{2}$ .

**Exemple :** On sait que :  $3.14 \leq \pi \leq 3.15$ , d'où :  $x^* = \frac{3.14 + 3.15}{2} = 3.145$  est une approximation de  $\pi$  avec une erreur absolue :  $\Delta x = \frac{3.15 - 3.14}{2} = \frac{0.01}{2} = 0.005$ .

- ✚ Souvent au lieu d'écrire  $x = x^* \mp \Delta x$  ou  $x \cong x^*(\mp \Delta x)$ , on écrit  $x = x^* \mp \delta x \cdot 100\%$  ou  $x \cong x^*(\mp \delta x \cdot 100\%)$  avec :  $\delta x = \frac{\Delta x}{|x^*|}$  qui est un majorant de l'erreur relative à  $x^*$ .
- ✚ A défaut de l'erreur relative effective,  $\delta x$  est appelée par abus de langage, erreur relative à  $x^*$ .

Si on connaît l'erreur relative  $\delta x$ , on en déduit un encadrement du nombre exacte  $x$  :

$$x^*(1 - \delta x) \leq x \leq x^*(1 + \delta x)$$

Ou encore :  $x = x^*(1 \mp \delta x)$ .

**Exemple :** un poids est donné par :  $50Kg \mp 1\%$ .

L'erreur relative à la valeur approchée  $x^* = 50Kg$  est donc  $\delta x = 0.01$ .

L'erreur absolue est donc :  $\Delta x = x^* \cdot \delta x = 50 \times 0.01 = 0.5Kg$ .

Donc le poids exacte  $x$  est encadré par :  $49.5Kg \leq x \leq 50.5Kg$ .



## I.3. Propagation des erreurs

Soient  $x$  et  $y$  deux quantités exactes positives,  $x^*$  et  $y^*$  deux approximations positives de  $x$  et  $y$ ,  $\Delta x$  et  $\Delta y$  les erreurs absolues sur  $x$  et  $y$ .

### I.3.1 Cas de l'addition

1.  $\Delta(x + y) = \Delta x + \Delta y$
2.  $\delta(x + y) \leq \max(\delta x, \delta y)$

#### I.3.1.1 Preuve

1. Nous avons :  $x^* - \Delta x \leq x \leq x^* + \Delta x$  et  $y^* - \Delta y \leq y \leq y^* + \Delta y$

$$\text{Donc : } (x^* + y^*) - (\Delta x + \Delta y) \leq x + y \leq (x^* + y^*) + (\Delta x + \Delta y)$$

Ainsi,  $(\Delta x + \Delta y)$  est un majorant de l'erreur absolue de  $x+y$ , et par suite :

$$\Delta(x + y) = \Delta x + \Delta y$$

2.  $\delta(x + y) = \frac{\Delta(x+y)}{x^*+y^*} = \frac{\Delta x + \Delta y}{x^*+y^*} = \frac{\Delta x}{x^*} \cdot \frac{x^*}{x^*+y^*} + \frac{\Delta y}{y^*} \cdot \frac{y^*}{x^*+y^*}$

$$\text{On pose : } \lambda_1 = \frac{x^*}{x^*+y^*} > 0 \text{ et } \lambda_2 = \frac{y^*}{x^*+y^*} > 0 \text{ et } \lambda_1 + \lambda_2 = 1$$

$$\begin{aligned} \delta(x + y) &= \delta x \cdot \lambda_1 + \delta y \cdot \lambda_2 \leq \max(\delta x, \delta y) \cdot \lambda_1 + \max(\delta x, \delta y) \cdot \lambda_2 \\ &\leq \underbrace{(\lambda_1 + \lambda_2)}_{=1} \cdot \max(\delta x, \delta y) \leq \max(\delta x, \delta y) \end{aligned}$$

### I.3.2 Cas de la soustraction

1.  $\Delta(x - y) = \Delta x + \Delta y$
2.  $\delta(x - y) \leq \frac{x^*+y^*}{x^*-y^*} \max(\delta x, \delta y)$

#### I.3.2.1 Preuve

1. Nous avons :  $x^* - \Delta x \leq x \leq x^* + \Delta x$  et  $y^* - \Delta y \leq y \leq y^* + \Delta y$

$$\Rightarrow -y^* - \Delta y \leq -y \leq -y^* + \Delta y$$

$$d'où: (x^* - y^*) - (\Delta x + \Delta y) \leq x - y \leq (x^* - y^*) + (\Delta x + \Delta y)$$

Ainsi,  $(\Delta x + \Delta y)$  est un majorant de l'erreur absolue de  $x-y$ , et par suite :

$$\Delta(x - y) = \Delta x + \Delta y$$

2.  $\delta(x - y) = \frac{\Delta(x-y)}{x^*-y^*} = \frac{\Delta x + \Delta y}{x^*-y^*} = \frac{\Delta x}{x^*} \cdot \frac{x^*}{x^*+y^*} \cdot \frac{x^*+y^*}{x^*-y^*} + \frac{\Delta y}{y^*} \cdot \frac{y^*}{x^*+y^*} \cdot \frac{x^*+y^*}{x^*-y^*}$   
 $= [\delta x \cdot \lambda_1 + \delta y \cdot \lambda_2] \frac{x^*+y^*}{x^*-y^*}$



$$\leq \frac{x^*+y^*}{x^*-y^*} [\max(\delta x, \delta y) \cdot \lambda_1 + \max(\delta x, \delta y) \cdot \lambda_2]$$

$$\leq \frac{x^*+y^*}{x^*-y^*} \left[ \underbrace{(\lambda_1 + \lambda_2)}_{=1} \cdot \max(\delta x, \delta y) \right] \leq \frac{x^*+y^*}{x^*-y^*} \max(\delta x, \delta y)$$

Avec :  $\lambda_1 = \frac{x^*}{x^*+y^*} > 0$  et  $\lambda_2 = \frac{y^*}{x^*+y^*} > 0$  et  $\lambda_1 + \lambda_2 = 1$

### 1.3.2.2 Remarques

- ✚ La soustraction est l'opération qui fait perdre le plus de précision.
- ✚ Dans le calcul approché il convient d'éviter au possible la soustraction de deux nombres approchés à peu près égaux. Pour cette raison, il faut faire des transformations algébriques avec l'expression arithmétique donnée. Mais si une telle soustraction s'impose, les termes de la différence doivent être pris avec un nombre suffisant de s.c.e. de réserve (si possible).

### 1.3.2.3 Exemples

1. Soient  $x^*=255$  et  $y^*=250$  avec  $\delta x = \delta y = 0.1\%$

On cherche  $\delta(x - y)$

Nous avons :  $\begin{cases} \Delta x = x^* \cdot \delta x = 255 \cdot 10^{-3} = 0.255 \\ \Delta y = y^* \cdot \delta y = 250 \cdot 10^{-3} = 0.250 \end{cases}$  et  $x^* - y^* = 5$

L'erreur relative est donc :  $\delta(x - y) = \frac{\Delta(x-y)}{x^*-y^*} = \frac{\Delta x + \Delta y}{x^*-y^*} = 10.1 \cdot 10^{-2} = 10.1\%$

Ainsi :  $x^*$  et  $y^*$  sont **101 fois plus précis** pour  $x$  et  $y$  (respectivement) que  $x^*-y^*$   
l'est pour  $x-y$ .

2. Soient  $x^*=56202$  et  $y^*=56198$  avec  $\delta x = \delta y = 0.01\%$

On cherche  $\delta(x - y)$  et  $\Delta(x - y)$

Nous avons :  $x^* - y^* = 4$

$$\delta(x - y) = \frac{\Delta(x-y)}{x^*-y^*} = \frac{\Delta x + \Delta y}{x^*-y^*} = \frac{\Delta x}{x^*} \cdot \frac{x^*}{x^*-y^*} + \frac{\Delta y}{y^*} \cdot \frac{y^*}{x^*-y^*}$$

$$= \frac{x^*+y^*}{x^*-y^*} \cdot \delta x = 281\% \text{ puisque : } \delta x = \delta y$$

Ainsi :  $x^*$  et  $y^*$  sont **28100 fois plus précis** pour  $x$  et  $y$  (respectivement) que  $x^*-y^*$   
l'est pour  $x-y$ .

Pour calculer  $\Delta(x - y)$ , on a :



$$\begin{cases} \Delta x = x^* \cdot \delta x = 26202 \cdot 10^{-4} = 5.6202 \Rightarrow x = x^* \mp \Delta x = 56202 \mp 5.6202 \\ \Delta y = y^* \cdot \delta y = 26198 \cdot 10^{-4} = 5.6198 \Rightarrow x = x^* \mp \Delta x = 56198 \mp 5.6198 \\ \Delta(x - y) = (x^* - y^*) \cdot \delta(x - y) = 11.24 \\ \Rightarrow x - y = (x^* - y^*) \mp \Delta(x - y) = 5 \mp 11.24 \text{ (erreur considérable)} \end{cases}$$

### I.3.3 Cas de la multiplication

1.  $\Delta(xy) = x^* \Delta y + y^* \Delta x$
2.  $\delta(xy) = \delta x + \delta y$

#### I.3.3.1 Preuve

1. On a :  $x^* - \Delta x \leq x \leq x^* + \Delta x$  et  $y^* - \Delta y \leq y \leq y^* + \Delta y$   
On suppose que :  $x^* - \Delta x > 0$  et  $y^* - \Delta y > 0$

$$\text{Il vient que : } (x^* - \Delta x)(y^* - \Delta y) \leq xy \leq (x^* + \Delta x)(y^* + \Delta y)$$

$$\text{Et donc : } x^* y^* - x^* \Delta y - y^* \Delta x + \Delta x \Delta y \leq xy \leq x^* y^* + x^* \Delta y + y^* \Delta x + \Delta x \Delta y$$

En négligeant l'erreur du second ordre :

$$x^* y^* - [x^* \Delta y + y^* \Delta x] \leq xy \leq x^* y^* + [x^* \Delta y + y^* \Delta x]$$

Et par la suite :  $\Delta(xy) = x^* \Delta y + y^* \Delta x$  puisque  $[x^* \Delta y + y^* \Delta x]$  est un majorant de l'erreur.

2.  $\delta(xy) = \frac{\delta(xy)}{x^* y^*} = \frac{x^* \Delta y + y^* \Delta x}{x^* y^*} = \frac{\Delta x}{x^*} + \frac{\Delta y}{y^*} = \delta x + \delta y$

### I.3.4 Cas de la division

1.  $\Delta\left(\frac{x}{y}\right) = \frac{x^* \Delta y + y^* \Delta x}{y^{*2}}$
2.  $\delta\left(\frac{x}{y}\right) = \delta x + \delta y$

#### I.3.4.1 Preuve

1. On a :  $x^* - \Delta x \leq x \leq x^* + \Delta x$

$$y^* - \Delta y \leq y \leq y^* + \Delta y \Rightarrow \frac{1}{y^* + \Delta y} \leq \frac{1}{y} \leq \frac{1}{y^* - \Delta y}$$

$$\text{D'où : } \frac{y^* - \Delta y}{y^* + \Delta y} \leq \frac{x}{y} \leq \frac{x^* + \Delta x}{y^* - \Delta y} \Leftrightarrow \frac{y^* - \Delta y}{y^* + \Delta y} \cdot \frac{y^* - \Delta y}{y^* - \Delta y} \leq \frac{x}{y} \leq \frac{y^* + \Delta y}{y^* - \Delta y} \cdot \frac{x^* + \Delta x}{y^* + \Delta y}$$



$$\Leftrightarrow \frac{x^*y^* - x^*\Delta y - y^*\Delta x + \Delta x\Delta y}{y^{*2} - \Delta^2 y} \leq \frac{x}{y} \leq \frac{x^*y^* + x^*\Delta y + y^*\Delta x + \Delta x\Delta y}{y^{*2} - \Delta^2 y}$$

Et en négligeant les erreurs du second ordre  $\Delta^2 y$  et  $\Delta x\Delta y$ , on obtient :

$$\frac{x^*y^* - x^*\Delta y - y^*\Delta x}{y^{*2}} \leq \frac{x}{y} \leq \frac{x^*y^* + x^*\Delta y + y^*\Delta x}{y^{*2}} \Leftrightarrow \frac{x^*}{y^*} - \frac{x^*\Delta y - y^*\Delta x}{y^{*2}} \leq \frac{x}{y} \leq \frac{x^*}{y^*} + \frac{x^*\Delta y - y^*\Delta x}{y^{*2}}$$

Ainsi :  $\frac{x^*\Delta y + y^*\Delta x}{y^{*2}}$  est un majorant de l'erreur absolue de  $\frac{x^*}{y^*}$  et donc :

$$\Delta\left(\frac{x}{y}\right) = \frac{x^*\Delta y + y^*\Delta x}{y^{*2}}$$

2. Pour l'erreur relative :

$$\delta\left(\frac{x}{y}\right) = \Delta\left(\frac{x}{y}\right) \cdot \frac{y^*}{x^*} = \frac{x^*\Delta y + y^*\Delta x}{y^{*2}} \cdot \frac{y^*}{x^*} = \delta x + \delta y$$

## I.4 Représentation décimale des nombres approchés

On sait que tout nombre réel positif  $x$  peut être représenté sous la forme d'un nombre décimal de développement limité ou illimité :

$$x = a_m 10^m + a_{m-1} 10^{m-1} + \dots + a_{m-n} 10^{m-n} + \dots$$

Où les  $a_i$  sont les chiffres du nombre réel  $x$ , avec  $a_m \neq 0$  où  $m$  est un entier naturel appelé *rang supérieur* du nombre réel  $x$ .

**Exemple :**  $\pi = 3.14159 \dots = 3 \cdot 10^0 + 1 \cdot 10^{-1} + 4 \cdot 10^{-2} + 1 \cdot 10^{-3} + 5 \cdot 10^{-4} + \dots$

**I.4.1 Remarque :** Dans la pratique, on n'utilise que des nombres approchés :

$$x \cong b_m 10^m + b_{m-1} 10^{m-1} + \dots + b_{m-n} 10^{m-n} + \dots \quad b_m \neq 0$$

- ✚ Tous les chiffres conservés  $b_i$  ( $i=m, \dots, m-n$ ) s'appellent chiffres significatifs du nombre approché  $x$ .
- ✚ Certains des  $b_i$  peuvent être nuls.
- ✚ Les exemples suivants illustrent les cas où le zéro n'est pas considéré comme chiffre significatif.
  - $x = 3 \cdot 10^{-3} + 0 \cdot 10^{-4} + 4 \cdot 10^{-5} + 0 \cdot 10^{-6}$  qui s'écrit en notation décimale  $x = \underline{0.003}040$ . Les zéros soulignés ne sont pas des chiffres significatifs.





- $x = 2 \cdot 10^8 + 0 \cdot 10^7 + 0 \cdot 10^6 + 1 \cdot 10^5 + 0 \cdot 10^4$  qui s'écrit en notation décimale  $x = 20010\underline{000}$ . Les zéros soulignés ne sont pas des chiffres significatifs.

## I.4.2 Définition

On appelle chiffre significatif d'un nombre approché, tout chiffre dans sa représentation décimale différent de zéro ; et un zéro s'il se trouve entre deux chiffres significatifs ou s'il constitue un chiffre réservé.

### I.4.2.1 Exemples

1. Une approximation à 6 décimales de 0.00301045 est :

0.003010 → Ce zéro traduit le fait que le nombre approché a conservé la décimale  $10^{-6}$  : c'est donc un chiffre significatif.  
 Etant placé entre deux chiffres significatifs (3 et 1), ce zéro est un chiffre significatif.  
 Ces zéros ne sont pas significatifs car ils ne servent qu'à indiquer le rang des autres chiffres.

2. Si on approxime le nombre 1897 à la centaine près par 1900, à la place des chiffres 9 et 4 négligés, on introduit des zéros. Ils ne servent qu'à fixer l'ordre de grandeur du nombre pris comme approximation. Ce ne sont pas des chiffres significatifs. Par contre si on approche le nombre 1899.7 à l'unité près par le nombre 1900, les deux zéros sont alors des chiffres significatifs.

## I.5 Chiffres exactes d'un nombre décimal approché

**I.5.1 Définition :** Un chiffre significatif d'un nombre approché  $x$  est dit exact (*c.s.e*) si l'erreur absolue de ce nombre *ne dépasse pas une demi unité* de rang du chiffre significatif.

Ainsi :

- Le  $n^{\text{ième}}$  chiffre significatif **après** la virgule est exacte si :  $\Delta x \leq 0.5 \cdot 10^{-n}$
- Le  $n^{\text{ième}}$  chiffre significatif **avant** la virgule est exacte si :  $\Delta x \leq 0.5 \cdot 10^{n-1}$



### I.5.1.1 Exemples

1. Pour  $x = 35.97$  et  $x^* = 36.00$ , nous avons :

$$\Delta x = |x - x^*| = |35.97 - 36.00| = 0.03 \leq 0.5 \cdot 10^{-1} \leq 0.5 \cdot 10^0 \leq 0.5 \cdot 10^1$$

Ainsi, les chiffres significatifs 3, 6 et le premier zéro après la virgule sont exacts.

2. Pour  $x = 2/3$  et  $x^* = 0.6671$ , nous avons :

$$\begin{aligned} \Delta x = |x - x^*| &= |2/3 - 0.6671| = \left| \frac{20000 - 20013}{3 \cdot 10^4} \right| = \frac{13}{3} 10^{-4} = 4.333 \dots 10^{-4} \\ &= 0.4333 \dots 10^{-3} \leq 0.5 \cdot 10^{-3} \leq 0.5 \cdot 10^{-2} \leq 0.5 \cdot 10^{-1} \end{aligned}$$

Ainsi, les chiffres significatifs 6, 6 et 7 sont exacts.

## I.5.2 Propriétés

- Si un chiffre significatif est exact, tous les chiffres à sa gauche sont exacts.
- Si un chiffre n'est pas exact, tous ceux à sa droite ne le sont pas.

### I.5.2.1 Remarque

Si l'erreur absolue ne dépasse pas une demi unité de rang du chiffre significatif, on dit que c'est une approximation au sens large ou encore que c'est une approximation à chiffres exacts dans un sens large.

**Exemple :** Pour  $x = 412.3567$  et  $x^* = 412.356$ .  $x^*$  est alors une approximation à 6 chiffres exacts dans un sens large puisque :  $\Delta x = |x - x^*| = 0.0007 = 0.7 \cdot 10^{-3} \leq 1 \cdot 10^{-3}$

## I.6 Troncature et arrondissement d'un nombre

L'arrondi est une méthode habituelle pour tronquer (couper en éliminant une partie) un nombre pour ne garder qu'un nombre fini de chiffres significatifs.

**Exemple :** Pour approximer le nombre  $\pi = 3.141492653589 \dots$ , on peut considérer la valeur approchée  $3.14$  ou encore  $3.14159$  et cela selon le besoin. Dans le premier cas on a tronqué  $\pi$  après 2 décimales. Dans le second cas on l'a tronqué après 5 décimales.

### I.6.1 Règle d'arrondissement



Pour arrondir un nombre jusqu'à  $n$  chiffres significatifs, il faut éliminer les chiffres à droite du  $n^{\text{ième}}$  chiffre significatif conservé si on se trouve après la virgule, sinon on remplace par des zéros :

- a. Si le  $(n+1)^{\text{ième}}$  chiffre significatif est  $>5$ , on augmente le  $n^{\text{ième}}$  chiffre de 1.
- b. Si le  $(n+1)^{\text{ième}}$  chiffre significatif est  $<5$ , les chiffres retenus restent inchangés.
- c. Si le  $(n+1)^{\text{ième}}$  chiffre significatif est 5 alors deux cas sont possibles :
  - i. Tous les chiffres rejetés, situés après le  $(n+1)^{\text{ième}}$  chiffre significatif sont des zéros : on applique la règle du chiffre pair. (le  $n^{\text{ième}}$  chiffre reste inchangé s'il est pair et on lui ajoute 1 s'il est impair).
  - ii. Parmi les chiffres rejetés, il existe au moins un qui soit non nul : on ajoute 1 au  $n^{\text{ième}}$  chiffre.

#### 1.6.1.1 Exemples

1. Arrondir  $x = 0.7897$  à trois chiffres significatifs.

$$7 > 5 \Rightarrow x \cong x^* \text{ où } x^* = 0.790$$

2. Arrondir  $x = 0.953$  à deux chiffres significatifs.

$$3 < 5 \Rightarrow x \cong x^* \text{ où } x^* = 0.95$$

3. Arrondir  $x = 1.6438\underline{5}00$  à cinq chiffres significatifs.

Tous les chiffres rejetés sont des zéro et le  $5^{\text{ième}}$  chiffre est pair :  $x \cong x^* = 1.6438$ .

4. Arrondir  $x = 2.64387\underline{5}00$  à six chiffres significatifs.

Tous les chiffres rejetés sont des zéro et le  $6^{\text{ième}}$  chiffre est impairs :

$$x \cong x^* = 2.64388.$$

5. Arrondir  $x = 1.6438\underline{5}001$  à cinq chiffres significatifs.

Parmi les chiffres rejetés il existe un chiffre différent de zéro. On ajoute donc 1 au

$$5^{\text{ième}} \text{ chiffre : } x \cong x^* = 1.6439.$$

**1.6.1.2 Remarque :** En appliquant la règle d'arrondissement, l'erreur absolue de la valeur approchée obtenue (erreur d'arrondi) ne dépasse pas une demi-unité de rang du dernier chiffre significatif retenu. Par conséquent, *un nombre correctement arrondi ne possède que des chiffres significatifs exacts.*



## I.7 Conclusion

- Si un nombre approché possède  $n$  chiffres significatifs exacts, alors son erreur relative est  $< 5 \cdot 10^{-n}$  (sauf si le nombre est 1 suivi de  $n-1$  zéros).
- Si l'erreur relative à  $x^*$  est  $\leq 0.5 \cdot 10^{-n}$  alors  $x^*$  possède au moins  $n$  chiffres significatifs exacts.





- 1- Méthodes directes : une méthode directe conduit à une solution en un nombre fini d'étapes, et s'il n'y a pas des erreurs d'arrondi, la solution serait celle du système.
- 2- Méthodes itératives : une méthode itérative fait passer d'un estimé  $x^{(k)}$  de la solution à un autre estimé  $x^{(k+1)}$  de cette solution.

En effet, il n'existe pas de règle définitive pour le choix entre méthodes directes ou indirectes. Cependant, les méthodes itératives sont rarement utilisées pour des systèmes à matrice pleine de faible dimension ( $n < 100$ ). Les méthodes itératives sont généralement préférées pour les systèmes de grande taille (l'accumulation des erreurs devient moins cruciale dans ce cas).

## II.2. Méthodes directes

### II.2.1 Méthodes d'élimination de Gauss

La méthode de Gauss engendre un algorithme fini exact dont l'idée est de transformer le système initial en un système triangulaire (inférieur ou supérieur).

#### II.2.1.1 Résolution d'un système triangulaire

On donne ici l'algorithme de résolution d'un système triangulaire inférieur. Il se fait en "remontant" les équations de la dernière ligne à la première.

$$\text{L'algorithme de résolution est : } \begin{cases} x_n = \frac{b_n}{a_{nn}} \\ x_i = \frac{1}{a_{ii}} (b_i - \sum_{j=i+1}^n a_{ij}x_j) \text{ pour } i = n-1, \dots, 1 \end{cases}$$

#### II.2.1.2 Méthode de Gauss

Soit à résoudre, le système suivant ( $n=3$ ) :

$$S1: \begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 & (E_1^{(1)}) \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2 & (E_2^{(1)}) \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 & (E_3^{(1)}) \end{cases}$$

On élimine le terme  $a_{21}x_1$  dans  $(E_2^{(1)})$  en prenant  $E_2^{(2)} = E_2^{(1)} - \frac{a_{21}}{a_{11}}E_1^{(1)}$



De même, on élimine le terme  $a_{31}x_1$  dans  $(E_3^{(1)})$  en prenant  $E_3^{(2)} = E_3^{(1)} - \frac{a_{31}}{a_{11}}E_1^{(1)}$

On obtient ainsi le système :

$$S2: \begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 & (E_1^{(1)}) \\ a_{22}^{(2)}x_2 + a_{32}^{(2)}x_3 = b_2^{(2)} & (E_2^{(2)}) \\ a_{23}^{(2)}x_2 + a_{33}^{(2)}x_3 = b_3^{(2)} & (E_3^{(2)}) \end{cases}$$

$$\text{Avec : } \begin{cases} a_{ij}^{(2)} = a_{ij} - \frac{a_{i1}}{a_{11}}a_{1j} \\ b_i^{(2)} = b_i - \frac{a_{i1}}{a_{11}}b_1 \end{cases} \quad i = 2, 3 \text{ et } j = 2, 3$$

En supposant que  $a_{22} \neq 0$ , on élimine de la même façon le terme  $a_{32}^{(2)}x_2$  grâce à la combinaison  $E_3^{(3)} = E_3^{(2)} - \frac{a_{32}^{(2)}}{a_{22}^{(2)}}E_2^{(2)}$  et on obtient un nouveau système équivalent au système initial :

$$S3: \begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 & (E_1^{(1)}) \\ a_{22}^{(2)}x_2 + a_{23}^{(2)}x_3 = b_2^{(2)} & (E_2^{(2)}) \\ a_{33}^{(3)}x_3 = b_3^{(3)} & (E_3^{(3)}) \end{cases}$$

$$\text{Avec : } \begin{cases} a_{33}^{(3)} = a_{33}^{(2)} - \frac{a_{32}^{(2)}}{a_{22}^{(2)}}a_{23}^{(2)} \\ b_3^{(3)} = b_3^{(2)} - \frac{a_{32}^{(2)}}{a_{22}^{(2)}}b_2^{(2)} \end{cases}$$

Le système S3 est triangulaire supérieur. La résolution se fait donc grâce à l'algorithme du paragraphe 3.1.1.

**Exemple** : soit à résoudre le système :

$$\begin{cases} 2x_1 + 3x_2 - x_3 = 5 \\ 4x_1 + 4x_2 - 3x_3 = 3 \\ -2x_1 + 3x_2 - x_3 = 1 \end{cases}$$



On écrit le système sous la forme  $Ax=b$  avec

$$A = \begin{pmatrix} 2 & 3 & -1 \\ 4 & 4 & -3 \\ -2 & 3 & -1 \end{pmatrix}; \quad b = \begin{pmatrix} 5 \\ 3 \\ 1 \end{pmatrix}; \quad x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

On a le pivot  $a_{11}^{(1)} \neq 0$ , on peut donc faire :

$$\left( \begin{array}{ccc|c} \langle 2 \rangle & 3 & -1 & 5 \\ 4 & 4 & -3 & 3 \\ -2 & 3 & -1 & 1 \end{array} \right) \longrightarrow \left( \begin{array}{ccc|c} 2 & 3 & -1 & 5 \\ 0 & -2 & -1 & -7 \\ 0 & 6 & -2 & 6 \end{array} \right)$$

$$[A^{(1)}|b^{(1)}] \quad \rightarrow \quad [A^{(2)}|b^{(2)}]$$

Le pivot  $a_{22}^{(2)} \neq 0$ , on peut donc faire :

$$\left( \begin{array}{ccc|c} 2 & 3 & -1 & 5 \\ 0 & \langle -2 \rangle & -1 & -7 \\ 0 & 6 & -2 & 6 \end{array} \right) \longrightarrow \left( \begin{array}{ccc|c} 2 & 3 & -1 & 5 \\ 0 & -2 & -1 & -7 \\ 0 & 0 & -5 & -15 \end{array} \right)$$

$$[A^{(2)}|b^{(2)}] \quad \rightarrow \quad [A^{(3)}|b^{(3)}]$$

En remplaçant dans le système d'équations, on a :

$$\begin{cases} 2x_1 + 3x_2 - x_3 = 5 \\ \quad -2x_2 - x_3 = -7 \\ \quad \quad -5x_3 = -15 \end{cases} \Leftrightarrow \begin{cases} x_1 = 1 \\ x_2 = 2 \\ x_3 = 3 \end{cases}$$

### II.2.1.3 Algorithme de Gauss

**1<sup>ère</sup> étape : Triangularisation** :  $[A, b] \rightarrow [U, b']$

**Pour k=1 à n-1 faire**

**Pour i=k+1 à n faire**

$$w = \frac{a_{ik}}{a_{kk}}$$

**Pour j=k+1 à n+1 faire**





$$a_{ij} = a_{ij} - w \cdot a_{kj}$$

**Fin pour**

**Fin pour**

**Fin pour**

**2<sup>ème</sup> étape : Résolution du système :  $Ux=b$ '**

**Pour  $i=n$  à 1 faire**

$$x_i = \frac{b_i - \sum_{j=i+1}^n a_{ij}x_j}{a_{ii}}$$

**Fin pour**

## II.3.2 Méthode de décomposition LU

### II.2.2.1 Introduction

La factorisation LU d'une matrice  $A_{n,n}$  est une astuce très importante dans le domaine de l'analyse numérique. Sa base est très simple, mais ses applications sont très nombreuses et très utiles.

### II.3.2.2 Principe

Décomposition de la matrice A de façon à la mettre sous la forme  $A=L.U$  où L est une matrice triangulaire unitaire inférieure et U est une matrice triangulaire supérieure.

Le système devient :

$$Ax = b \Leftrightarrow L U x = b \Leftrightarrow \begin{cases} Ly = b \\ Ux = y \end{cases}$$

La résolution du système  $Ax=b$  revient à résoudre les deux systèmes  $Ly=b$  et  $Ux=y$ . Puisque L et U sont triangulaires, la résolution est immédiate.



$$L = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ l_{21} & 1 & 0 & \cdots & 0 \\ l_{31} & l_{32} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \cdots & 1 \end{pmatrix} \quad U = \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1n-1} & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n-1} & u_{2n} \\ 0 & 0 & \cdots & u_{3n-1} & u_{3n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & u_{nn} \end{pmatrix}$$

La meilleure façon d'obtenir cette factorisation, est d'utiliser l'élimination de Gauss. Ainsi, la matrice  $U$  est la matrice finale après élimination de Gauss  $A^{(n)}$ .

$$U = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} \\ 0 & 0 & a_{33}^{(3)} & \cdots & a_{3n}^{(3)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{nn}^{(n)} \end{pmatrix}$$

Il reste à trouver la matrice  $L$  (triangulaire inférieure unitaire).

En effet, les différentes étapes de la triangularisation peuvent s'écrire sous forme matricielle. Si on pose  $A=A^{(0)}$ , la matrice  $A^{(1)}$  du second système est obtenue en multipliant à gauche par :

$$E^{(1)} = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{a_{21}}{a_{11}} & 1 & 0 \\ -\frac{a_{31}}{a_{11}} & 0 & 1 \end{pmatrix} \rightarrow A^{(1)} = E^{(1)}A^{(0)} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} \\ 0 & a_{32}^{(2)} & a_{33}^{(2)} \end{pmatrix}$$

De la même façon,  $A^{(2)}$  obtenue en multipliant  $A^{(1)}$  et  $E^{(2)}$  :

$$E^{(2)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{a_{32}^{(2)}}{a_{22}^{(2)}} & 1 \end{pmatrix} \rightarrow A^{(2)} = E^{(2)}A^{(1)} = E^{(2)}E^{(1)}A^{(0)} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} \\ 0 & 0 & a_{33}^{(3)} \end{pmatrix}$$

Ainsi,  $A^{(0)} = (E^{(2)}E^{(1)})^{-1}A^{(2)} = (E^{(1)})^{-1}(E^{(2)})^{-1}A^{(2)}$ . Les matrices  $E^{(i)}$  sont inversibles car leurs déterminants sont égaux à 1 et leurs inverses sont faciles à calculer. En effet, on



peut vérifier qu'on les obtient en transformant les termes sous la diagonale en leurs opposants.

On a donc :

$$L = (E^{(1)})^{-1}(E^{(2)})^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ \frac{a_{21}}{a_{11}} & 1 & 0 \\ \frac{a_{31}}{a_{11}} & \frac{a_{32}}{a_{22}} & 1 \\ \frac{a_{n1}}{a_{11}} & \frac{a_{n2}}{a_{22}} & 1 \end{pmatrix}$$

Dans le cas général, on a :  $L = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ l_{21} & 1 & 0 & \dots & 0 \\ l_{31} & l_{32} & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \dots & 1 \end{pmatrix}$  avec  $l_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}$

### II.2.2.3 Algorithme de factorisation LU

**Début**

$$u_{11} = a_{11}$$

**Pour j=2 à n faire**

$$u_{1j} = a_{1j}$$

$$l_{j1} = \frac{a_{j1}}{a_{11}}$$

**Fin pour**

**Pour i=2 à n-1 faire**

$$u_{ii} = a_{ii} - \sum_{k=1}^{i-1} l_{ik}u_{ki}$$

**Pour j=i+1 à n faire**

$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik}u_{kj}$$



$$l_{ji} = \frac{1}{u_{ii}} \left[ a_{ji} - \sum_{k=1}^{i-1} l_{jk} u_{ki} \right]$$

**Fin pour**

**Fin pour**

$$u_{nn} = a_{nn} - \sum_{k=1}^{n-1} l_{nk} u_{kn}$$

**Fin**

#### II. 2.2.4 Utilité de la factorisation LU

On remarque facilement que la méthode LU nécessite plus de calcul et d'espace mémoire que la méthode de résolution de Gauss. En effet, la méthode de Gauss est en quelque sorte une partie de la méthode LU (construction de la matrice U). Cependant, il ne faut pas douter de l'utilité de la méthode de factorisation LU. Dans plusieurs situations en analyse numérique, il y a plusieurs systèmes du type  $Ax_i = b_i$  à résoudre pour  $i = 1, \dots, M$  avec  $M$  très grand. Puisque la matrice  $A$  est constante pour tous ses systèmes, il est donc avantageux de calculer une fois  $L$  et  $U$  et résoudre les deux systèmes triangulaires à chaque fois, ce qui revient plus facile à faire.

### II.3.3 Méthode de Cholesky

#### II.2.3.1 Rappel

Une matrice  $A$  est symétrique si et seulement si :  $A = A^t$

Elle est définie positive si et seulement si :  $\forall x \in R^n \setminus \{0\} : x^t Ax > 0$

**Exemple :** soit la matrice  $\begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix}$

Cette matrice est symétrique.

On a :  $(x_1, x_2, x_3) \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = x_1^2 + x_2^2 + x_3^2 + (x_1 + x_2 + x_3)^2 > 0 \forall x \neq 0$



Ainsi, la matrice est définie positive.

**Remarques :** soit  $A \in \mathbb{R}^{n \times n}$  une matrice symétrique.

- $A$  est définie positive si et seulement si toutes ses valeurs propres sont positives.
- $A$  est définie positive si et seulement si toutes les sous-matrices :

$$\begin{pmatrix} a_{11} & \cdots & a_{1r} \\ \vdots & \ddots & \vdots \\ a_{r1} & \cdots & a_{rr} \end{pmatrix}; 1 \leq r \leq n$$

Ont un déterminant positif.

Il est intéressant de trouver une factorisation de  $A$  qui prend en compte sa symétrie.

### II.2.3.2 Théorème : (Cholesky)

Soit  $A$  une matrice carrée d'ordre  $n$  non singulière ( $\det(A) \neq 0$ ) et symétrique. Pour qu'il existe une matrice triangulaire inférieure  $L$ , de même dimension que  $A$ , telle que  $A = LL^t$ , il faut et il suffit que  $A$  soit définie positive.

**Remarque :**

- $L$  n'est pas unique.
  - La décomposition devient unique si l'on fixe à l'avance les éléments diagonaux  $l_{ii}$  avec  $l_{ii} > 0$ .
  - On remarque que cette décomposition est un peu similaire à la décomposition LU. Cependant, au lieu d'utiliser deux matrices, une suffit pour stocker les valeurs de  $L$ .
  - Un autre point important est que la matrice triangulaire inférieure produite par la factorisation LU est différente de celle produite par la factorisation de Cholesky
- Avant de donner l'algorithme général de factorisation de Cholesky, étudions un exemple.

**Exemple :** pour trouver la décomposition de Cholesky de la matrice :



$$A = \begin{pmatrix} 4 & 2 & -6 \\ 2 & 10 & 9 \\ -6 & 9 & 26 \end{pmatrix}$$

On doit résoudre l'équation : 
$$\begin{pmatrix} 4 & 2 & -6 \\ 2 & 10 & 9 \\ -6 & 9 & 26 \end{pmatrix} = \underbrace{\begin{pmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{pmatrix}}_L \underbrace{\begin{pmatrix} l_{11} & l_{21} & l_{31} \\ 0 & l_{22} & l_{32} \\ 0 & 0 & l_{33} \end{pmatrix}}_{L^t}$$

Pour ce faire, on écrit la matrice de multiplication pour trouver un système d'équations à 6 inconnus. L'astuce est de trouver les  $l_{ij}$  dans le bon ordre. On obtient les équations :

$$l_{11}^2 = 4 \quad \Rightarrow \quad l_{11} = 2$$

$$l_{21}l_{11} = 2 \quad \Rightarrow \quad l_{21} = 1$$

$$l_{31}l_{11} = -6 \quad \Rightarrow \quad l_{31} = -3$$

$$l_{21}^2 + l_{22}^2 = 10 \quad \Rightarrow \quad l_{22} = 3$$

$$l_{31}l_{21} + l_{32}l_{22} = 9 \quad \Rightarrow \quad l_{32} = 4$$

$$l_{31}^2 + l_{32}^2 + l_{33}^2 = 26 \quad \Rightarrow \quad l_{33} = 1$$

En conclusion, on a : 
$$L = \begin{pmatrix} 2 & 0 & 0 \\ 1 & 3 & 0 \\ -3 & 4 & 1 \end{pmatrix}$$

Pour une matrice d'ordre  $n \times n$ , on calcule suivant l'ordre :

$$\begin{pmatrix} 1 & & & & \\ 2 & 3 & & & \\ \vdots & 4 & 5 & & \\ \vdots & \vdots & \vdots & \ddots & \\ 2 & 4 & 6 & \cdots & 2n-1 \end{pmatrix}$$



Afin d'obtenir les éléments  $l_{ij}$  de la matrice  $L$  on multiplie les matrices  $L$  et  $L^t$ , puis on identifie les coefficients respectifs dans l'égalité :  $A = LL^t$  pour obtenir les équations :

$$\begin{cases} l_{11}^2 = a_{11} \\ l_{i1}^2 + l_{i2}^2 + \dots + l_{ii}^2 = a_{ii} \\ l_{i1}l_{ji} + l_{i2}l_{ji} + l_{i1}l_{ji} + \dots + l_{ii}l_{ji} = a_{ij}, i > j \\ l_{ij} = 0, i < j \end{cases}$$

On déduit ensuite :

$$\begin{cases} l_{11} = \sqrt{a_{11}} \\ l_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2}, \quad i = \overline{2, n} \\ l_{ji} = \frac{1}{l_{ii}} \left( a_{ij} - \sum_{k=1}^{i-1} l_{ik}l_{jk} \right), \quad i < j \\ l_{ji} = 0, i > j \end{cases}$$

### II.2.3.3 Algorithme de Cholesky

**Début**

**Pour k=1 à n faire**

$$l_{kk} = \sqrt{a_{kk} - \sum_{s=1}^{k-1} l_{ks}^2}$$

**Pour j=k+1 à n faire**

$$l_{jk} = \frac{1}{l_{kk}} \left( a_{kj} - \sum_{s=1}^{k-1} l_{js}l_{ks} \right)$$

**Fin pour**

**Fin pour**

**Fin**



### II.3. Calcul du déterminant

Les transformations élémentaires ne modifient pas la valeur du déterminant d'une matrice (des matrices semblables ont le même déterminant).

Le déterminant d'une matrice triangulaire est le produit de ses éléments diagonaux.

Après application de l'algorithme de Gauss du système  $Ax = b$ , on obtient un nouveau système équivalent  $Ux = b'$  et le déterminant de la matrice initiale A se calcule par :

$$\det(A) = \det(U) = \prod_{i=1}^n u_{ii}$$

La même formule est utilisée pour calculer le déterminant après une factorisation LU.

Pour la factorisation de Cholesky, on utilise la propriété :  $\det(AB) = \det(A) \cdot \det(B)$

$$\text{Ainsi : } \det(A) = \det(L) \cdot \det(L^t) = (\det(L))^2$$

### II.4 Conclusion

- Les méthodes directes sont utilisées surtout pour la résolution des petits ( $n < 20$ ) et moyens ( $n < 100$ ) systèmes d'équations linéaires.
- On dit que la meilleure méthode directe au sens du temps de calcul est celle qui demande le moins d'opérations.
- La méthode de Gauss ou sa variante LU est souvent préférée pour des systèmes à matrice quelconque.
- La méthode de Cholesky est recommandée pour résoudre des systèmes linéaires à matrice symétrique définie positive.





## Chapitre III : Méthodes itératives de résolution des systèmes d'équations linéaires

### III.1 Introduction

On se propose le problème de la résolution du système linéaire de Cramer  $Ax = b$ .

Où  $A$  est une matrice carrée de rang  $n$ .

Les méthodes que nous présentons ci-après sont une généralisation au cas  $n$ -dimensionnel des méthodes de résolution de  $f(x) = 0$  étudiées dans le chapitre précédent.

Ces méthodes consistent à utiliser un vecteur estimé initial  $X^{(0)} = (x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)})^t$  de la solution exacte  $X^* = (x_1^*, x_2^*, \dots, x_n^*)^t$  du système  $Ax = b$  et de générer une séquence de vecteurs :  $X^{(k+1)} = F^{(k)}(X^{(k)}, \dots, X^{(k)})$ .

### III.2 Principe

Soit à résoudre le système  $Ax = b$  où  $A$  est une matrice carrée d'ordre  $n$ .

Ecrivons  $A$  sous la forme  $A = M - N$ , le système devient donc :  $(M - N)x = b$  ou encore :

$$Mx = Nx + b$$

La méthode itérative associée à l'égalité précédente consiste, à partir d'un vecteur initial  $X^{(0)}$  à générer la suite  $X^{(1)}, X^{(2)}, \dots, X^{(k+1)}$  de la manière suivante :

$$X^{(1)} = M^{-1}NX^{(0)} + M^{-1}b$$

$$X^{(2)} = M^{-1}NX^{(1)} + M^{-1}b$$

.....



$$X^{(k+1)} = M^{-1}NX^{(k)} + M^{-1}b$$

Cette suite d'égalités peut être représentée par la relation itérative suivante :

$$X^{(k+1)} = TX^{(k)} + V \quad k = 0, 1, 2, \dots$$

Où :  $T = M^{-1}N$  et  $V = M^{-1}b$

La matrice d'itération  $T$  et le vecteur  $V$  sont indépendants de  $k$ .

La question que l'on se pose à propos de la méthode itérative de résolution de  $Ax = b$  par :

$X^{(k+1)} = TX^{(k)} + V$  est celle de la convergence. En d'autres termes  $X^{(k+1)}$  converge-t-il vers le vecteur solution de  $Ax = b$  ?

Pour répondre à cette question, définissons le vecteur d'erreur :

$$e^{(k)} = X^{(k)} - X^*$$

associé à la  $k^{\text{ième}}$  itération.

Puisque  $X^*$  vérifie :

$$X^* = TX^* + V$$

Et

$$X^{(k)} = TX^{(k-1)} + V$$

En soustrayant les deux équations précédentes, on obtient :

$$e^{(k)} = Te^{(k-1)}$$

En effet :  $e^{(1)} = Te^{(0)}$ ;  $e^{(2)} = Te^{(1)} = T^2e^{(0)}$ ;  $e^{(3)} = T^3e^{(0)}$  ...

Enfin :  $e^{(k)} = Te^{(k-1)} = T^ke^{(0)}$



La convergence de ce processus est assurée si, quelque soit  $e^{(0)}$  (donc quelque soit  $X^{(0)}$ ), on a :  $\lim_{k \rightarrow \infty} e^{(k)} = 0$  ou  $\lim_{k \rightarrow \infty} T^k e^{(0)} = 0$

Ce qui revient à dire que :  $\lim_{k \rightarrow \infty} T^k = 0$ .

### III.2.1 Théorème :

La suite définie par  $X^{(0)} \in R^n$  et  $X^{(k+1)} = TX^{(k)} + V$  converge vers  $X^* = (I - T)^{-1}V$  quelque soit  $X^{(0)}$  si et seulement si  $\rho(T) < 1$ .

Comme on a d'une manière générale :  $\rho(A) \leq \|A\|$ , une condition suffisante de convergence du système est telle que  $\|A\| < 1$ .

### III.2.2 Décomposition de la matrice A

Il faut choisir une matrice  $T$  qui assure la convergence et telle que  $TX + V$  ne soit pas très chère à calculer. Nous allons donc chercher des décompositions de la forme  $M - N$  de la matrice  $A$  de telle façon que  $M$  soit facilement inversible et vérifie  $\rho(M^{-1}N) < 1$  (ou  $\|M^{-1}N\| < 1$  : condition suffisante).

Définissons les matrices  $D, L, U$  telles que :

$D$  est une matrice diagonale :  $d_{ii} = a_{ii} \quad \forall i$

$L$  est une matrice inférieure :  $\begin{cases} l_{ij} = -a_{ij} ; i > j \\ l_{ij} = 0 \quad ; i \leq j \end{cases}$

$U$  est une matrice supérieure :  $\begin{cases} u_{ij} = -a_{ij} ; i < j \\ u_{ij} = 0 \quad ; i \geq j \end{cases}$

A partir de cela, on a la relation :  $A = D - L - U$

Qui correspond aux trois types de décompositions suivantes :



### Méthode de Jacobi

$$A = M - N \quad \text{où } M = D \text{ et } N = L + U$$

Dans ce cas la matrice T sera :  $T_j = D^{-1}(L + U)$

### Méthode de Gauss-Seidel

$$A = M - N \quad \text{où } M = D - L \text{ et } N = U$$

Dans ce cas la matrice T sera :  $T_{GS} = (D - L)^{-1}U$

### Méthode de relaxation

$$A = M - N \quad \text{où } M = \frac{D}{\omega} - L \text{ et } N = \frac{1 - \omega}{\omega}D + U$$

Dans ce cas la matrice T sera :  $T_\omega = \left(\frac{D}{\omega} - L\right)^{-1}\left(\frac{1-\omega}{\omega}D + U\right)$

## III.3 Méthode de Jacobi

### III.3.1 Principe

La matrice A étant décomposée en :  $A = M - N = D - (L + U)$

Le système  $Ax = b$  devient :  $DX = (L + U)X + b$

La méthode itérative de Jacobi s'écrit alors :

$$DX^{(k+1)} = (L + U)X^{(k)} + b$$

Soit :

$$X^{(k+1)} = D^{-1}(L + U)X^{(k)} + D^{-1}b$$



Qui peut s'écrire sous forme développée :

$$x_1^{(k+1)} = (b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)} - \dots - a_{1n}x_n^{(k)})/a_{11}$$

$$x_2^{(k+1)} = (b_2 - a_{21}x_1^{(k)} - a_{23}x_3^{(k)} - \dots - a_{2n}x_n^{(k)})/a_{22}$$

.....

$$x_n^{(k+1)} = (b_n - a_{n1}x_1^{(k)} - a_{n2}x_2^{(k)} - \dots - a_{nn-1}x_{n-1}^{(k)})/a_{nn}$$

Cette méthode suppose des pivots  $a_{ii}$  non nuls. Dans le cas où un pivot est nul, un simple échange des lignes suffit pour vérifier cette condition.

### III.3.2 Test d'arrêt

On note le vecteur résidu :  $r^{(k)} = b - AX^{(k)}$

Ou encore :  $r_i^{(k)} = b_i - \sum_{j=1}^n a_{ij}x_j^{(k)}$

Il existe plusieurs tests d'arrêt parmi lesquels on peut prendre :

- 1-  $\frac{\|r^{(k)}\|}{\|b\|} < \varepsilon$
- 2-  $\frac{\|X^{(k)} - X^{(k-1)}\|}{\|X^{(k)}\|} < \varepsilon$
- 3-  $\|X^{(k)} - X^{(k-1)}\| < \varepsilon$

### III.3.3 Algorithme de Jacobi

Données :  $b, A, X^{(0)}, \varepsilon$

**Début**

k=0

**Tant que**  $\frac{\|r^{(k)}\|}{\|b\|} \geq \varepsilon$  **faire**

**Pour** i=1 à n **faire**



$$r_i^{(k)} = b_i - \sum_{j=1}^n a_{ij}x_j^{(k)}$$

$$x_i^{(k+1)} = (x_i^{(k)} + r_i^{(k)})/a_{ii}$$

**Fin pour**

k=k+1

**Fin tant que**

**Fin**

### III.3.4 Conditions de convergence

L'itération de Jacobi s'écrit :

$$X^{(k+1)} = D^{-1}(L + U)X^{(k)} + D^{-1}b$$

Soit :

$$X^{(k+1)} = T_J X^{(k)} + V$$

Pour assurer la convergence, on doit avoir  $\rho(T_J) < 1$ , comme le calcul de  $\rho(T_J)$  est très souvent compliqué, on se contente de la condition suffisante, à savoir  $\|T_J\| < 1$ . Ce qui se traduit par :  $\sum_{j=1}^n |t_{ij}| < 1 \quad \forall i$

Puisqu'on a :  $T_J = D^{-1}(L + U)$ , on peut écrire :  $\sum_{j=1}^n |l_{ij} + u_{ij}| < |d_{ii}| \quad \forall i$

En revenant au système initial :

$$\sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}| < |a_{ii}| \quad \forall i = \overline{1, n}$$

#### III.3.4.1 Théorème

Une condition suffisante pour que  $T_J$  converge est que la matrice  $A$  du système  $Ax = b$  soit à diagonale fortement dominante.



*Note : Le processus itératif convergent jouit de la propriété importante d'autocorrection, qui fait qu'une erreur de calcul isolée n'entache pas le résultat final. Une approximation erronée peut être considérée comme un nouveau vecteur initial.*

### III.3.5 Exemple :

Résoudre le système suivant par la méthode itérative de Jacobi avec 3 décimales exactes.

$$\begin{cases} 4x_1 + 0.24x_2 - 0.08x_3 = 8 \\ 0.09x_1 + 3x_2 - 0.15x_3 = 9 \\ 0.04x_1 - 0.08x_2 + 4x_3 = 20 \end{cases}$$

#### III.3.5.1 Solution

Vérifions d'abord la convergence. La matrice est à diagonale fortement dominante puisque :

$$a_{11} = 4 > 0.24 + |-0.08| = 0.32$$

$$a_{22} = 3 > 0.09 + |-0.15| = 0.24$$

$$a_{33} = 4 > 0.04 + |-0.08| = 0.12$$

Ainsi, le processus itératif de Jacobi est convergent.

Mettons le système sous la forme réduite :

$$\begin{cases} x_1 = 2 - 0.06x_2 - 0.02x_3 \\ x_2 = 3 - 0.03x_1 + 0.05x_3 \\ x_3 = 5 - 0.01x_1 + 0.02x_2 \end{cases}$$

On peut choisir  $X^{(0)} = (2, 3, 5)^t$

En remplaçant ces valeurs dans le système réduit, on obtient :

$$\begin{cases} x_1^{(1)} = 2 - 0.06 \times 3 - 0.02 \times 5 = 1.92 \\ x_2^{(1)} = 3 - 0.03 \times 2 + 0.05 \times 5 = 3.19 \\ x_3^{(1)} = 5 - 0.01 \times 2 + 0.02 \times 3 = 5.04 \end{cases}$$

En appliquant une deuxième itération, on obtient :  $X^{(2)} = (1.9094, 3.1944, 5.0446)^t$



Pour la troisième itération, on trouve :  $X^{(3)} = (1.9092, 3.19495, 5.0448)^t$

En prenant  $e = 0.5 \times 10^{-3}$  :  $|x_1^{(3)} - x_1^{(2)}| = 0.0002 < 0.5 \times 10^{-3}$

$$|x_2^{(3)} - x_2^{(2)}| = 0.0005 = 0.5 \times 10^{-3}$$

$$|x_3^{(3)} - x_3^{(2)}| = 0.0002 < 0.5 \times 10^{-3}$$

La solution donnée avec trois décimales exactes est :

$$x_1 = 1.909 \mp 0.001, x_2 = 3.195 \mp 0.001, x_3 = 5.045 \mp 0.001$$

### III.4 Méthode de Gauss-Seidel

#### III.4.1 Principe

La matrice A étant décomposée en :  $A = M - N = (D - L) - U$

Dans la méthode itérative de Gauss-Seidel, on réécrit la formule d'itération de la manière suivante :

$$X^{(k+1)} = (D - L)^{-1}UX^{(k)} + (D - L)^{-1}b$$

Puisque l'inversion de la matrice (D-L) peut être difficile à calculer, on préfère écrire le système sous la forme :

$$(D - L)X^{(k+1)} = UX^{(k)} + b$$

Qui devient :  $DX^{(k+1)} = LX^{(k+1)} + UX^{(k)} + b$

Et finalement :  $X^{(k+1)} = D^{-1}LX^{(k+1)} + D^{-1}UX^{(k)} + D^{-1}b$

Le développement de cette récurrence vectorielle donne :

$$x_1^{(k+1)} = (b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)} - \dots - a_{1n}x_n^{(k)})/a_{11}$$





$$x_2^{(k+1)} = (b_2 - a_{21}x_1^{(k+1)} - a_{23}x_3^{(k)} - \dots - a_{2n}x_n^{(k)})/a_{22}$$

.....

$$x_n^{(k+1)} = (b_n - a_{n1}x_1^{(k+1)} - a_{n2}x_2^{(k+1)} - \dots - a_{nn-1}x_{n-1}^{(k+1)})/a_{nn}$$

On constate que la différence entre cette méthode et celle de Jacobi est l'utilisation immédiate des nouveaux estimés  $X^{(k+1)}$  à l'itération (k+1).

La condition d'avoir des pivots non nuls reste nécessaire aussi pour la méthode de Gauss-Seidel.

### III.4.2 Algorithme de Gauss-Seidel

Données :  $b, A, X^{(0)}, \varepsilon$

**Début**

k=0

**Tant que**  $|x_i^{(k+1)} - x_i^{(k)}| > \varepsilon$  **faire**

**Pour** i=1 à n **faire**

$$x_i^{(k+1)} = [b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)}]/a_{ii}$$

**Fin pour**

k=k+1

**Fin tant que**

**Fin**

### III.4.3 Convergence de la méthode de Gauss-Seidel

La matrice d'itération de Gauss-Seidel est définie par :  $T_{GS} = (D - L)^{-1}U$

Le vecteur V est défini par :  $V_{GS} = (D - L)^{-1}b$

Une condition suffisante pour la convergence de la méthode de Gauss-Seidel est que la matrice A soit une matrice à diagonale fortement dominante :



$$\sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}| < |a_{ii}| \quad \forall i = \overline{1, n}$$

### Remarques

- ❖ La méthode de Gauss-Seidel est plus rapide en convergence que la méthode de Jacobi. En effet, on utilise dans la même itération les valeurs déjà estimées. Dans un certain sens, le  $(k+1)$  ième itéré est plus proche de la limite pour la méthode de Gauss-Seidel que pour la méthode de Jacobi.
- ❖ La permutation des lignes dans le système initial peut facilement conduire à une divergence. Une deuxième vérification paraît nécessaire pour conclure de la convergence du système.
- ❖ Pour la méthode de Gauss-Seidel, un seul vecteur suffit pour enregistrer les valeurs des itérés successifs. Une fois un itéré de l'itération précédente est utilisé dans le calcul, il n'est pas nécessaire de le garder pour la prochaine itération. Ceci implique un gain dans l'espace de stockage.

#### III.4.4 Exemple

Résoudre le système suivant par la méthode itérative de Gauss-Seidel avec 4 décimales exactes.

$$\begin{cases} 10x_1 + x_2 + x_3 = 12 \\ 2x_1 + 10x_2 + x_3 = 13 \\ 2x_1 + 2x_2 + 10x_3 = 14 \end{cases}$$

En prenant :  $X^{(0)} = (1.2, 0, 0)^t$

##### III.4.4.1 Solution

Vérifions d'abord la convergence. La matrice est à diagonale fortement dominante puisque :

$$a_{11} = 10 > 1 + 1 = 2$$

$$a_{22} = 10 > 2 + 1 = 3$$



$$a_{33} = 10 > 2 + 2 = 4$$

Ainsi, le processus itératif de Gauss-Seidel est convergent.

Mettons le système sous la forme réduite :

$$\begin{cases} x_1 = 1.2 - 0.1x_2 - 0.1x_3 \\ x_2 = 1.3 - 0.2x_1 - 0.1x_3 \\ x_3 = 1.4 - 0.2x_1 - 0.2x_2 \end{cases}$$

L'approximation initiale est :  $X^{(0)} = (1.2, 0, 0)^t$

En remplaçant dans le processus itératif :

$$\begin{cases} x_1^{(1)} = 1.2 - 0.1 \times 0 - 0.1 \times 0 = 1.20 \\ x_2^{(1)} = 1.3 - 0.2 \times \underline{1.2} - 0.1 \times 0 = 1.06 \\ x_3^{(1)} = 1.4 - 0.2 \times \underline{1.2} - 0.2 \times \underline{1.06} = 0.948 \end{cases}$$

Pour la deuxième itération, on obtient :

$$\begin{cases} x_1^{(2)} = 1.2 - 0.1 \times 0.06 - 0.1 \times 0.948 = 0.9992 \\ x_2^{(2)} = 1.3 - 0.2 \times \underline{0.9992} - 0.1 \times 0.948 = 1.00536 \\ x_3^{(2)} = 1.4 - 0.2 \times \underline{0.9992} - 0.2 \times \underline{1.00536} = 0.999098 \end{cases}$$

Pour les autres itérations, on a :

$$X^{(3)} = (0.9996, 1.0002, 1.0001)^t$$

$$X^{(4)} = (1.0000, 1.0000, 1.0000)^t$$

$$X^{(5)} = (1.0000, 1.0000, 1.0000)^t$$

Ainsi, la solution approchée du système est :

$$x_1 = 1.0000 \mp 0.0001, x_2 = 1.0000 \mp 0.0001 \text{ et } x_3 = 1.0000 \mp 0.0001$$

**Remarque :** la solution exacte est :  $(1, 1, 1)^t$

### III.5 Méthode de Relaxation

#### III.5.1. Introduction

On modifie légèrement les méthodes précédentes en introduisant un paramètre  $w$ , le coefficient de relaxation. Ce paramètre est généralement constant.

La relaxation sur la méthode de Jacobi n'apporte en général aucun gain appréciable. Appliqué à la méthode de Gauss-Seidel, elle permet d'améliorer la rapidité de la convergence.

#### III.5.2. Description

L'idée est que si la « correction » apportée à une composante va « dans le bon sens », on a intérêt à l'augmenter en la multipliant par un facteur supérieur à 1 ( $\omega > 1$  : sur-relaxation).

Au contraire si on risque de diverger ou osciller, il vaut mieux amortir la correction en la multipliant par un facteur inférieur à 1 ( $\omega < 1$  : sous-relaxation)

Une condition nécessaire mais non suffisante de convergence de ces méthodes est que le paramètre  $w$  soit compris entre 0 et 2.

Pour cette méthode, on prend la décomposition suivante :

$$P = \frac{D}{w} - E \quad \text{et} \quad N = \frac{1-w}{w}D + F$$

En remplaçant dans l'équation  $x^{(k+1)} = P^{-1}Nx^{(k)} + P^{-1}b$ , on obtient

$$x^{(k+1)} = \left(\frac{D}{w} - E\right)^{-1} \left(\frac{1-w}{w}D + F\right)x^{(k)} + \left(\frac{D}{w} - E\right)^{-1}b$$

La matrice d'itération est donnée comme suit :



$$B = \left( \frac{D}{\omega} - E \right)^{-1} \left( \frac{1-\omega}{\omega} D + F \right)$$

L'idée de la méthode de sur-relaxation (SOR = Successive Over Relaxation) est d'utiliser la méthode de Gauss Seidel pour calculer un itéré intermédiaire  $\tilde{x}^{(k+1)}$  qu'on "relaxe" ensuite pour améliorer la vitesse de convergence de la méthode.

On se donne  $0 < \omega < 2$ , et on modifie l'algorithme de Gauss–Seidel de la manière suivante :

$$\begin{cases} x_0 \in \mathbb{R}^n \\ a_{i,i} \tilde{x}_i^{(k+1)} = - \sum_{j < i} a_{i,j} x_j^{(k+1)} - \sum_{i < j} a_{i,j} x_j^{(k)} + b_i \\ x_i^{(k+1)} = \omega \tilde{x}_i^{(k+1)} + (1 - \omega) x_i^{(k)}, \quad i = 1, \dots, n. \end{cases}$$

(Pour  $\omega = 1$  on retrouve la méthode de Gauss–Seidel.)

L'algorithme ci-dessus peut aussi s'écrire (en multipliant par  $a_{i,i}$  la ligne 3 de l'algorithme

$$\begin{cases} x^{(0)} \in \mathbb{R}^n \\ a_{i,i} x_i^{(k+1)} = \omega \left[ - \sum_{j < i} a_{i,j} x_j^{(k+1)} - \sum_{j > i} a_{i,j} x_j^{(k)} + b_i \right] \\ \quad + (1 - \omega) a_{i,i} x_i^{(k)}. \end{cases}$$

### III.5.3. Méthodes SOR et SSOR et Jacobi

$$x_i^{(k+1)} = \frac{\omega}{a_{ii}} \left( b_i - \sum_{\substack{i=1 \\ j \neq i}}^n a_{ij} x_j^{(k)} \right) + (1 - \omega) x_i^{(k)}, \quad i = 1, \dots, n,$$

*Exemple :*

Soit le système de 3 équations à 3 inconnues



$$\begin{cases} 3x_1 + x_2 - x_3 = 2 \\ x_1 + 5x_2 + 2x_3 = 17 \\ 2x_1 - x_2 - 6x_3 = -18 \end{cases}$$

Suivant la formule décrite précédemment, la méthode de relaxation s'écrit dans ce cas pour la première itération à partir d'un vecteur  $x^{(0)} = (0, 0, 0)^T$ .

$$x^{(1)} = \left(\frac{D}{w} - E\right)^{-1} \left(\frac{1-w}{w} D + F\right) x^{(0)} + \left(\frac{D}{w} - E\right)^{-1} b$$

$$A = \begin{bmatrix} 3 & 1 & -1 \\ 1 & 5 & 2 \\ 2 & -1 & -6 \end{bmatrix} \quad D = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & -6 \end{bmatrix} \quad -E = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 2 & -1 & 0 \end{bmatrix} \quad -F = \begin{bmatrix} 0 & 1 & -1 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \end{bmatrix}$$

$$D/w = \begin{bmatrix} 3/1,1 & 0 & 0 \\ 0 & 5/1,1 & 0 \\ 0 & 0 & -6/1,1 \end{bmatrix} = \begin{bmatrix} 2,73 & 0 & 0 \\ 0 & 4,55 & 0 \\ 0 & 0 & -5,45 \end{bmatrix}$$

$$(D/w) - E = \begin{bmatrix} 2,73 & 0 & 0 \\ 1 & 4,55 & 0 \\ 2 & -1 & -5,45 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 0 \\ -2 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 2,73 & 0 & 0 \\ 2 & 4,55 & 0 \\ 4 & -2 & -5,45 \end{bmatrix}$$

$$(-0,09) * D = \begin{bmatrix} -0,27 & 0 & 0 \\ 0 & -0,45 & 0 \\ 0 & 0 & 0,54 \end{bmatrix} + \begin{bmatrix} 0 & -1 & 1 \\ 0 & 0 & -2 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} -0,27 & -1 & +1 \\ 0 & -0,45 & -2 \\ 0 & 0 & 0,54 \end{bmatrix} = \left(\frac{1-w}{w} D + F\right)$$



$$\left(\frac{D}{w} - E\right)^{-1} = \begin{bmatrix} 2,73 & 0 & 0 \\ 2 & 4,55 & 0 \\ 4 & -2 & -5,45 \end{bmatrix}^{-1}$$

Qui est l'inverse de la matrice

En utilisant le principe de la méthode d'élimination de Gauss, on calcule l'inverse de la matrice de la manière suivante :  $[A|Id] \rightarrow [Id|A]$

$$\left[ \begin{array}{ccc|ccc} 2,73 & 0 & 0 & 1 & 0 & 0 \\ 2 & 4,55 & 0 & 0 & 1 & 0 \\ 4 & -2 & -5,45 & 0 & 0 & 1 \end{array} \right] \begin{array}{l} L_2 = L_2 - (2/2,73)L_1 \\ L_3 = L_3 - (4/2,73)L_1 \end{array}$$

$$\left[ \begin{array}{ccc|ccc} 2,73 & 0 & 0 & 1 & 0 & 0 \\ 0 & 4,55 & 0 & -0,73 & 1 & 0 \\ 0 & -2 & -5,45 & -1,46 & 0 & 1 \end{array} \right] L_3 = L_3 - (-2/4,55)L_2$$

$$\left[ \begin{array}{ccc|ccc} 2,73 & 0 & 0 & 1 & 0 & 0 \\ 0 & 4,55 & 0 & -0,73 & 1 & 0 \\ 0 & 0 & -5,45 & -1,8 & 0,44 & 1 \end{array} \right] \quad \left[ \begin{array}{ccc|ccc} 1 & 0 & 0 & 0,37 & 0 & 0 \\ 0 & 1 & 0 & -0,16 & 0,22 & 0 \\ 0 & 0 & 1 & 0,33 & -0,08 & -0,18 \end{array} \right]$$

$$\left(\frac{D}{w} - E\right)^{-1} = \begin{bmatrix} 0,37 & 0 & 0 \\ -0,16 & 0,22 & 0 \\ 0,33 & -0,08 & -0,18 \end{bmatrix}$$

$$\left(\frac{D}{w} - E\right)^{-1} b = \begin{bmatrix} 0,37 & 0 & 0 \\ -0,16 & 0,22 & 0 \\ 0,33 & -0,08 & -0,18 \end{bmatrix} \begin{bmatrix} 2 \\ 17 \\ -18 \end{bmatrix} = \begin{bmatrix} 0,74 \\ 3,42 \\ 2,6 \end{bmatrix}$$



$$x^{(1)} = \left(\frac{D}{w} - E\right)^{-1} \left(\frac{1-w}{w} D + F\right) x^{(0)} + \left(\frac{D}{w} - E\right)^{-1} b$$

$$\begin{cases} x_1^{k+1} = -0,099x_1^k - 0,37x_2^k + 0,37x_3^k + 0,74 \\ x_2^{k+1} = 0,043x_1^k - 0,061x_2^k - 0,6x_3^k + 3,57 \\ x_3^{k+1} = -0,089x_1^k - 0,294x_2^k + 0,39x_3^k + 2,9 \end{cases}$$

0	0	0
0.7333	3.5787	2.9128
0.4158	2.0090	2.7929
0.9792	2.0948	2.9957
0.9657	2.0000	2.9879
0.9990	2.0056	2.9998
0.9980	2.0000	2.9993
1.0000	2.0003	3.0000
0.9999	2.0000	3.0000
1.0000	2.0000	3.0000
1.0000	2.0000	3.0000

Prenons comme vecteur initial  $x^{(0)}=(0,0,0)^T$

$$\begin{cases} x_1^{(1)} = -0,099(0) - 0,37(0) + 0,37(0) + 0,74 \\ x_2^{(1)} = 0,043(0) - 0,061(0) - 0,6(0) + 3,57 \\ x_3^{(1)} = -0,089(0) - 0,294(0) + 0,39(0) + 2,9 \end{cases}$$

$$\begin{cases} x_1^{(1)} = -0,099(0) - 0,37(0) + 0,37(0) + 0,74 \\ x_2^{(1)} = 0,043(0) - 0,061(0) - 0,6(0) + 3,57 \\ x_3^{(1)} = -0,089(0) - 0,294(0) + 0,39(0) + 2,9 \end{cases}$$

0.9980	2.0000	2.9993
1.0000	2.0003	3.0000
0.9999	2.0000	3.0000
1.0000	2.0000	3.0000
1.0000	2.0000	3.0000

Résultats des trois méthodes Jacobi; Gauss Seidel et Relaxation

0	0	0	0	0	0	0	0	0
0.6667	3.4000	3.0000	0.6667	3.2667	2.6778	0.7333	3.5787	2.9128
0.5333	2.0667	2.6556	0.4704	2.2348	2.7843	0.4158	2.0090	2.7929
0.8630	2.2311	2.8333	0.8498	2.1163	2.9306	0.9792	2.0948	2.9957
0.8674	2.0941	2.9158	0.9381	2.0402	2.9727	0.9657	2.0000	2.9879
0.9406	2.0602	2.9401	0.9775	2.0154	2.9899	0.9990	2.0056	2.9998
0.9600	2.0358	2.9702	0.9915	2.0057	2.9962	0.9980	2.0000	2.9993
0.9781	2.0199	2.9807	0.9968	2.0021	2.9986	1.0000	2.0003	3.0000
0.9869	2.0121	2.9894	0.9988	2.0008	2.9995	0.9999	2.0000	3.0000
0.9924	2.0069	2.9936	0.9996	2.0003	2.9998	1.0000	2.0000	3.0000
0.9956	2.0041	2.9963	0.9998	2.0001	2.9999	1.0000	2.0000	3.0000

On remarque que la méthode de Gauss-Seidel converge plus rapidement que Jacobi, la relaxation permet de réduire le nombre d'itérations avec un coefficient de relaxation  $w=1,1$

III.5.4 Résolution Matricielle

-1

**B = D (E + F): Méthode de Jacobi**





-1

**$B = (D - E) F$  : Méthode de Gauss-Seidel**

**$B = \left(\frac{D}{w} - E\right)^{-1} \left(\frac{1-w}{w} D + F\right)$  : Méthode de Relaxation**

$D$  : diagonale de  $A$ .

$E$  : triangulaire inférieure avec des 0 sur la diagonale.

$F$  : triangulaire supérieure avec des 0 sur la diagonale.

### III.6 Convergence des méthodes itératives

#### III.6.1. Conditions Suffisantes

1. Si  $A$  est à diagonale dominante alors les méthodes de Jacobi et de Gauss-Seidel convergent
2. Si  $A$  est strictement symétrique définie positive alors la méthode de Gauss-Seidel converge et la méthode de la relaxation converge pour  $0 < \omega < 2$ .
3. Si  $\|A\|_1 = \max_{j=1\dots n} \sum_{i=1}^n |a_{ij}| < 1$
4. Si  $\|A\|_\infty = \max_{i=1\dots n} \sum_{j=1}^n |a_{ij}| < 1$

#### III.6.2. Condition nécessaire & Suffisantes

La méthode itérative de forme  $x^{(k+1)} = Bx^{(k)} + C$ , converge si et seulement si

$$\rho(B) < 1$$

Une matrice carrée  $A$  est à diagonale dominante si

$$\forall i \in [1, n], |a_{ii}| > \sum_{j=1, j \neq i}^n |a_{ij}|$$

$$\begin{pmatrix} 7 & 2 & 0 \\ 3 & 5 & -1 \\ 0 & 5 & -6 \end{pmatrix} \quad \begin{array}{l} |a_{11}| > |a_{12}| + |a_{13}| \quad |7| > |2| + |0| \\ |a_{22}| > |a_{21}| + |a_{23}| \quad |5| > |3| + |-1| \\ |a_{33}| > |a_{31}| + |a_{32}| \quad |-6| > |0| + |5| \end{array}$$

C'est donc une matrice à diagonale dominante.



$$\begin{bmatrix} 20 & 12 & 25 \\ 12 & -18 & 2 \\ 1 & 2 & 5 \end{bmatrix} \quad \begin{array}{l} |a_{11}| < |a_{12}| + |a_{13}| \quad |20| < |12| + |25| \\ |a_{22}| > |a_{21}| + |a_{23}| \quad |-18| > |12| + |2| \\ |a_{33}| > |a_{31}| + |a_{32}| \quad |5| > |1| + |2| \end{array}$$

Ce n'est donc pas une matrice à diagonale dominante.

Une matrice A est symétrique si  $A^T = A$

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad A^T = \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \\ a_{13} & a_{23} & a_{33} \end{bmatrix}$$

$$A = \begin{bmatrix} 20 & 12 & 5 \\ 12 & 15 & 2 \\ 5 & 2 & 25 \end{bmatrix} \quad A^T = \begin{bmatrix} 20 & 12 & 5 \\ 12 & 15 & 2 \\ 5 & 2 & 25 \end{bmatrix}$$

Mineur principal d'une matrice carrée :

Soit  $A \in \mathbb{R}^{n \times n}$  est une matrice. Les mineurs principaux d'ordre k de cette matrice sont les déterminants des matrices tronquées  $(a_{ij})_{1 \leq i, j \leq k}$  pour k allant de 1 à n .

Exemple

$$A = \begin{bmatrix} 2 & 0 & 1 \\ 0 & -1 & 1 \\ 1 & 0 & -2 \end{bmatrix}$$

Les mineurs principaux de la matrice A sont

$\Delta_i, i=1; 2; 3$ , tels que :

$$\Delta_1 = 2$$

$$\Delta_2 = \begin{vmatrix} 2 & 0 \\ 0 & -1 \end{vmatrix} = (2) * (-1) - (0) * (0) = -2$$

$$\Delta_3 = \begin{vmatrix} 2 & 0 & 1 \\ 0 & -1 & 1 \\ 1 & 0 & -2 \end{vmatrix} = 2 \begin{vmatrix} -1 & 1 \\ 0 & -2 \end{vmatrix} + 1 \begin{vmatrix} 0 & -1 \\ 1 & 0 \end{vmatrix} = 4 + 1 = 5$$

La matrice symétrique  $A \in M_{n \times n}$  est définie positive si et seulement si tous ses mineurs principaux sont strictement positifs ( $\forall i=1, n \Delta_i > 0$ )

$$A = \begin{bmatrix} 20 & 12 & 5 \\ 12 & 15 & 2 \\ 5 & 2 & 25 \end{bmatrix} \quad A^T = \begin{bmatrix} 20 & 12 & 5 \\ 12 & 15 & 2 \\ 5 & 2 & 25 \end{bmatrix}$$

1)  $\Delta_1 = 20 > 0$ :

2)  $\Delta_2 = \begin{vmatrix} 20 & 12 \\ 12 & 15 \end{vmatrix} = (20) * (15) - (12) * (12) = 156 > 0$

3)  $\Delta_3 = \begin{vmatrix} 20 & 12 & 5 \\ 12 & 15 & 2 \\ 5 & 2 & 25 \end{vmatrix} = 20 \begin{vmatrix} 15 & 2 \\ 2 & 25 \end{vmatrix} - 12 \begin{vmatrix} 12 & 2 \\ 5 & 25 \end{vmatrix} + 5 \begin{vmatrix} 12 & 15 \\ 5 & 2 \end{vmatrix}$   
 $= 7420 - 3480 - 255 = 3685 > 0$



Les normes matricielles de  $\|A\|$  en termes des éléments de  $A$  que l'on note  $a_{ij}$

$$\|A\|_1 = \max_{j=1\dots n} \sum_{i=1}^n |a_{ij}|$$

$$\|A\|_\infty = \max_{i=1\dots n} \sum_{j=1}^n |a_{ij}|$$

$$A = \begin{bmatrix} 20 & 12 & 5 \\ 12 & 15 & 2 \\ 5 & 2 & 25 \end{bmatrix}$$

$$B = \begin{bmatrix} 2 & 0 & 1 \\ 0 & -1 & 1 \\ 1 & 0 & -2 \end{bmatrix}$$

$$\|A\|_1 = \max_{j=1\dots n} \sum_{i=1}^n |a_{ij}| = \max[|20| + |12| + |5|, |12| + |15| + |2|, |5| + |2| + |25|] \\ = \max\{37, 29, 32\} = 37$$

$$\|B\|_\infty = \max_{i=1\dots n} \sum_{j=1}^n |a_{ij}| = \max[|2| + |0| + |1|, |0| + |-1| + |1|, |1| + |0| + |-2|] \\ = \max\{3, 2, 3\} = 3$$

Le **rayon spectral** d'une matrice est le maximum des modules des valeurs propres

$$\rho(A) = \max|\lambda_i|$$

Pour le calcul des valeurs propres  $\lambda_i$ , on calcule le déterminant de  $A - \lambda I$ .

Exemple

$$A = \begin{bmatrix} 0 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 0 \end{bmatrix} \quad I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \lambda I = \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix}$$

$$A - \lambda I = \begin{bmatrix} 0 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 0 \end{bmatrix} - \lambda I = \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix} = \begin{bmatrix} -\lambda & -1 & 1 \\ -1 & 1 - \lambda & -1 \\ 1 & -1 & -\lambda \end{bmatrix}$$

$$\det(A - \lambda I) = \begin{vmatrix} -\lambda & -1 & 1 \\ -1 & 1 - \lambda & -1 \\ 1 & -1 & -\lambda \end{vmatrix} = -\lambda \begin{vmatrix} 1 - \lambda & -1 \\ -1 & -\lambda \end{vmatrix} - (-1) \begin{vmatrix} -1 & -1 \\ 1 & -\lambda \end{vmatrix} + 1 \begin{vmatrix} -1 & 1 - \lambda \\ 1 & -1 \end{vmatrix} \\ = -\lambda((- \lambda)(1 - \lambda) - 1) + (\lambda + 1) + (1 - 1 + \lambda) \\ = -\lambda(-\lambda + \lambda^2 - 1) + 2\lambda + 1 \\ = \lambda^2 - \lambda^3 + \lambda + 2\lambda + 1 = -\lambda^3 + \lambda^2 + 3\lambda + 1 \\ = -\lambda^3 + \lambda^2 + 3\lambda + 1 \\ = -(\lambda + 1)(\lambda^2 - 2\lambda - 1)$$



$$\det(A - \lambda I) = -(\lambda + 1) \left( \lambda - (1 + \sqrt{2}) \right) \left( \lambda - (1 - \sqrt{2}) \right)$$

Les valeurs propres de A sont :

$$\lambda_1 = -1, \quad \lambda_2 = 1 + \sqrt{2}, \quad \lambda_3 = 1 - \sqrt{2}$$

$$\rho(A) = \max |\lambda_i| = 1 + \sqrt{2}$$

### Conditions Suffisantes

1. Si A est à **diagonale dominante** alors les méthodes de **Jacobi et de Gauss-Seidel convergent**
2. Si A est **strictement symétrique définie positive** alors la méthode de **Gauss-Seidel converge et la méthode de la relaxation converge pour  $0 < \omega < 2$** .

$$3. \|A\|_1 = \max_{j=1 \dots n} \sum_{i=1}^n |a_{ij}| < 1$$

$$4. \|A\|_\infty = \max_{i=1 \dots n} \sum_{j=1}^n |a_{ij}| < 1$$

### Condition nécessaire & Suffisantes

La méthode itérative de forme  $x^{(k+1)} = Bx^{(k)} + C$ , converge si et seulement si

$$\rho(B) < 1$$

### Etude de Convergence

#### Exemple 1

Calculer les approximations successives de la solution du système par les méthodes de Jacobi et de Gauss-Seidel. Dans quels cas ces méthodes convergent-elles vers la solution du problème ?

$$A = \begin{pmatrix} a & b \\ b & a \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$\begin{cases} ax + by = 1 \\ bx + ay = 1 \end{cases} \quad \begin{cases} x^{(k+1)} = \frac{1}{a} (1 - by^{(k)}) \\ y^{(k+1)} = \frac{1}{b} (1 - ax^{(k)}) \end{cases}$$



Commençons par les conditions suffisantes,

*A est à diagonale dominante si*  $\forall i \in [1, n], |a_{ii}| > \sum_{j=1, j \neq i}^n |a_{ij}|$

*A est à diagonale si*  $|a| > |b|$

*alors les méthodes de Jacobi et de Gauss Seidel convergent si*

$$|a| > |b|$$

A est strictement symétrique définie positive

$$A = \begin{pmatrix} a & b \\ b & a \end{pmatrix} A^T = \begin{pmatrix} a & b \\ b & a \end{pmatrix}$$

Donc  $A=A^T$  ce qui veut dire que A est Symétrique

Donc  $A=A^T$  ce **qui** veut dire que A est Symétrique

$A \in M_{n \times n}$  est définie positive si et seulement si tous ses mineurs principaux

Sont strictement positifs (  $i=1, n \Delta_i > 0$  )

$$\text{Det}(A) = \begin{vmatrix} a & b \\ b & a \end{vmatrix} = a^2 - b^2 \text{Det}(A) > 0 \text{ si } a^2 > b^2 \quad |a| > |b|$$

La méthode de Gauss Seidel converge si  $|a| > |b|$

Condition nécessaire & Suffisantes (**Méthode de Jacobi**)

**Matrice de Jacobi:  $J=D^{-1}(E+F)$**

$$D = \begin{pmatrix} a & 0 \\ 0 & a \end{pmatrix} -E = \begin{pmatrix} 0 & 0 \\ b & 0 \end{pmatrix} -F = \begin{pmatrix} 0 & b \\ 0 & 0 \end{pmatrix}$$

$$D^{-1} = \begin{pmatrix} 1/a & 0 \\ 0 & 1/a \end{pmatrix}$$

$$E+F = \begin{pmatrix} 0 & -b \\ -b & 0 \end{pmatrix}$$

$$\text{Donc } D^{-1}(E+F) = \begin{pmatrix} 1/a & 0 \\ 0 & 1/a \end{pmatrix} * \begin{pmatrix} 0 & -b \\ -b & 0 \end{pmatrix} = \begin{pmatrix} 0 & -b/a \\ -b/a & 0 \end{pmatrix}$$

$$=A^T \text{ ce qui veut } J = \begin{pmatrix} 0 & -b/a \\ -b/a & 0 \end{pmatrix} \text{ ire que } A$$

Condition nécessaire & Suffisantes (**Méthode de Jacobi**)

$$=A^T J = \begin{pmatrix} 0 & -b/a \\ -b/a & 0 \end{pmatrix}^i$$

Pour que la méthode de Jacobi converge si et seulement si  $\rho(j) < 1$  Calculons  $\text{det}(j-$

$$\text{II}) \Rightarrow \lambda = \frac{b}{a} \text{ ou } \lambda = -\frac{b}{a}$$

$$\rho(j) = \max\{|\lambda_i|\} = \frac{|b|}{|a|}$$

La méthode de Jacobi converge si  $\rho(j) < 1 \Rightarrow |b| < |a|$



La solution successive par la méthode de Gauss Seidel du système

$$A = \begin{pmatrix} a & b \\ b & a \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \begin{cases} x^{(k+1)} = \frac{1}{a} (1 - by^{(k)}) \\ y^{(k+1)} = \frac{1}{b} (1 - ax^{(k+1)}) \end{cases}$$

Pour que la méthode de Gauss Seidel converge si et seulement si  $\rho(G) < 1$

G: est la matrice de Gauss-Seidel,  $G = (D-E)^{-1}F$

G: est la matrice de Gauss-Seidel,  $G = (D-E)^{-1}F$

$$G = \left( \begin{pmatrix} a & 0 \\ 0 & a \end{pmatrix} - \begin{pmatrix} 0 & 0 \\ -b & 0 \end{pmatrix} \right)^{-1} * \begin{pmatrix} 0 & -b \\ 0 & 0 \end{pmatrix}$$

$$G = \begin{pmatrix} a & 0 \\ b & a \end{pmatrix}^{-1} * \begin{pmatrix} 0 & -b \\ 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} a & 0 \\ b & a \end{pmatrix}^{-1} = ?$$

$$\left( \begin{array}{cc|cc} a & 0 & 1 & 0 \\ b & a & 0 & 1 \end{array} \right) \begin{array}{l} l_2 = l_2 - \frac{b}{a}l_1 \end{array} \mapsto \left( \begin{array}{cc|cc} a & 0 & 1 & 0 \\ 0 & a & -b/a & 1 \end{array} \right) \mapsto \left( \begin{array}{cc|cc} 1 & 0 & 1/a & 0 \\ 0 & 1 & -b/a^2 & 1/a \end{array} \right)$$

$$\begin{pmatrix} a & 0 \\ b & a \end{pmatrix}^{-1} = \begin{pmatrix} 1/a & 0 \\ -b/a^2 & 1/a \end{pmatrix}$$

$$G = \begin{pmatrix} a & 0 \\ b & a \end{pmatrix}^{-1} * \begin{pmatrix} 0 & -b \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 1/a & 0 \\ -b/a^2 & 1/a \end{pmatrix} * \begin{pmatrix} 0 & -b \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & -b/a \\ 0 & b^2/a^2 \end{pmatrix}$$

$$G = \begin{pmatrix} 0 & -b/a \\ 0 & b^2/a^2 \end{pmatrix}$$

$$\det(G - \lambda I) = \det \left( \begin{pmatrix} 0 & -b/a \\ 0 & b^2/a^2 \end{pmatrix} - \lambda \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right) = \begin{vmatrix} -\lambda & -b/a \\ 0 & b^2/a^2 - \lambda \end{vmatrix} \\ = -\lambda \left( \frac{b^2}{a^2} - \lambda \right)$$

$$\det(G - \lambda I) = 0 \Rightarrow -\lambda \left( \frac{b^2}{a^2} - \lambda \right) = 0 \Rightarrow \lambda = 0 \vee \lambda = \frac{b^2}{a^2}$$

$$\rho(G) = \max\{|\lambda_i|\} = \max\left\{ \left| 0 \right|, \left| \frac{b^2}{a^2} \right| \right\} = \frac{|b^2|}{|a^2|}$$

La méthode de Gauss Seidel converge si  $\rho(G) < 1 \Rightarrow |b^2| < |a^2|$   
 $\Rightarrow |b| < |a|$

### Exemple 2



1. On considère le système  $Ax = b$ , où:  $A = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$

La matrice de relaxation est donnée par :

$$\mathcal{L}_w = \left(\frac{D}{w} - E\right)^{-1} \left(\frac{1-w}{w} D + F\right)$$

Où  $D$ ,  $-E$  et  $-F$  sont respectivement les matrices Diagonale, Triangulaire inférieure et Triangulaire supérieure avec  $A = D - E - F$ .

1- Ecrire la matrice  $L_w$  de la méthode de relaxation.

2- déterminer une condition nécessaire et suffisante sur le paramètre  $w$  pour la convergence de la méthode de relaxation.

a) Ecrire la matrice  $L_w$  de la méthode de relaxation.

$$D = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad -E = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad -F = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

La matrice de relaxation est donnée par :

$$D/w = \begin{bmatrix} 1/\omega & 0 & 0 \\ 0 & 1/\omega & 0 \\ 0 & 0 & 1/\omega \end{bmatrix}$$

$$\mathcal{L}_w = \left(\frac{D}{w} - E\right)^{-1} \left(\frac{1-w}{w} D + F\right)$$

$$(D/w) - E = \begin{bmatrix} 1/\omega & 0 & 0 \\ 0 & 1/\omega & 0 \\ 0 & 0 & 1/\omega \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1/\omega & 0 & 0 \\ 1 & 1/\omega & 0 \\ 0 & 0 & 1/\omega \end{bmatrix}$$

$$\left(\frac{D}{w} - E\right)^{-1} = \begin{bmatrix} 1/\omega & 0 & 0 \\ 1 & 1/\omega & 0 \\ 0 & 0 & 1/\omega \end{bmatrix}^{-1} \quad \text{Qui est l'inverse de la matrice } \left(\frac{D}{w} + E\right)$$

En utilisant le principe de la méthode d'élimination de Gauss, on calcul l'inverse de la matrice de la manière suivante  $[A|I_d] \longrightarrow [I_d|A]$

$$\left[ \begin{array}{ccc|ccc} 1/\omega & 0 & 0 & 1 & 0 & 0 \\ 1 & 1/\omega & 0 & 0 & 1 & 0 \\ 0 & 0 & 1/\omega & 0 & 0 & 1 \end{array} \right] \quad L_2 = L_2 - (\omega)L_1 \quad \left[ \begin{array}{ccc|ccc} 1/\omega & 0 & 0 & 1 & 0 & 0 \\ 0 & 1/\omega & 0 & -\omega & 1 & 0 \\ 0 & 0 & 1/\omega & 0 & 0 & 1 \end{array} \right]$$



$$\left[ \begin{array}{ccc|ccc} 1 & 0 & 0 & \omega & 0 & 0 \\ 0 & 1 & 0 & -\omega^2 & \omega & 0 \\ 0 & 0 & 1 & 0 & 0 & \omega \end{array} \right] \quad \left( \frac{D}{w} - E \right)^{-1} = \begin{bmatrix} \omega & 0 & 0 \\ -\omega^2 & \omega & 0 \\ 0 & 0 & \omega \end{bmatrix}$$

$$\left( \frac{1-w}{w} D + F \right) = \begin{bmatrix} (1-\omega)/\omega & 0 & 0 \\ 0 & (1-\omega)/\omega & 0 \\ 0 & 0 & (1-\omega)/\omega \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} (1-\omega)/\omega & 0 & 0 \\ 0 & (1-\omega)/\omega & -1 \\ 0 & 0 & (1-\omega)/\omega \end{bmatrix}$$

$$\mathcal{L}_w = \left( \frac{D}{w} - E \right)^{-1} \left( \frac{1-w}{w} D + F \right) = \begin{bmatrix} \omega & 0 & 0 \\ -\omega^2 & \omega & 0 \\ 0 & 0 & \omega \end{bmatrix} * \begin{bmatrix} (1-\omega)/\omega & 0 & 0 \\ 0 & (1-\omega)/\omega & -1 \\ 0 & 0 & (1-\omega)/\omega \end{bmatrix}$$

$$= \begin{bmatrix} (1-\omega) & 0 & 0 \\ -\omega(1-\omega) & (1-\omega) & -\omega \\ 0 & 0 & 1-\omega \end{bmatrix}$$

b) déterminer une condition nécessaire et suffisante sur le paramètre  $\omega$  pour la convergence de la méthode de relaxation.

$$\det(\mathcal{L}_w - \lambda I) = \det \left( \begin{pmatrix} (1-\omega) & 0 & 0 \\ -\omega(1-\omega) & (1-\omega) & -\omega \\ 0 & 0 & 1-\omega \end{pmatrix} - \begin{pmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{pmatrix} \right)$$

$$= \det \begin{pmatrix} (1-\omega) - \lambda & 0 & 0 \\ -\omega(1-\omega) & (1-\omega) - \lambda & -\omega \\ 0 & 0 & (1-\omega) - \lambda \end{pmatrix}$$

$$= ((1-\omega) - \lambda) * \det \begin{pmatrix} (1-\omega) - \lambda & 0 \\ 0 & (1-\omega) - \lambda \end{pmatrix}$$

$$\det(\mathcal{L}_w - \lambda I) = ((1-\omega) - \lambda) * ((1-\omega) - \lambda)^2 = ((1-\omega) - \lambda)^3$$

$$\det(\mathcal{L}_w - \lambda I) = 0 \quad \Rightarrow \quad ((1-\omega) - \lambda)^3 = 0$$

Les valeurs propres de  $\mathcal{L}_w$  sont  $\lambda_1 = \lambda_2 = \lambda_3 = 1 - \omega$

donc  $\rho(\mathcal{L}_w) = |1 - \omega|$





*on a alors les équivalences suivantes :*

*La procédure de relaxation est convergente :*

$$\begin{aligned}\rho(\mathcal{L}_\omega) < 1 &\Leftrightarrow |1 - \omega| < 1 \\ &\Leftrightarrow 0 < \omega < 2\end{aligned}$$

### **III.7 Conclusion**

Dans la méthode de Jacobi (ou JOR), chaque composante de l'approximation de la solution peut être calculée indépendamment des autres. Cette méthode est donc facilement parallélisable. Au contraire, pour la méthode de Gauss-Seidel (ou SOR), ce calcul ne peut se faire que séquentiellement, mais sans qu'on ait toutefois besoin de stocker l'approximation de la solution à l'étape précédente, d'où un gain de mémoire.



## Chapitre IV : Calcul de valeurs et de vecteurs propres

### IV.1 Introduction

Nous abordons dans ce chapitre le problème du calcul de valeurs propres (et, éventuellement, de vecteurs propres d'une matrice d'ordre  $n$ . C'est un problème beaucoup plus difficile que celui de la résolution d'un système linéaire. En effet, les valeurs propres d'une matrice étant les racines de son polynôme caractéristique, on pourrait naïvement penser qu'il suffit de factoriser ce dernier pour les obtenir.

### IV.2 Localisation des valeurs propres

#### IV.2.1. Définition :

Si  $A$  est une matrice de  $n \times n$ , alors un vecteur non nul  $x$  est dit vecteur propre de  $A$  si

$$Ax = \lambda x$$

$\lambda$  est appelé valeur propre de  $A$ , et  $x$  vecteur propre de  $A$  correspondant à  $\lambda$ .

#### Exemple :

$$A = \begin{bmatrix} 3 & 0 \\ 8 & -1 \end{bmatrix} \quad x = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

- Si  $\lambda > 1$  Dilatation.
- Si  $0 < \lambda < 1$  Contraction.
- Si  $\lambda < 0$  Changement de direction.



#### IV.2.2. Calcul analytique des valeurs propres et des vecteurs propres

$$Ax = \lambda x \Leftrightarrow (\lambda I - A)x = 0$$

Nous avons une solution non nulle ssi  $\det(A - \lambda I) = 0$  Équation caractéristique de A.

Les valeurs satisfaisant cette équation sont les **valeurs propres** de A.

*Exemple 1 :*

$$A = \begin{bmatrix} 3 & 2 \\ -1 & 0 \end{bmatrix}$$

$$\lambda I - A = \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 3 & 2 \\ -1 & 0 \end{bmatrix}$$

$$\det(\lambda I - A) = \det \begin{bmatrix} \lambda - 3 & -2 \\ 1 & \lambda \end{bmatrix}$$

L'équation caractéristique de A est  $\lambda^2 - 3\lambda + 2 = 0$ .

*Exemple 2 :*

Trouver les valeurs et les vecteurs propres de :

$$A = \begin{bmatrix} 10 & 0 & 0 \\ 1 & -3 & -7 \\ 0 & 2 & 6 \end{bmatrix}$$

$$\det \begin{bmatrix} 10 - \lambda & 0 & 0 \\ 1 & -3 - \lambda & -7 \\ 0 & 2 & 6 - \lambda \end{bmatrix}$$

Les valeurs propres sont :  $\lambda_1=10$ ,  $\lambda_2=4$  et  $\lambda_3=-1$



$$\lambda_1=10$$

$$\begin{bmatrix} 10 & 0 & 0 \\ 1 & -3 & -7 \\ 0 & 2 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 10 \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad \begin{cases} 0x_1 & & = 0 \\ x_1 & -13x_2 & -7x_3 = 0 \\ & 2x_2 & -4x_3 = 0 \end{cases} \quad \begin{cases} x_1 = 1 \\ x_2 = \frac{2}{33} \\ x_3 = \frac{1}{33} \end{cases}$$

$$\lambda_2=4$$

$$\begin{bmatrix} 10 & 0 & 0 \\ 1 & -3 & -7 \\ 0 & 2 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 4 \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad \begin{cases} 6x_1 & & = 0 \\ x_1 & -7x_2 & -7x_3 = 0 \\ & 2x_2 & +2x_3 = 0 \end{cases} \quad \begin{cases} x_1 = 0 \\ x_2 = -1 \\ x_3 = 1 \end{cases}$$

$$\lambda_3=-1$$

$$\begin{bmatrix} 10 & 0 & 0 \\ 1 & -3 & -7 \\ 0 & 2 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = -1 \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad \begin{cases} 11x_1 & & = 0 \\ x_1 & -2x_2 & -7x_3 = 0 \\ & 2x_2 & +7x_3 = 0 \end{cases} \quad \begin{cases} x_1 = 0 \\ x_2 = +1 \\ x_3 = -\frac{2}{7} \end{cases}$$

### IV.2.3. La localisation des valeurs propres :

#### IV.2.3.1. Théorème de Gerschgorin

$$Ax = \lambda x \rightarrow$$

$$\sum_{j=1}^n a_{ij} x_j = \lambda x_i$$

→ pour  $i = 1$  à  $n$ .

$$\sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} x_j = (\lambda - a_{ii}) x_i$$

Si nous choisissons  $x_i$  tel que

$$|x_i| = \max_j |x_j|$$

$$|\lambda - a_{ii}| = \left| \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} \frac{x_j}{x_i} \right|$$



$$|\lambda - a_{ii}| \leq \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}| \left| \frac{x_j}{x_i} \right| \leq \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}|$$

Posons

$$r_i = \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}|$$

C'est à dire la sommation des valeurs absolues des éléments de la ligne  $i$  sauf  $a_{ii}$ .

#### IV.2.3.2. La localisation des valeurs propres

Les disques de Guerschgorin sont définis par :

$$D_i = \left\{ \lambda \mid |\lambda - a_{ii}| \leq r_i \right\} \quad i = 1 \text{ à } n.$$

*Théorème :*

Les valeurs propres de la matrice  $A$  sont éléments de l'union des disques  $D_i$ .

$$S = \bigcup_{i=1}^n D_i$$

*Note :*

1. On vérifie toutes les possibilités pour  $i = 1$  à  $n$  qui pourraient vérifier  $|x_i| = \max_j |x_j|$  pour chaque valeur (vecteur) propres.
2. Puisque les valeurs propres peuvent être complexes, nous obtenons des disques dans le plan complexe.
3. Si la matrice  $A$  est symétrique, les valeurs propres sont réelles et les disques deviennent de simples intervalles.
4.  $a_{ii}$  : centre du disque.  $r_i$  : rayon du disque.

*Exemple :*



$$A = \begin{bmatrix} 4 & -1 & 1 \\ 1 & 1 & 1 \\ -2 & 0 & -6 \end{bmatrix}$$

$$D_1 = \left\{ \lambda \mid |\lambda - a_{11}| \leq r_1 \right\}, \quad r_1 = \sum_{j=2}^3 |a_{1j}|.$$

$$|\lambda - 4| \leq 2$$

$\lambda$  Peut-être complexe :

$$\lambda = \lambda_R + i\lambda_I$$

$$\left| (\lambda_R - 4) + i(\lambda_I - 0) \right| \leq 2$$

$$\sqrt{(\lambda_R - 4)^2 + (\lambda_I)^2} \leq 2$$

$$(\lambda_R - 4)^2 + (\lambda_I)^2 \leq 4$$

$D_1$  : centre = (4,0), rayon = 2

$D_2$  : centre = (1,0), rayon = 2

$D_3$  : centre = (-6,0), rayon = 2

*Corollaires :*

1. Les valeurs propres de la matrice A sont aussi éléments de l'union des disques  $D_i^T$  construit à partir de sa transposée  $A^T$  :

$$S^T = \bigcup_{i=1}^n D_i^T$$

Note : En effet, A et  $A^T$  ont les mêmes valeurs propres.

2. Les valeurs propres de la matrice A appartiennent donc à l'intersection de S et  $S^T$  :

$$\lambda \in (S \cap S^T)$$

3. Une borne supérieure pour la plus grande valeur propre est donc :



$$|\lambda| \leq \min \left\{ \max_{i=1..n} \left[ \sum_{j=1}^n |a_{ij}| \right], \max_{j=1..n} \left[ \sum_{i=1}^n |a_{ij}| \right] \right\}$$

*Exemple (suite) :*

Pouvons-nous améliorer le résultat ?

$$A^T = \begin{bmatrix} 4 & 1 & -2 \\ -1 & 1 & 0 \\ 1 & 1 & -6 \end{bmatrix}$$

$D_1^T$  : centre = (4,0), rayon = 3

$D_2^T$  : centre = (1,0), rayon = 1

$D_3^T$  : centre = (-6,0), rayon = 2

$$\lambda \in (S \cap S^T)$$

$$\lambda_1 = -5.7685$$

$$\lambda_2 = 3.4694$$

$$\lambda_3 = 1.2992$$

### IV.3 Méthode de la puissance

Supposons que nous cherchions la plus grande valeur propre de la matrice A (n x n) et que cette matrice possède n vecteurs linéairement indépendants  $x_1, x_2, \dots, x_n$  associés respectivement aux valeurs propres  $\lambda_1, \lambda_2, \dots, \lambda_n$  qui sont dans l'ordre.

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n| > 0$$

La valeur propre  $\lambda_1$  est dite **dominante**.

Tout vecteur x de  $\mathbb{R}^n$  peut donc s'écrire :

$$x = \sum_{i=1}^n c_i x_i$$



Puisque les vecteurs propres de  $A$  forment une base de  $\mathbb{R}^n$ . En multipliant ce vecteur par  $A$ , nous obtenons :

$$x^{(1)} = Ax^{(0)} = \sum_{i=1}^n c_i Ax_i = \sum_{i=1}^n c_i \lambda_i x_i$$

Si nous répétons cette opération  $k$  fois :

$$\begin{aligned} x^{(k)} &= A^k x^{(0)} = \sum_{i=1}^n c_i (\lambda_i)^k x_i \\ &= \lambda_1^k \left[ c_1 x_1 + c_2 \left( \frac{\lambda_2}{\lambda_1} \right)^k x_2 + \dots + c_n \left( \frac{\lambda_n}{\lambda_1} \right)^k x_n \right] \end{aligned}$$

Or nous avons  $\left| \frac{\lambda_i}{\lambda_1} \right| < 1$  pour  $i \neq 1$

Donc le terme  $\lambda_1^k c_1 x_1$  devient dominant dès que  $k$  est assez grand. Donc pour  $k$  assez grand, nous avons :

$$x^{(k)} \cong \lambda_1^k c_1 x_1$$

$$x^{(k+1)} \cong \lambda_1^{k+1} c_1 x_1 = \lambda_1 (\lambda_1^k c_1 x_1) \cong \lambda_1 x^{(k)}$$

$$x^{(k+1)} = Ax^{(k)} \cong \lambda_1 x^{(k)}$$

En choisissant 2 composantes  $i$ , nous obtenons :

$$\lambda_1 \cong \frac{x_i^{(k+1)}}{x_i^{(k)}}$$

- Nous obtenons donc la plus grande valeur propre et son vecteur propre en même temps.
- Dépassement de la capacité de l'ordinateur ?

Pour éviter de dépasser les capacités de l'ordinateur, nous avons intérêt à normaliser les vecteur itérés  $x_k$  à chaque étape ; nous pouvons par exemple ramener à 1 la plus grande composante (en valeur absolue) de  $x_k$ , en appliquant l'algorithme :





$$\begin{array}{ll}
 x^{(1)} = Ax^{(0)} & y^1 = \frac{1}{m_1} x^{(1)} \\
 x^{(2)} = Ay^{(1)} & y^2 = \frac{1}{m_2} x^{(2)} \\
 \vdots & \vdots \\
 x^{(k)} = Ay^{(k-1)} & y^k = \frac{1}{m_k} x^{(k)}
 \end{array}$$

Où  $m_k$  est la composante de  $x_k$  de module maximum. Dans ces conditions, nous avons finalement, pour  $k$  assez grand :

$$y^k \cong y^{k+1} = \frac{1}{m_{k+1}} x^{(k+1)} = \frac{1}{m_{k+1}} Ay^k$$

Donc

$$m_{k+1} y^k \cong Ay^k$$

Remarque :

$$\lim_{k \rightarrow \infty} m_k = \lambda_1 \qquad \lim_{k \rightarrow \infty} y^k = \vec{v}_1$$



Exemple :

$$A = \begin{bmatrix} 10 & 0 & 0 \\ 1 & -3 & -7 \\ 0 & 2 & 6 \end{bmatrix} \quad x^{(0)} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 10 & 0 & 0 \\ 1 & -3 & -7 \\ 0 & 2 & 6 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 10 \\ 1 \\ 0 \end{bmatrix} = 10 \begin{bmatrix} 1 \\ 0.1 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 10 & 0 & 0 \\ 1 & -3 & -7 \\ 0 & 2 & 6 \end{bmatrix} \begin{bmatrix} 1 \\ 0.1 \\ 0 \end{bmatrix} = \begin{bmatrix} 10 \\ 0.7 \\ 0.2 \end{bmatrix} = 10 \begin{bmatrix} 1 \\ 0.07 \\ 0.02 \end{bmatrix}$$

$$\begin{bmatrix} 10 & 0 & 0 \\ 1 & -3 & -7 \\ 0 & 2 & 6 \end{bmatrix} \begin{bmatrix} 1 \\ 0.07 \\ 0.02 \end{bmatrix} = \begin{bmatrix} 10 \\ 0.65 \\ 0.26 \end{bmatrix} = 10 \begin{bmatrix} 1 \\ 0.065 \\ 0.026 \end{bmatrix}$$

$$\begin{bmatrix} 10 & 0 & 0 \\ 1 & -3 & -7 \\ 0 & 2 & 6 \end{bmatrix} \begin{bmatrix} 1 \\ 0.065 \\ 0.026 \end{bmatrix} = \begin{bmatrix} 10 \\ 0.623 \\ 0.286 \end{bmatrix} = 10 \begin{bmatrix} 1 \\ 0.0623 \\ 0.0286 \end{bmatrix}$$

$$\begin{bmatrix} 10 & 0 & 0 \\ 1 & -3 & -7 \\ 0 & 2 & 6 \end{bmatrix} \begin{bmatrix} 1 \\ 0.0623 \\ 0.0286 \end{bmatrix} = \begin{bmatrix} 10 \\ 0.6129 \\ 0.2962 \end{bmatrix} = 10 \begin{bmatrix} 1 \\ 0.06129 \\ 0.02962 \end{bmatrix}$$

$$\lambda_1 = 10 \quad \vec{x}_1 = \begin{bmatrix} 1 \\ 2/33 \\ 1/33 \end{bmatrix}$$

Remarques :

1. Nous avons :



$$Ax = \lambda x$$

$$A^{-1}Ax = A^{-1}\lambda x = \lambda A^{-1}x$$

$$x = \lambda A^{-1}x$$

$$A^{-1}x = \frac{1}{\lambda}x$$

Pour obtenir la plus petite valeur propre, nous pouvons donc utiliser la méthode des puissances sur  $a^{-1}$ .

2. Si  $|\lambda_1| \approx |\lambda_2|$  (...  $|\lambda_{n-1}| \approx |\lambda_n|$ ), alors la convergence sera très lente car  $\left|\frac{\lambda_2}{\lambda_1}\right| \ll 1$  doit être vrai pour avoir une convergence rapide.
3. Le choix du vecteur initial influence beaucoup la rapidité de la convergence. Si le vecteur initial est près du vecteur propre, alors les coefficients  $c_i$ ,  $i \neq 1$ , seront petit par rapport à  $c_1$ .

$$x^{(k)} = A^k x^{(0)} = c_1 \lambda_1^k x_1 + c_2 \lambda_2^k x_2 + \dots + c_n \lambda_n^k x_n$$



*Exemple :*

$$A = \begin{bmatrix} 10 & 0 & 0 \\ 1 & -3 & -7 \\ 0 & 2 & 6 \end{bmatrix} \quad A^{-1} = \begin{bmatrix} 0.1 & 0 & 0 \\ 0.15 & -1.5 & -1.75 \\ -0.05 & 0.5 & 0.75 \end{bmatrix} \quad x^{(0)} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$A^{-1} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.1 \\ -3.1 \\ 1.2 \end{bmatrix} = -3.1 \begin{bmatrix} -0.0323 \\ 1 \\ -0.3871 \end{bmatrix}$$

$$A^{-1}y^1 = \begin{bmatrix} -0.0032 \\ -0.8274 \\ 0.2113 \end{bmatrix} = -0.8274 \begin{bmatrix} 0.0039 \\ 1 \\ -0.2554 \end{bmatrix}$$

$$A^{-1}y^2 = \begin{bmatrix} 0.004 \\ -1.0525 \\ 0.3083 \end{bmatrix} = -1.0525 \begin{bmatrix} -0.0004 \\ 1 \\ -0.2929 \end{bmatrix}$$

$$A^{-1}y^3 = \begin{bmatrix} 0.0000 \\ -0.9875 \\ 0.2803 \end{bmatrix} = -0.9875 \begin{bmatrix} 0 \\ 1 \\ -0.2839 \end{bmatrix}$$

Valeurs exactes :

$$\lambda_3 = -1 \quad \bar{x}_3 = \begin{bmatrix} 0 \\ 1 \\ -0.2857 \end{bmatrix}$$

### IV.3 Conclusion

La méthode de la puissance fournit une très bonne approximation des valeurs propres extrémales d'une matrice et de vecteurs propres associés. Dans la suite, on note  $\lambda_1$  et  $\lambda_n$  les valeurs propres d'une matrice A d'ordre ayant respectivement le plus petit et le plus grand module.

## Chapitre V : Analyse matricielle

### V.1 Introduction

On rappelle dans ce chapitre un certain nombre de résultats relatifs aux espaces vectoriels de dimension finie et aux matrices, qui sont utilisés dans l'ensemble du cours. La plupart des notions abordées sont supposées déjà connues du lecteur, à l'exception peut-être des normes matricielles.

### V.2 Espaces vectoriels

Nous commençons par rappeler la notion d'espace vectoriel.

Un espace vectoriel sur  $\mathbb{K}$  est un ensemble non vide  $E$  sur lequel est définie une loi interne notée « + », appelée addition, et une loi externe notée « . » , appelée multiplication par un scalaire, possédant les propriétés suivantes :

1.  $(E, +)$  est un groupe commutatif (ou abélien),
2.  $\forall (\lambda, \mu) \in \mathbb{K}^2$  et  $\forall v \in E$ ,  $(\lambda + \mu)v = \lambda v + \mu v$ ,
3.  $\forall (\lambda, \mu) \in \mathbb{K}^2$  et  $\forall v \in E$ ,  $\lambda(\mu v) = (\lambda\mu)v$ ,
4.  $\forall \lambda \in \mathbb{K}$  et  $\forall (v, w) \in E^2$ ,  $\lambda(v + w) = \lambda v + \lambda w$ ,
5.  $\forall v \in E$ ,  $1_{\mathbb{K}}v = v$ ,

Le scalaire  $1_{\mathbb{K}}$  étant l'élément unitaire du corps  $\mathbb{K}$ . Les éléments de l'espace vectoriel  $E$  sont appelés **vecteurs**.

On dit qu'une partie non vide  $F$  d'un espace vectoriel  $E$  est un sous-espace vectoriel de  $E$  si et seulement si

$$\forall (v, w) \in F^2, \forall (\lambda, \mu) \in \mathbb{K}^2, \lambda v + \mu w \in F.$$

En particulier, l'ensemble des combinaisons linéaires d'une famille  $\{v_i\}_{i=1, \dots, p}$  de  $p$  vecteurs de  $E$  est un **sous-espace vectoriel** de  $E$ , appelé sous-espace engendré par la famille de vecteurs. On le note

$$\text{Vect}\{v_1, \dots, v_p\} = \{v = \lambda_1 v_1 + \dots + \lambda_p v_p, \text{ avec } \lambda_i \in \mathbb{K}, i = 1, \dots, p\}.$$



La famille  $\{v_i\}_{i=1,\dots,p}$  est alors appelée famille génératrice de ce sous-espace.

Un espace vectoriel sur  $\mathbb{K}$  est dit de **dimension finie** s'il admet une famille génératrice de cardinal fini. Sinon, il est dit de **dimension infinie**.

Dans toute la suite, nous ne considérons que des espaces vectoriels de dimension finie.

Une famille de vecteurs  $\{v_i\}_{i=1,\dots,p}$  d'un espace vectoriel  $E$  est dite **libre** si les vecteurs

$v_1, \dots, v_p$  sont linéairement indépendants, c'est-à-dire si la relation

$$\lambda_1 v_1 + \dots + \lambda_p v_p = \mathbf{0},$$

Où  $\mathbf{0}$  est l'élément nul de  $E$  et  $\lambda_i \in \mathbb{K}$ ,  $i = 1, \dots, p$ , implique que  $\lambda_1 = \dots = \lambda_p = 0$ .

Dans le cas contraire, la famille est dite **liée**.

On appelle **base** de l'espace vectoriel  $E$  toute famille libre et génératrice de  $E$ . Si la famille  $\{e_i\}_{i=1,\dots,n}$  est une base de  $E$ , tout vecteur de  $E$  admet une décomposition unique de la forme

$$v = \sum_{i=1}^n v_i e_i, \quad \forall v \in E,$$

Les scalaires  $v_i$ ,  $i = 1, \dots, n$ , étant appelés les composantes du vecteur  $v$  dans la base

$\{e_i\}_{i=1,\dots,n}$ .

Le cardinal d'une base quelconque d'un espace vectoriel  $E$  de dimension finie s'appelle la **dimension** de  $E$  et se note  $\dim E$ .

### V.3 Matrices

Soit  $m$  et  $n$  deux entiers strictement positifs. On appelle matrice à  $m$  lignes et  $n$  colonnes à coefficients dans  $\mathbb{K}$  un ensemble  $A$  de  $mn$  scalaires  $a_{ij}$  de  $\mathbb{K}$ ,  $i = 1, \dots, m$ ,  $j = 1, \dots, n$ , présentés dans le tableau rectangulaire suivant

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$$

Les scalaires  $a_{ij}$ ,  $i = 1, \dots, m$ ,  $j = 1, \dots, n$ , sont appelés **coefficients**, ou **éléments**, de la matrice  $A$ .



On appelle **diagonale** d'une matrice  $A$  d'ordre  $n$  l'ensemble des coefficients  $a_{ii}$ ,  $i = 1, \dots, n$ . Cette diagonale divise la matrice en une partie **sur-diagonale**, composée des éléments dont l'indice de ligne est strictement inférieur à l'indice de colonne, et une partie sous-diagonale formée des éléments pour lesquels l'indice de ligne est strictement supérieur à l'indice de colonne.

Étant donné  $A \in M_{m,n}(\mathbb{R})$ , on note  $A^T \in M_{n,m}(\mathbb{R})$ , la matrice transposée de  $A$  telle que  $(A^T)_{ij} = (A)_{ji}$ ;  $1 \leq i \leq n$ ;  $1 \leq j \leq m$ , On a alors  $(A^T)^T = A$ .

On appelle **vecteur ligne** (resp. **vecteur colonne**) une matrice n'ayant qu'une ligne (resp. Colonne). Nous supposons toujours qu'un vecteur est un vecteur colonne, c'est-à-dire que l'on représentera le vecteur  $v$  dans la base  $\{e_i\}_{i=1,\dots,n}$  par

$$v = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix}$$

### V.3.1 Opérations sur les matrices

Nous rappelons à présent quelques opérations essentielles définies sur les matrices.

#### Égalité de matrices

Soit  $A$  et  $B$  deux matrices de  $M_{n,m}(\mathbb{K})$ . On dit que  $A$  est égale à  $B$  si  $a_{ij} = b_{ij}$  pour  $i = 1, \dots, m$ ,  $j = 1, \dots, n$ .

#### Somme de deux matrices

Soit  $A$  et  $B$  deux matrices de  $M_{n,m}(\mathbb{K})$ . On appelle somme des matrices  $A$  et  $B$  la matrice  $C$  de  $M_{n,m}(\mathbb{K})$  dont les coefficients sont  $c_{ij} = a_{ij} + b_{ij}$ ,  $i = 1, \dots, m$ ,  $j = 1, \dots, n$ .

L'élément neutre pour la somme de matrices est la matrice nulle, notée  $0$ , dont les coefficients sont tous égaux à zéro. On rappelle que l'on a par ailleurs

$$(A + B)^T = A^T + B^T \text{ et } (A + B)^* = A^* + B^*, \forall A, B \in M_{m,n}(\mathbb{K}).$$

#### Multiplication d'une matrice par un scalaire



Soit  $A$  une matrice de  $M_{m,n}(\mathbb{K})$  et  $\lambda$  un scalaire. Le résultat de la multiplication de la matrice  $A$  par le scalaire  $\lambda$  est la matrice  $C$  de  $M_{m,n}(\mathbb{K})$  dont les coefficients sont  $c_{ij} = \lambda a_{ij}$ ,  $i = 1, \dots, m, j = 1, \dots, n$ .

### *Produit de deux matrices)*

Soit  $A$  une matrice de  $M_{m,p}(\mathbb{K})$  et  $B$  une matrice de  $M_{p,n}(\mathbb{K})$ . Le produit des matrices  $A$  et  $B$  est la matrice  $C$  de  $M_{m,n}(\mathbb{K})$  dont les coefficients sont donnés par

$$c_{ij} = \sum_{k=1}^p a_{ik}b_{jk}, \quad i = 1, \dots, m, \quad j = 1, \dots, n.$$

Le produit de matrices est associatif et distributif par rapport à la somme de matrices, mais il n'est pas commutatif en général.

### *V.2.2 Liens entre applications linéaires et matrices*

Dans cette section, on va établir qu'une matrice est la représentation d'une application linéaire entre deux espaces vectoriels, chacun de dimension finie, relativement à des bases données. Pour cela, quelques rappels sont nécessaires.

### *Application linéaire*

Soient  $E$  et  $F$  deux espaces vectoriels sur le même corps  $\mathbb{K}$  et  $f$  une application de  $E$  dans  $F$ . On dit que  $f$  est une **application linéaire** si

$$f(\lambda v + w) = \lambda f(v) + f(w), \quad \forall (v, w) \in E^2, \quad \forall \lambda \in \mathbb{K}.$$

L'ensemble des applications linéaires de  $E$  dans  $F$  est noté  $\mathcal{L}(E, F)$ .

### *Noyau, image et rang*

Soit  $f$  une application de  $\mathcal{L}(E, F)$ . On appelle noyau (kernel en anglais) de  $f$ , et l'on note  $\text{Ker } f$ , l'ensemble

$$\text{Ker } f = \{x \in E \mid f(x) = 0\}$$

On dit que  $f$  est injective si  $\text{Ker } f = \{0\}$ .

On appelle image de  $f$ , et l'on note  $\text{Im } f$ , l'ensemble

$$\text{Im } f = \{y \in F \mid \exists x \in E, y = f(x)\}$$

Et le rang de  $f$  est la dimension de  $\text{Im } f$ . L'application  $f$  est dite **surjective** si  $\text{Im } f = F$ .

Enfin, on dit que  $f$  est **bijjective**, ou que c'est un **isomorphisme**, si elle est injective et bijective.





Soit  $A$  une matrice de  $M_{m,n}(\mathbb{K})$ , avec  $\mathbb{K} = \mathbb{R}$  ou  $\mathbb{C}$ . **Le noyau** de  $A$  est le sous-espace vectoriel de  $\mathbb{K}^n$  défini par

$$\text{Ker}(A) = \{x \in \mathbb{K}^n \mid Ax = 0\}$$

**L'image** de  $A$  est le sous-espace vectoriel de  $\mathbb{K}^m$  défini par

$$\text{Im}(A) = \{y \in \mathbb{K}^m \mid \exists x \in \mathbb{K}^n \text{ tel que } Ax = y\}$$

Et le rang de  $A$  est la dimension de cette image,

$$\text{rg}(A) = \dim(\text{Im}(A))$$

En vertu du théorème du rang, on a, pour toute matrice de  $M_{m,n}(\mathbb{K})$ , la relation

$$\dim(\text{Ker}(A)) + \text{rg}(A) = n.$$

### V.3.3 Inverse d'une matrice

Soit  $A$  une matrice d'ordre  $n$ . On dit que  $A$  est **inversible** (ou **régulière**) s'il existe une (unique) matrice, notée  $A^{-1}$ , telle que  $AA^{-1} = A^{-1}A = I_n$  ( $A^{-1}$  est appelée la **matrice inverse** de  $A$ ). Une matrice non inversible est dite **singulière**.

Il ressort de cette définition qu'une matrice  $A$  inversible est la matrice d'un endomorphisme bijectif.

Par conséquent, une matrice  $A$  d'ordre  $n$  est inversible si et seulement si  **$\text{rg}(A) = n$** .

Si une matrice  $A$  est inversible, son inverse est évidemment inversible et  **$(A^{-1})^{-1} = A$** .

### V.3.4 Trace et déterminant d'une matrice

#### Trace d'une matrice

La trace d'une matrice  $A$  d'ordre  $n$  est la somme de ses coefficients diagonaux :

$$\text{tr}(A) = \sum_{i=1}^n a_{ii},$$

On montre facilement les relations.

$$\text{tr}(A + B) = \text{tr}(A) + \text{tr}(B), \quad \text{tr}(AB) = \text{tr}(BA), \quad \text{tr}(\alpha A) = \alpha \text{tr}(A), \quad \forall \alpha \in \mathbb{K}, \forall A, B \in M_n(\mathbb{K})$$

La seconde ayant comme conséquence le fait que la trace d'une matrice est invariante par changement de base.

En effet, pour toute matrice  $A$  et toute matrice inversible  $P$  de même ordre, on a

$$\text{tr}(PAP^{-1}) = \text{tr}(P^{-1}PA) = \text{tr}(A).$$



*Déterminant d'une matrice*

On appelle déterminant d'une matrice A d'ordre n le scalaire défini par la formule de *Leibniz*.

$$\det(A) = \sum_{\sigma \in \mathfrak{S}_n} \varepsilon(\sigma) \prod_{i=1}^n a_{\sigma(i)i}$$

Où  $\varepsilon(\sigma)$  désigne la signature d'une permutation  $\sigma$  de  $\mathfrak{S}_n$ .

Rappelons qu'une permutation d'un ensemble est une bijection de cet ensemble dans lui-même. On note  $\mathfrak{S}_n$  le groupe (pour la loi de composition) des permutations de l'ensemble  $\{1, \dots, n\}$ , avec  $n \in \mathbb{N}$ . La signature d'une permutation  $\sigma$  de  $\mathfrak{S}_n$  est le nombre, égal à 1 ou -1, défini par

$$\varepsilon(\sigma) = \prod_{1 \leq i < j \leq n} \frac{\sigma(i) - \sigma(j)}{i - j}.$$

*Mineur, cofacteur*

Soit A une matrice d'ordre n. On appelle mineur associé à l'élément  $a_{ij}$ ,  $1 \leq i, j \leq n$ , de A le déterminant d'ordre n-1 de la matrice obtenue par suppression de la i<sup>ème</sup> et de la j<sup>ème</sup> colonne de A. On appelle cofacteur associé à ce même élément le scalaire

$$\text{Cof}_{ij}(A) = (-1)^{i+j} \begin{vmatrix} a_{11} & \dots & a_{1j-1} & a_{1j+1} & \dots & a_{1n} \\ \vdots & & \vdots & \vdots & & \vdots \\ a_{i-11} & \dots & a_{i-1j-1} & a_{i-1j+1} & \dots & a_{i-1n} \\ a_{i+11} & \dots & a_{i+1j-1} & a_{i+1j+1} & \dots & a_{i+1n} \\ \vdots & & \vdots & \vdots & & \vdots \\ a_{n1} & \dots & a_{nj-1} & a_{nj+1} & \dots & a_{nn} \end{vmatrix}$$

Enfin, on appelle **matrice des cofacteurs**, ou **comatrice**, de A est la matrice d'ordre n constituée de l'ensemble des cofacteurs de A,

$$\tilde{A} = (\text{Cof}_{ij}(A))_{1 \leq i, j \leq n}$$

*Formule de Laplace*

Soit A une matrice d'ordre n. On a

$$\det(A) = \sum_{k=1}^n a_{ik} \text{Cof}_{ik}(A) = \sum_{k=1}^n a_{kj} \text{Cof}_{kj}(A), \quad \forall (i, j) \in \{1, \dots, n\}^2.$$

Lorsque la matrice A est inversible, on a obtenu une formule pour l'inverse de A,

$$A^{-1} = \frac{1}{\det(A)} \tilde{A}^T$$

Qui ne nécessite que des calculs de déterminants.



### V.3.5 Valeurs et vecteurs propres

Les valeurs propres d'une matrice  $A$  d'ordre  $n$  sont les  $n$  racines  $\lambda_i$ ,  $i = 1, \dots, n$ , réelles ou complexes, distinctes ou confondues, du polynôme caractéristique

$$\lambda \in \mathbb{C} \rightarrow \det(A - \lambda I_n)$$

Associé à  $A$ . Le spectre d'une matrice  $A$ , noté  $\sigma(A)$ , est l'ensemble des valeurs propres de  $A$ . On rappelle les propriétés suivantes :

$$\operatorname{tr}(A) = \sum_{i=1}^n \lambda_i, \quad \det(A) = \prod_{i=1}^n \lambda_i.$$

Par conséquent, la matrice  $A$  est singulière si au moins une de ses valeurs propres est nulle.

Enfin, le rayon spectral d'une matrice  $A$  est le nombre défini par

$$\rho(A) = \max_{1 \leq i \leq n} |\lambda_i|.$$

À toute valeur propre  $\lambda$  d'une matrice  $A$  est associé au moins un vecteur non nul  $v$  tel que

$$Av = \lambda v$$

Et appelé vecteur propre de la matrice  $A$  correspondant à la valeur propre  $\lambda$ .

### V.3.6 Matrices semblables

On dit que deux matrices  $A$  et  $B$  d'ordre  $n$  sont semblables s'il existe une matrice d'ordre  $n$  inversible  $P$  telle que  $A = PBP^{-1}$ .

L'exploitation la notion de matrices semblables permet entre autres de réduire la complexité du problème de l'évaluation des valeurs propres d'une matrice. En effet, si l'on sait transformer une matrice donnée en une matrice semblable diagonale ou triangulaire, le calcul des valeurs propres devient alors immédiat.



### V.3.7 Quelques matrices particulières

#### Matrices symétriques

Soit  $A \in M_n(\mathbb{R})$ . On dit que la matrice  $A$  est **symétrique** si  $A = A^T$ , **antisymétrique** si  $A = -A^T$  et **orthogonale** si  $A^T A = A A^T = I_n$ .

#### Matrices diagonales

Les matrices diagonales interviennent à de nombreuses reprises en algèbre linéaire. Elles vérifient des propriétés qui rendent leur manipulation particulièrement aisée d'un point de vue calculatoire.

Une matrice  $A$  d'ordre  $n$  est dite **diagonale** si on a  $a_{ij} = 0$  pour les couples d'indices  $(i, j) \in \{1, \dots, n\}^2$  tels que  $i \neq j$ .

#### Matrices triangulaires

Les matrices triangulaires forment une classe de matrices intervenant très couramment en algèbre linéaire numérique.

On dit qu'une matrice  $A$  d'ordre  $n$  est triangulaire supérieure (resp. inférieure) si on a  $a_{ij} = 0$  pour les couples d'indices  $(i, j) \in \{1, \dots, n\}^2$  tels que  $i > j$  (resp.  $i < j$ ).

#### Matrices bandes

Une matrice bande est une matrice carrée dont les coefficients non nuls sont localisés dans une **bande** autour de la diagonale principale. Plus précisément, on a la définition suivante.

Soit  $n$  un entier strictement positif. On dit qu'une matrice  $A \in M_n(\mathbb{R})$  est une matrice bande s'il existe des entiers positifs  $p$  et  $q$  strictement inférieurs à  $n$  tels que  $a_{ij} = 0$  pour tous les couples d'entiers  $(i, j) \in \{1, \dots, n\}^2$  tels que  $i - j > p$  ou  $j - i > q$ . La largeur de bande de la matrice vaut  $p + q + 1$ , avec  $p$  éléments a priori non nuls à gauche de la diagonale et  $q$  éléments à droite sur chaque ligne.

#### Matrices à diagonale dominante

On dit qu'une matrice  $A$  d'ordre  $n$ , est à **diagonale dominante** par lignes (respectivement par colonnes) si



$$|a_{ii}| \geq \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}| \quad (\text{resp. } |a_{ii}| \geq \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ji}|), \quad 1 \leq i \leq n.$$

On dit que A est à diagonale strictement dominante (par lignes ou par colonnes respectivement) si ces inégalités sont strictes.

## V.4 Normes et produits scalaires

### V.4.1 Définitions

Soit E un espace vectoriel sur le corps  $\mathbb{K}$ , avec  $\mathbb{K} = \mathbb{R}$  ou  $\mathbb{C}$ . On dit qu'une application  $\|\cdot\|$  de E dans  $\mathbb{R}$  est une norme sur

1.  $\|v\| \geq 0$ ,  $\forall v \in E$ , et  $\|v\| = 0$  si et seulement si  $v = \mathbf{0}$ ,
2.  $\|\alpha v\| = |\alpha| \|v\|$ ,  $\forall \alpha \in \mathbb{K}$ ,  $\forall v \in E$ ,
3. elle vérifie l'**inégalité triangulaire**, c'est-à-dire

$$\|u + v\| \leq \|u\| + \|v\|, \quad \forall u, v \in E.$$

Un produit scalaire sur un espace vectoriel E sur  $\mathbb{R}$  est une application  $(\cdot, \cdot)$  de  $E \times E$  dans  $\mathbb{R}$  possédant les propriétés suivantes :

1. elle est **bilinéaire** (resp. **sesquilinéaire**), c'est-à-dire linéaire par rapport à la première variable

$$(\alpha u + v, w) = \alpha (u, w) + (v, w), \quad \forall u, v, w \in E, \quad \forall \alpha \in \mathbb{R} \text{ (resp. } \mathbb{C}),$$

et linéaire (resp. antilinéaire) par rapport à la seconde

$$(u, \alpha v + w) = \alpha (u, v) + (u, w) \quad (\text{resp. } (u, \alpha v + w) = \bar{\alpha} (u, v) + (u, w)),$$

$$\forall u, v, w \in E, \quad \forall \alpha \in \mathbb{R} \text{ (resp. } \mathbb{C}),$$

2. elle est **symétrique** (resp. à **symétrie hermitienne**), c'est-à-dire

$$(u, v) = (v, u) \quad (\text{resp. } (u, v) = \overline{(v, u)}), \quad \forall u, v \in E,$$

3. elle est **définie positive**<sup>11</sup>, c'est-à-dire

$$(v, v) \geq 0, \quad \forall v \in E, \quad \text{et } (v, v) = 0 \text{ si et seulement si } v = \mathbf{0}.$$

### V.4.2 Produits scalaires et normes vectoriels

Nous nous intéressons maintenant aux produits scalaires et normes définis sur l'espace vectoriel de dimension finie  $\mathbb{K}^n$ , avec  $\mathbb{K} = \mathbb{R}$ ,  $n \in \mathbb{N}^*$ .

L'application  $(\cdot, \cdot) : \mathbb{K}^n \times \mathbb{K}^n \rightarrow \mathbb{K}^n$  définie par



$$(\mathbf{u}, \mathbf{v}) = \mathbf{v}^T \mathbf{u} = \mathbf{u}^T \mathbf{v} = \sum_{i=1}^n u_i v_i \text{ si } \mathbb{K} = \mathbb{R},$$

Est appelée produit scalaire canonique (et produit scalaire euclidien lorsque  $\mathbb{K} = \mathbb{R}$ ). On note La norme induite par ce produit scalaire, appelée norme euclidienne dans le cas réel, est alors

$$\|\mathbf{v}\|_2 = \sqrt{(\mathbf{v}, \mathbf{v})} = \left( \sum_{i=1}^n |v_i|^2 \right)^{1/2}$$

### V.4.3 Normes de matrices

Une matrice carrée A est à diagonale dominante si

$$\forall i \in [1, n], |a_{ii}| > \sum_{j=1, j \neq i}^n |a_{ij}|$$

$$\begin{pmatrix} 7 & 2 & 0 \\ 3 & 5 & -1 \\ 0 & 5 & -6 \end{pmatrix} \quad \begin{array}{l} |a_{11}| > |a_{12}| + |a_{13}| \quad |7| > |2| + |0| \\ |a_{22}| > |a_{21}| + |a_{23}| \quad |5| > |3| + |-1| \\ |a_{33}| > |a_{31}| + |a_{32}| \quad |-6| > |0| + |5| \end{array}$$

La matrice symétrique  $A \in M_{n \times n}$  est définie positive si et seulement si tous ses mineurs principaux sont strictement positifs ( $\Delta_i=1, n \Delta_i > 0$ )

$$A = \begin{bmatrix} 20 & 12 & 5 \\ 12 & 15 & 2 \\ 5 & 2 & 25 \end{bmatrix} \quad A^T = \begin{bmatrix} 20 & 12 & 5 \\ 12 & 15 & 2 \\ 5 & 2 & 25 \end{bmatrix}$$

1.  $\Delta_1 = 20 > 0$ :

2.  $\Delta_2 = \begin{vmatrix} 20 & 12 \\ 12 & 15 \end{vmatrix} = (20) * (15) - (12) * (12) = 156 > 0$

3.  $\Delta_3 = \begin{vmatrix} 20 & 12 & 5 \\ 12 & 15 & 2 \\ 5 & 2 & 25 \end{vmatrix} = 20 \begin{vmatrix} 15 & 2 \\ 2 & 25 \end{vmatrix} - 12 \begin{vmatrix} 12 & 2 \\ 5 & 25 \end{vmatrix} + 5 \begin{vmatrix} 12 & 15 \\ 5 & 2 \end{vmatrix}$   
 $= 7420 - 3480 - 255 = 3685 > 0$

Les normes matricielles de  $\|A\|$  en termes des éléments de A que l'on note  $a_{ij}$



$$\|A\|_1 = \max_{j=1\dots n} \sum_{i=1}^n |a_{ij}| \quad \|A\|_\infty = \max_{i=1\dots n} \sum_{j=1}^n |a_{ij}|$$

$$A = \begin{bmatrix} 20 & 12 & 5 \\ 12 & 15 & 2 \\ 5 & 2 & 25 \end{bmatrix} \quad B = \begin{bmatrix} 2 & 0 & 1 \\ 0 & -1 & 1 \\ 1 & 0 & -2 \end{bmatrix}$$

$$\|A\|_1 = \max_{j=1\dots n} \sum_{i=1}^n |a_{ij}| = \max[|20| + |12| + |5|, |12| + |15| + |2|, |5| + |2| + |25|] \\ = \max\{37, 29, 32\} = 37$$

$$\|B\|_\infty = \max_{i=1\dots n} \sum_{j=1}^n |b_{ij}| = \max[|2| + |0| + |1|, |0| + |-1| + |1|, |1| + |0| + |-2|] \\ = \max\{3, 2, 3\} = 3$$

## V.5 Conclusion

La notion de norme est particulièrement utile en algèbre linéaire numérique pour quantifier l'erreur de l'approximation de la solution d'un système linéaire par une méthode itérative (voir le chapitre III),



## Conclusion générale

Dans la métrique, nous sommes souvent confrontés à travailler sur des problèmes de nature continue. Cependant, les ordinateurs sont conçus pour manipuler des données discrètes. A titre d'exemple, les ordinateurs ne possèdent que l'approximation. Aussi, ils n'utilisent que des approximations pour les fonctions mathématiques de base telle que le sin, cos, etc. c'est là où intervient l'analyse numérique en proposant différents algorithmes et méthodes, Par exemples : résolution des systèmes d'équations linéaires, détermination des valeurs et vecteurs propres d'une matrice,

Il reste aussi dans cette direction d'autres notions à traiter par la suite comme la résolution des équations différentielles, calcul des zéros d'une fonction, évaluation des intégrales, interpolation et approximation ,..





## Fiche de TP 1

L'algorithme du Gauss consiste à transformer un système linéaire en un système triangulaire supérieur qui lui est équivalent ;

Autrement dit (vision matricielle), l'algorithme du pivot transforme le système :

$Ax = b$  avec  $A$  quelconque, en le système équivalent  $Ux = v$  avec  $U$  triangulaire supérieure.

Rappelons pour finir l'algorithme, tel qu'il a été présenté dans le cours :

**1<sup>ère</sup> étape : Triangularisation :  $[A, b] \rightarrow [U, b']$**

**Pour  $k=1$  à  $n-1$  faire**

**Pour  $i=k+1$  à  $n$  faire**

$$w = \frac{a_{ik}}{a_{kk}}$$

**Pour  $j=k$  à  $n+1$  faire**

$$a_{ij} = a_{ij} - w \cdot a_{kj}$$

**Fin pour**

**Fin pour**

**Fin pour**

**2<sup>ème</sup> étape : Résolution du système :  $Ux=b'$**

**Pour  $i=n$  à  $1$  faire**

$$x_i = \frac{b_i - \sum_{j=i+1}^n a_{ij}x_j}{a_{ii}}$$

**Fin pour**

**Ecrire un programme Matlab qui réalise cette méthode et tester le avec l'exemple cité dans le cours.**



**Aide**

Vous utilisez les fonctions Matlab suivantes :

Input pour l'entrée

disp pour l'affichage

Vous utilisez les opérateurs de division, de multiplication.



## Solution de la fiche de TP 1

### SOUS MATLAB

```
clear all
close all
%%taille de la matrice
taille=input('Taille de la matrice : ');

%%Remplissage matrice
for t = 1:taille
for p = 1:taille
str=sprintf('Entrer l'élément (%d,%d) de la matrice A : ',t,p);
a(t,p) = input(str);
end
end

%%remplissage vecteur b
for t = 1:taille
str=sprintf('Entrer l'élément (%d) du vecteur b : ',t);
b(t) = input(str);
end

%%Affichage A et B
disp(a)
disp(b')

%%Concatenation
A=[a,b']

n=size(A,1)

for k=1:n-1
```



```
for i=k+1:n
w=A(i,k)/A(k,k)
for j=k:n+1
A(i,j)=A(i,j)-w*A(k,j)
end
end
end
A
%%Resolution du systeme
for i=n:-1:1
s=0;
for j=i+1:n
s=s+A(i,j)*x(j);
end
x(i)=(A(i,n+1)-s)/A(i,i);
end
x
```



## Fiche de TP 2

Décomposition de la matrice A de façon à la mettre sous la forme  $A=L.U$  où L est une matrice triangulaire unitaire inférieure et U est une matrice triangulaire supérieure.

Le système devient :

$$Ax = b \Leftrightarrow L U x = b \Leftrightarrow \begin{cases} Ly = b \\ Ux = y \end{cases}$$

La résolution du système  $Ax=b$  revient à résoudre les deux systèmes  $Ly=b$  et  $Ux=y$ .  
Puisque L et U sont triangulaires, la résolution est immédiate.

$$L = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ l_{21} & 1 & 0 & \cdots & 0 \\ l_{31} & l_{32} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \cdots & 1 \end{pmatrix} \quad U = \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1n-1} & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n-1} & u_{2n} \\ 0 & 0 & \cdots & u_{3n-1} & u_{3n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & u_{nn} \end{pmatrix}$$

Voici l'algorithme de factorisation présenté dans le cours :

```

Début
  u11 = a11
  Pour j=2 à n faire
    u1j = a1j
    lj1 = aj1 / u11
  Fin pour
  Pour i=2 à n-1 faire
    uii = aii - ∑k=1i-1 likuki
    Pour j=i+1 à n faire
      uij = aij - ∑k=1i-1 likukj
      lji = (aji - ∑k=1i-1 ljkuki) / uii
    Fin pour
  Fin pour
  unn = ann - ∑k=1n-1 lnkukn
Fin
```

### Question

Ecrire un programme Matlab qui réalise la factorisation et la résolution du système d'équations.



## Solution de la fiche de TP 2

### SOUS MATLAB

```
clear all
close all
%%taille de la matrice
taille=input('Taille de la matrice : ');

%%Remplissage matrice
for t = 1:taille
for p = 1:taille
str=sprintf('Entrer l'élément (%d,%d) de la matrice A : ',t,p);
A(t,p) = input(str);
end
end

%%remplissage vecteur b
for t = 1:taille
str=sprintf('Entrer l'élément (%d) du vecteur b : ',t);
B(t) = input(str);
end

n = length(A);
U = A;
L = eye(n);

% cONSTRUCTION U and L
for k = 1 : n-1
for i = k+1 : n
factor = U(i, k) / U(k, k);
L(i, k) = factor;
U(i, k:n) = U(i, k:n) - factor*U(k, k:n);
end
end
```



```
end
%MAINTENANT VOUS AVEZ L ET U
L
U
% TROUVER VECTEUR Y
Y = zeros(n,1);
Y(1) = B(1);
for i = 2 : n
    Y(i) = B(i) - L(i,i-1:-1:1)*Y(i-1:-1:1);
end

%substitution INVERSE POUR AVOIR LE X A PARTIR DE Y
x=zeros(n,1);
x(n) = Y(n) / U(n,n);
for i = n-1 : -1 : 1
    x(i) = (Y(i) - U(i,i+1:n)*x(i+1:n)) / U(i,i);
end
```



## Fiche de TP 3

### Mise en œuvre sous Matlab

Pour le système suivant, on va utiliser les méthodes de Jacobi et de Gauss-Seidel (en considérant que le vecteur initiale est  $X_0=(0,0,0)$ ):

$$\begin{cases} 4x_1 - x_2 + x_3 = 7 \\ 4x_1 - 8x_2 + x_3 = -21 \\ -2x_1 + x_2 + 5x_3 = 15 \end{cases}$$

Ecrire un programme qui initialise l'exemple donné et selon un choix il fait appel à l'une des deux fonctions que vous devez les implémenter selon les déclarations suivantes :

`function [X,niter] = jacobi(A,b,X0,nmax,tol)`

`function [X,niter] = gseidel(A,b,X0,nmax,tol)`

Où    A est la matrice du système  
       b est le vecteur des données  
       X0 Le vecteur initial  
       nmax est le nombre d'itération maximale  
       tol est l'erreur comme condition d'arrêt.

### Rappel des algorithmes

#### Algorithme de Jacobi

Données :  $b, A, X^{(0)}, \varepsilon$

**Début**

$k=0$

**Tant que**  $\frac{\|r^{(k)}\|}{\|b\|} \geq \varepsilon$  **faire**

**Pour**  $i=1$  à  $n$  **faire**

$$r_i^{(k)} = b_i -$$

$$\sum_{j=1}^n a_{ij}x_j^{(k)}$$





$$x_i^{(k+1)} = (x_i^{(k)} + r_i^{(k)})/a_{ii}$$

**Fin pour**

k=k+1

**Fin tant que**

**Fin**

### Algorithme de Gauss-Seidel

**Début**

k=0

**Tant que**  $|x_i^{(k+1)} - x_i^{(k)}| > \varepsilon$  **faire**

**Pour** i=1 à n **faire**

$$x_i^{(k+1)} = [b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)}]/a_{ii}$$

**Fin pour**

k=k+1

**Fin tant que**

**Fin**



## Solution de la fiche de TP 3

### SOUS MATLAB

```

clear all;
clc;

%Initialisation de la matrice du système
a=[4 -1 1;4 -8 1;-2 1 5];

%Initialisation du vecteur des données
b=[7 ;-21 ;15];

%Nombre de variables et d'équations
n=3;

%Le vecteur X0
X0 = [0;0;0];
disp(['*Résolution numérique de l''équation linéaire A.X=B *'])
disp(['*** S=1. Pour la méthode de Jacobi ***'])
disp(['*** S=2. Pour la méthode de Gauss-Seidel ***'])
disp(['*** S=3. Pour Quitter ***'])
disp(['*** Tapez S entre <1-3> ***'])
s=input('S =');
switch s
case 1
disp(['*** Début de la méthode de Jacobi ***'])
[X,N] = jacobi(a,b,X0,50,0.0001) %La fonction jacobi.m
%50 Nombre maximal d'itérations
% N le nombre d'itération
% 0.0001 la précision
case 2

```



```

disp(['*** Début de la méthode de Gauss-Seidel ***'])
[X,N] = gseidel(a,b,X0,50,0.0001) %La fonction gseidel.m
case 3
quit
otherwise
disp('Erreur! la valeur de S entre <1-3> ');
end

```

```

%=====FIN=====
=====

```

```

%=====La fonction jacobi.m
=====

```

```

function [X,niter] = jacobi(A,b,X0,nmax,tol)

```

```

n = length(b); X = X0;

```

```

for niter = 1:nmax

```

```

% Calculer l'itération suivante

```

```

for i = 1:n

```

```

j = [1:i-1,i+1:n];

```

```

somme = A(i,j)*X0(j);

```

```

X(i) = (b(i)-somme)/A(i,i);

```

```

end

```

```

% Tester la convergence

```

```

if norm(X-X0) < tol

```

```

return

```

```

end

```

```

% L'ancien X0 devient le nouveau X

```

```

X0 = X;

```

```

end

```

```

% En cas de divergence

```



```
%=====La fonction gseidel.m
```

```
=====
```

```
function [X,niter] = gseidel(A,b,X0,nmax,tol)
```

```
n = length(b); X = X0;
```

```
for niter = 1:nmax
```

```
% Calculer l'itération suivante
```

```
for i = 1:n
```

```
somme1 = A(i,1:i-1)*X(1:i-1);
```

```
somme2 = A(i,i+1:n)*X0(i+1:n);
```

```
X(i)=(b(i)-somme1-somme2)/A(i,i);
```

```
end
```

```
% Tester la convergence
```

```
if norm(X-X0) < tol
```

```
return
```

```
end
```

```
% L'ancien X0 devient le nouveau X
```

```
X0 = X;
```

```
end
```

```
% En cas de divergence
```

```
disp('Pas de convergence')
```

```
%=====FIN=====
```

```
=====
```



## Fiche de TP 4

### Méthode de Cholesky

Il est demandé à chaque étudiant préparer le travail suivant :

1. Une description de la méthode avec un exemple.
2. L'algorithme et le code Matlab associé.

### Solution de la fiche de TP 4

#### Rappel

Une matrice  $A$  est symétrique si et seulement si :  $A = A^t$

Elle est définie positive si et seulement si :  $\forall x \in \mathbb{R}^n \setminus \{0\} : x^t A x > 0$

Exemple : soit la matrice  $\begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix}$

Cette matrice est symétrique.

On a :  $(x_1, x_2, x_3) \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = x_1^2 + x_2^2 + x_3^2 + (x_1 + x_2 + x_3)^2 > 0 \forall x \neq 0$

Ainsi, la matrice est définie positive.

Remarques : soit  $A \in \mathbb{R}^{n \times n}$  une matrice symétrique.

- $A$  est définie positive si et seulement si toutes ses valeurs propres sont positives.
- $A$  est définie positive si et seulement si toutes les sous-matrices :

$$\begin{pmatrix} a_{11} & \cdots & a_{1r} \\ \vdots & \ddots & \vdots \\ a_{r1} & \cdots & a_{rr} \end{pmatrix}; 1 \leq r \leq n$$

Ont un déterminant positif.

Il est intéressant de trouver une factorisation de  $A$  qui prend en compte sa symétrie.

*Théorème : (Cholesky)*



Soit  $A$  une matrice carrée d'ordre  $n$  non singulière ( $\det(A) \neq 0$ ) et symétrique. Pour qu'il existe une matrice triangulaire inférieure  $L$ , de même dimension que  $A$ , telle que  $A = LL^t$ , il faut et il suffit que  $A$  soit définie positive.

**Remarque :**

- $L$  n'est pas unique.
- La décomposition devient unique si l'on fixe à l'avance les éléments diagonaux  $l_{ii}$  avec  $l_{ii} > 0$ .
- On remarque que cette décomposition est un peu similaire à la décomposition LU. Cependant, au lieu d'utiliser deux matrices, une suffit pour stocker les valeurs de  $L$ .
- Un autre point important est que la matrice triangulaire inférieure produite par la factorisation LU est différente de celle produite par la factorisation de Cholesky

Avant de donner l'algorithme général de factorisation de Cholesky, étudions un exemple.

**Exemple :** pour trouver la décomposition de Cholesky de la matrice :

$$A = \begin{pmatrix} 4 & 2 & -6 \\ 2 & 10 & 9 \\ -6 & 9 & 26 \end{pmatrix}$$

$$\text{On doit résoudre l'équation : } \begin{pmatrix} 4 & 2 & -6 \\ 2 & 10 & 9 \\ -6 & 9 & 26 \end{pmatrix} = \underbrace{\begin{pmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{pmatrix}}_L \underbrace{\begin{pmatrix} l_{11} & l_{21} & l_{31} \\ 0 & l_{22} & l_{32} \\ 0 & 0 & l_{33} \end{pmatrix}}_{L^t}$$

Pour ce faire, on écrit la matrice de multiplication pour trouver un système d'équations à 6 inconnus. L'astuce est de trouver les  $l_{ij}$  dans le bon ordre. On obtient les équations :

$$\begin{aligned} l_{11}^2 &= 4 & \Rightarrow & l_{11} = 2 \\ l_{21}l_{11} &= 2 & \Rightarrow & l_{21} = 1 \\ l_{31}l_{11} &= -6 & \Rightarrow & l_{31} = -3 \\ l_{21}^2 + l_{22}^2 &= 10 & \Rightarrow & l_{22} = 3 \\ l_{31}l_{21} + l_{32}l_{22} &= 9 & \Rightarrow & l_{32} = 4 \\ l_{31}^2 + l_{32}^2 + l_{33}^2 &= 26 & \Rightarrow & l_{33} = 1 \end{aligned}$$

$$\text{En conclusion, on a : } L = \begin{pmatrix} 2 & 0 & 0 \\ 1 & 3 & 0 \\ -3 & 4 & 1 \end{pmatrix}$$

Pour une matrice d'ordre  $n \times n$ , on calcule suivant l'ordre :



$$\begin{pmatrix} 1 & & & & & \\ 2 & 3 & & & & \\ \vdots & 4 & 5 & & & \\ \vdots & \vdots & \vdots & \ddots & & \\ 2 & 4 & 6 & \cdots & 2n-1 & \end{pmatrix}$$

Afin d'obtenir les éléments  $l_{ij}$  de la matrice  $L$  on multiplie les matrices  $L$  et  $L^t$ , puis on identifie les coefficients respectifs dans l'égalité :  $A = LL^t$  pour obtenir les équations :

$$\begin{cases} l_{11}^2 = a_{11} \\ l_{i1}^2 + l_{i2}^2 + \cdots + l_{ii}^2 = a_{ii} \\ l_{i1}l_{ji} + l_{i2}l_{ji} + l_{i3}l_{ji} + \cdots + l_{ii}l_{ji} = a_{ij}, i > j \\ l_{ij} = 0, i < j \end{cases}$$

On déduit ensuite :

$$\begin{cases} l_{11} = \sqrt{a_{11}} \\ l_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2}, \quad i = \overline{2, n} \\ l_{ji} = \frac{1}{l_{ii}} \left( a_{ij} - \sum_{k=1}^{i-1} l_{ik}l_{jk} \right), \quad i < j \\ l_{ji} = 0, i > j \end{cases}$$

### Algorithme de Cholesky

Début

Pour  $k=1$  à  $n$  faire

$$l_{kk} = \sqrt{a_{kk} - \sum_{s=1}^{k-1} l_{ks}^2}$$

Pour  $j=k+1$  à  $n$  faire

$$l_{jk} = \frac{1}{l_{kk}} (a_{kj} - \sum_{s=1}^{k-1} l_{js}l_{ks})$$

Fin pour

Fin pour

Fin

**SOUS MATLAB**

```
clear all
close all
%%taille de la matrice
taille=input('Taille de la matrice : ');

%%Remplissage matrice
for t = 1:taille
for p = 1:taille
str=sprintf('Entrer l'élément (%d,%d) de la matrice A : ',t,p);
a(t,p) = input(str);
end
end

%%remplissage vecteur b
for t = 1:taille
str=sprintf('Entrer l'élément (%d) du vecteur b : ',t);
b(t) = input(str);
end

%%Affichage A et B
M=A
disp(a)
disp(b')
n = length( M );
L = zeros( n, n );
for i=1:n
    L(i, i) = sqrt( M(i, i) - L(i, :)*L(i, :)' );

    for j=(i + 1):n
```





```
L(j, i) = ( M(j, i) - L(i, :)*L(j, :)' )/L(i, i);  
end  
end  
L  
y = L \ b  
x = L' \ y
```

## Fiche de TP 5

*Valeurs propres et vecteurs propres*

Quelques commandes de Matlab utiles pour ce TP :

- `lambda = eig(A)`: définit un vecteur lambda contenant les valeurs propres de A;
- `[V,L] = eig(A)`: définit une matrice V contenant les vecteurs propres de A, et une matrice diagonale L contenant ses valeurs propres;
- `A(:,1)`, `A(1,:)`, `A(1,1)`: première colonne/ligne de A, premier élément de A;
- `A = [B C]`: appond deux matrices ayant même nombre de lignes;
- pour représenter graphiquement une fonction  $f(t)$  sur  $[0, 2\pi]$ , on pourra utiliser `x = []; for t=0:0.1:2*pi x=[x f(t)]; end plot(x','-')`
- `sqrt(a)` calcule la racine carrée d'un nombre a.

Calculer les valeurs et vecteurs propres de

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \quad B = \begin{bmatrix} 3 & 4 \\ -1 & -1 \end{bmatrix} \quad C = \begin{bmatrix} -7 & -6 \\ 9 & 8 \end{bmatrix} \quad D = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$$

et vérifier les résultats. Si V et L contiennent respectivement les vecteurs propres et valeurs propres de A, calculer  $A*V$  et  $V*L$  et expliquer le résultat.



## Références

1. C. Brezinski, Introduction à la pratique du calcul numérique, Dunod, Paris 1988.
2. G. Allaire et S.M. Kaber, Algèbre linéaire numérique, Ellipses, 2002.
3. M. Crouzeix et A.-L. Mignot, Analyse numérique des équations différentielles, Masson, 1983.
4. J.-P. Demailly, Analyse numérique et équations différentielles. Presses Universitaires de Grenoble, 1996.
5. E. Hairer, S. P. Norsett et G. Wanner, Solving Ordinary Differential Equations, Springer, 1993.
6. P. G. Ciarlet, Introduction à l'analyse numérique matricielle et à l'optimisation, Masson, Paris, 1982.
7. M. Lakrib, Cours d'analyse numérique, OPU Alger 2008.
8. B. Demidovitch, I. Maron, Elément de calcul numérique, Ed. MIR, Moscou, 1979.



## Annexe

### Introduction à Matlab

Matlab est un logiciel de calcul numérique, utilisé dans de nombreux domaines d'application. Il se fonde sur le calcul matriciel. Matlab est d'ailleurs un raccourci pour **Matrix Laboratory**. Le but de ce document est d'aider les débutants en Matlab, en introduisant les commandes les plus courantes.

#### *Les Commandes Matlab*

##### *Navigation*

- `cd` change de répertoire
- `pwd` indique le répertoire courant
- `dir` liste le contenu du répertoire

##### *Initialisations*

- `clear`
- `clc` efface la fenêtre de commande
- `who` liste les variables
- `whos` liste les variables et affiche les détails

##### *Syntaxe*

- `,` séparateur d'instructions
- `;` séparateur d'instructions sans affichage
- `Début :pas :fin` génération de séquences
- `...` pour continuer à la ligne
- `%` ligne de commentaire
- `pi`  $\pi$
- `!` appel des commandes du système d'exploitation

##### *Calcul*

- `+ - _ /`
- `\` produit à droite



- `.` Opération termes à termes. exemple : `X.*Y`
- `^` élévation en puissance
- `'` transposition complexe
- `.'` transposition sans conjugaison
- `real` partie réelle
- `imag` partie imaginaire
- `abs` module
- `angle` argument, phase
- `roots` racine d'un polynôme
- `round, floor, ceil` valeurs approchées, partie entière

### *Matrices et vecteurs*

- `zeros` matrice remplie de 0
- `ones` matrice remplie de 1
- `eye` matrice identité
- `magic` matrice magique
- `rand, randn` matrices aléatoires
- `linspace` répartition linéaire de points dans un vecteur
- `logspace` répartition logarithmique de points dans un vecteur
- `find` renvoie les indices de termes vérifiant certaines conditions
- `sort` ordonne les éléments d'un vecteur ou d'une matrice
- `sum` somme des éléments
- `prod` produit des éléments
- `min,` max plus petit et plus grand élément
- `[X Y]` concaténation horizontale
- `[X ; Y]` concaténation verticale
- `X(:)` transforme X en un vecteur colonne
- `size` dimensions d'une matrice
- `length` longueur d'un vecteur
- `diag` extrait la diagonale d'une matrice, crée une matrice diagonale
- `inv` inverse d'une matrice



- trace      trace d'une matrice

### *Algèbre linéaire*

- eig      valeurs et vecteurs propres
- svd      décomposition en valeurs singulières
- kron      produit tensoriel de Kronecker

### *Statistiques*

- mean      moyenne
- median      valeur médiane
- std      écart-type
- corrcoef      coefficient de corrélation
- cov      matrice de covariance

### *Traitement du signal*

- fft      transformée de Fourier discrète
- ifft      transformée de Fourier discrète inverse
- dct      transformée en cosinus discrète
- idct      transformée en cosinus discrète inverse
- freqz      réponse en fréquence d'un \_ltre numérique
- impz      réponse impulsionnelle d'un \_ltre numérique
- zplane      tracé du lieu des zéros et des pôles d'un \_ltre numérique
- filter      applique un filtre numérique

### *Automatique*

- acker      formule d'Ackermann
- place      placement de pôles
- lqr      régulateur linéaire quadratique optimal - temps continu
- lqe      estimateur de Kalman - temps continu
- c2d      passage d'un modèle d'état continu à un modèle discret
- d2c      devine ? ! !

### *Programmation*

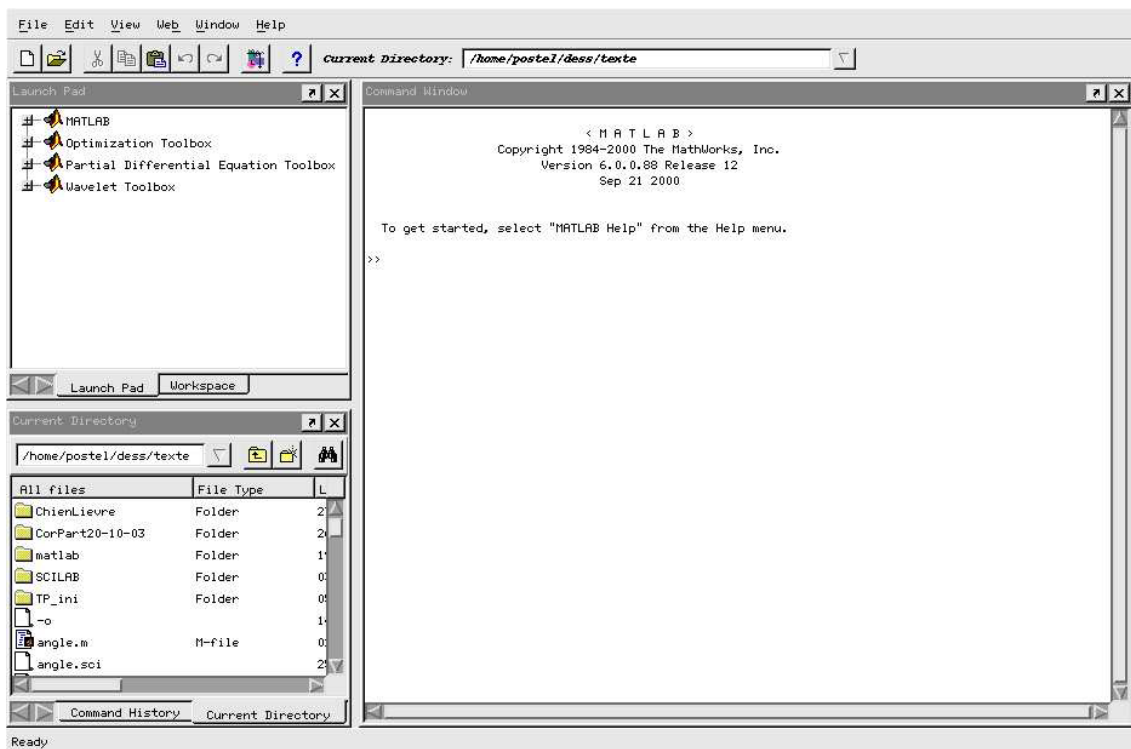
- if, elseif, else

## Méthodes numériques



- while
- for
- switch, case, otherwise
- end
- break
- continue

### *Fenêtre de commandes Matlab*





### **Espace E-learning**

Les cours en power point et pdf ainsi que les Fiches de TP avec solutions sont disponibles sur l'espace de l'enseignement à distance de l'université USTO-MB.

Suivre le lien

<https://elearning.univ-usto.dz/course/view.php?id=140>





---

## PARTIE COURS



### CHAPITRE 1

 **COURS 1 LES ERREURS**

Document PDF

 **Chapitre 1: Généralités sur l'analyse numérique et le calcul scientifique**

Document PDF

### CHAPITRE 2

 **COURS 2 METHODES RESO SE DIRECTES**

Document PDF

 **Méthodes directes de résolution des systèmes linéaires**

Document PDF

### CHAPITRE 3

 **COURS 3 METHODES RESO SE ITERATIVES**

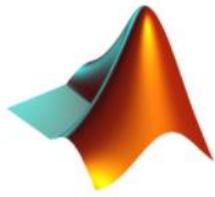
Document PDF

 **COURS 3 CONVERGENCE DES METHODES ITERATIVES**

Document PDF

 **Chapitre3-Méthodes itératives pour la résolution d'un système d'équations linéaires**

Document PDF



 [Glossaire des commandes Matlab](#)

وثيقة PDF

 [Premiers pas en Matlab](#)


وثيقة PDF

 [Fiche de TP N01](#)

وثيقة PDF

 [Solution fiche de TP N01](#)

وثيقة PDF

 [Fiche de TP N02](#)

وثيقة PDF

 [Solution fiche de TP N02](#)

وثيقة PDF

 [Fiche de TP N03](#)

وثيقة PDF

 [Solution fiche de TP N03](#)

وثيقة PDF