

MEMOIRE DE FIN D'ETUDE

Présenté pour l'obtention du Diplôme de **MASTER**

Domaine : Mathématiques et Informatique

Filière: Mathématiques

Option : mathématiques fondamentales et appliquées

Par

GHEZEROUM LEMYA

Sujet

**Etude comparatives des méthodes itératives
stationnaires et non stationnaires pour la
résolution des systèmes d'équations linéaires**

Date de soutenance : 28 Mai 2017

Devant le jury :

Mr.BENSALOUA Chenitti	MCB. Univ de M'sila	Président
Mr.LAKHEHALI Belkacem.	MCB. Univ de M'sila	Rapporteur
Mr.GAGUI Bachir.	MCB. Univ de M'sila	Examineur

Promotion : 2016 / 2017



Table des matières

Introduction	3
1 Notions de base	5
1.1 Quelques définitions et notions sur les matrices	6
1.2 Généralité sur les systèmes linéaires	8
1.2.1 Classification des systèmes linéaires	9
1.2.2 L'existence et l'unicité de la solution	9
1.2.3 Classification des méthodes des résolutions	10
1.3 Les normes	12
1.3.1 Normes vectorielles	12
1.3.2 Normes matricielles,	12
1.4 Conditionnement d'une matrice	14
2 Résolutions des systèmes linéaires par méthodes itératives	19
2.1 Comment construire une méthode itérative?	20
2.2 Etude de la convergence des méthodes itératives	24
2.2.1 Conditions des la convergence des méthodes itératives	25
2.3 Méthodes itératives stationnaires	28
2.3.1 Méthode de Jacobi :	28
2.3.2 Méthodes de Gauss-Seidel	31
2.3.3 Méthode de relaxation	31

2.3.4	Résultats de convergence de Jacobi et Gauss-Seidel	33
2.3.5	Résultats de convergence pour la méthode de relaxation	34
2.4	Méthodes itératives non stationnaire	36
2.4.1	Méthode du Gradient	39
2.4.2	Méthode du Gradient à pas fixe	39
2.4.3	Méthode de la plus profonde descente	40
2.4.4	Méthode du gradient conjugué	41
2.4.5	Méthode du gradient conjugué préconditionné	43
3	Application numérique par MATLAB	45
3.1	Exemple 1	46
3.2	Exemple 2	58
3.3	Exemple 3	62
	Bibliographie	66

Table des matières

Introduction

L'analyse matricielle étudie deux problèmes fondamentaux : l'inversion de matrices ou la résolution de systèmes linéaires qui fait l'objet du présent mémoire.

Dans les applications mathématiques, beaucoup des matrices particulières creuses, tri-diagonales, symétriques, définies positives qui proviennent de la formulation de problèmes discrétisée par différentes méthodes : Différences finies, éléments finis et méthodes spectrale. Tous les problèmes mathématiques nous menent, après discrétisation des systèmes linéaires $Ax = b$.

En analyse numérique, une méthode itérative est un procédé algorithmique utilisé pour résoudre un problème, par exemple la recherche d'une solution d'un système d'équations ou d'un problème d'optimisation. En débutant par le choix d'un point initial considéré comme une première ébauche de solution, la méthode procède par itérations au cours desquelles elle détermine une succession de solutions approximatives raffinées qui se rapprochent graduellement de la solution cherchée. Les points générés sont appelés des itérés.

Les méthodes itératives contrastent avec les méthodes directes qui résolvent le problème en une seule étape (par exemple la solution d'un système linéaire $Ax = b$ obtenue en calculant la matrice inverse de A).

Les méthodes itératives se substituent avantageusement aux autres lorsque :

- celles-ci sont inapplicables, coûteuses ou simplement inconnues ;
- le problème est mal conditionné ou comprend un grand nombre de variables, car les

solutions successives limitent la propagation des erreurs.

Par contre, la question de la vitesse de convergence (ou encore d'une éventuelle divergence) reste cruciale : c'est l'objet d'un vaste champ d'investigations de l'analyse numérique.

Le but de ce mémoire est de présenter plusieurs méthodes de résolution des systèmes linéaires des équations algébriques et les comparer entre eux.

On présente dans le premier chapitre : quelques définitions et notions sur les matrices, une généralité sur les systèmes linéaires (l'existence, l'unicité et classes des méthodes de résolutions), les normes (vectorielles et matricielles) et enfin, le conditionnement.

Dans le deuxième chapitre : on parle d'une classe importante de résolution de systèmes linéaires est celle des méthodes itératives et quelques conditions de convergence, et après on expose ces deux classes : stationnaires et non stationnaires.

Enfin, dans le troisième chapitre : on donne une comparaison entre quelques méthodes itératives pour la résolution des systèmes linéaires.

Chapitre 1

Notions de base

Le premier chapitre est constitué des rappelles et des concepts de base, à ce niveau nous rappelons quelques définitions et notions sur les matrices, une généralité sur les systèmes linéaires (l'existence, l'unicité et classes des méthodes de résolutions), les normes (vectorielles et matricielles) et enfin, le conditionnement.

1.1 Quelques définitions et notions sur les matrices

Définition 1.1.1 Soient m et n deux entiers positifs. On appelle matrice à m lignes et n colonnes, ou matrice $m \times n$, ou matrice (m, n) , à coefficients dans k , un ensemble de $m \times n$ scalaires $a_{ij} \in K$, avec $i = 1, \dots, m$ et $j = 1, \dots, n$, représentés dans le tableau rectangulaire suivant :

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdot & \dots & a_{1n} \\ a_{21} & a_{22} & \cdot & \cdot & a_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \vdots & \cdot & \cdot & \cdot & 0 \\ a_{m1} & \cdot a_{m2} \cdot & \cdot & \cdot & a_{mn} \end{pmatrix}$$

Quand $K = \mathbb{R}$ ou $K = \mathbb{C}$, on écrit respectivement $A \in \mathbb{R}^{m \times n}$ ou $A \in \mathbb{C}^{m \times n}$.

Nous utiliserons l'abréviation $A = (a_{ij})$ avec $i=1, \dots, m$ et $j = 1, \dots, n$. l'entier i est appelé indice de ligne, et l'entier j est appelé indice de colonne.

Soit $A = (a_{ij})_{1 \leq i, j \leq n}$ une matrice carrée de taille $n \times n$

Définition 1.1.2 On dit que A est **symétrique** si $A^t = A$ i.e : $a_{ij} = a_{ji}$ pour tout $1 \leq i, j \leq n$.

Définition 1.1.3 On dit que A est **symétrique défini positif** si elle est symétrique et pour tout $x \in \mathbb{R}^n, x \neq 0$ on a :

$$\langle Ax, x \rangle = x^t Ax = \sum_{1 \leq i, j \leq n} a_{ij} x_i x_j > 0.$$

Définition 1.1.4 On dit que A est **inversible** ou **régulière** s'il existe une matrice carrée A^{-1} (appelée matrice inverse) telle que :

$$A \times A^{-1} = A^{-1} \times A = I.$$

et singulière ou non inversible dans le cas contraire.

Autrement dit : le déterminant de A est non nul i.e $\det(A) \neq 0$.

Définition 1.1.5 On dit que A est **triangulaire inférieure** si $a_{ij} = 0$ pour tout $i < j$

$$A = \begin{pmatrix} a_{11} & 0 & 0 & \dots & 0 \\ a_{21} & a_{22} & 0 & \dots & \vdots \\ \dots & \dots & \dots & \dots & \dots \\ \vdots & \dots & \dots & \dots & 0 \\ a_{n1} & \dots & \dots & \dots & a_{nn} \end{pmatrix}$$

Définition 1.1.6 On dit que A est **triangulaire supérieure** si $a_{ij} = 0$ pour tout $i > j$.

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & \dots & a_{1n} \\ 0 & a_{22} & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & a_{n-1,n} \\ 0 & \dots & \dots & 0 & a_{nn} \end{pmatrix}$$

Définition 1.1.7 On dit que A est **tridiagonale** si $a_{ij} = 0$ pour tout i, j tels que $|i - j| \geq 2$ i.e si les seuls coefficients non nuls sont les coefficients diagonaux, ceux juste au-dessus de la diagonale, ceux juste au-dessous de la diagonale.

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & \dots & 0 \\ a_{21} & a_{22} & a_{23} & \dots & \vdots \\ 0 & \dots & \dots & \dots & \dots \\ \vdots & \dots & \dots & \dots & a_{n-1,n} \\ 0 & \dots & 0 & a_{n,n-1} & a_{nn} \end{pmatrix}$$

Définition 1.1.8 Le **rayon spectral** est le plus grand module des valeurs propres de A , noté $\rho(A)$.

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

Définition 1.1.9 une matrice $A \in \mathbb{R}^{n \times n}$ est dit à **diagonale dominante par lignes** si

$$|a_{ii}| \geq \sum_{j=1, j \neq i}^n |a_{ij}|, \text{ avec } i = 1, \dots, n.$$

Tandis qu'elle est dit à **diagonale dominante par colonnes** si

$$|a_{ii}| \geq \sum_{j=1, j \neq i}^n |a_{ji}|, \text{ avec } i = 1, \dots, n.$$

Si les inégalités ci-dessus sont **stricts**, A est dit à diagonale dominante **stricte** par lignes (ou par colonnes respectivement)

Remarque 1.1.1 une matrice à diagonale dominante stricte qui est symétrique avec des termes diagonaux strictement positifs est également définie positive.

Définition 1.1.10 Soit $A \in \mathbb{C}^{m \times n}$; La matrice $B = A^*$ est appelée **adjointe** (ou **transposée conjuguée**) de A si $b_{ij} = \bar{a}_{ji}$, ou \bar{a}_{ji} est le complexe conjugué de a_{ji} .

Définition 1.1.11 une matrice $A \in \mathbb{C}^{m \times n}$ est dit **hermitienne** ou **outoadjointe** si $A^t = A^*$, c'est-à-dire si $A^* = A$. Elle est dit **unitaire** si $A^*A = AA^* = I$. Enfin, si $AA^* = A^*A$, A est dit **normale**.

1.2 Généralité sur les systèmes linéaires

Définition 1.2.1 On appelle système de n équations linéaire à p inconnues (x_1, x_2, \dots, x_p) le système :

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1p}x_p = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2p}x_p = b_2 \\ \dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{np}x_p = b_n \end{cases}$$

Un système d'équations linéaires peut aussi s'écrire sous la forme matricielle $Ax = b$

ou :

A est un matrice de taille $n \times p$.

x est un vecteur de taille p .

b est un vecteur de taille n .

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdot & \dots & a_{1p} \\ a_{21} & a_{22} & \dots & \dots & a_{2p} \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & \dots & a_{np} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_p \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \cdot \\ \cdot \\ b_p \end{pmatrix}$$

1.2.1 Classification des systèmes linéaires

Il y'a 3 cas de système $Ax = b$:

1. Si $n = p$: alors la matrice A est carrée et le système $Ax = b$ admet une solution unique $x = A^{-1}b$ si A inversible.
2. Si $n > p$: alors le nombre d'équations supérieure aux nombres d'inconnues. Dans ce cas . En générale le système $Ax = b$ n'aura pas de solutions
3. Si $n < p$: alors le nombre d'inconnues supérieure aux nombres d'équations. Dans ce cas il existe une infinité de solutions.

Remarque 1.2.1 *Nous étudierons un seul cas où $n = p$ (A est une matrice carrée).*

1.2.2 L'existence et l'unicité de la solution

Dans ce cas, on est assuré de l'existence et l'unicité de la solution du système $Ax = b$ si l'une des conditions équivalentes suivantes est vérifier :

1. A est inversible ;
2. $Rg(A) = n$;

3. Le système homogène $Ax = 0$ admet seulement la solution nulle.

Théorème 1.2.1 *Si A est inversible le système $Ax = b$ possède une solution unique. Si A est singulière alors le système $Ax = b$ possède soit une infinité de solutions soit aucune solution*

Théorème 1.2.2 *Les affirmations suivantes sont équivalentes pour une matrice carrée.*

1. A est inversible ;
2. $\det(A) \neq 0$;
3. $\text{Ker}(A) = \{0\}$;
4. $\text{Rg}(A) = n$;
5. Les vecteurs colonnes ou lignes de A sont linéairement indépendants.

Remarque 1.2.2 *Si la taille de la matrice est assez grand, le calcul de déterminant de A sera très difficile. Dans ce cas, nous avons recourt à ce théorème.*

Théorème 1.2.3 *Soit $A = (a_{ij})_{1 \leq i, j \leq n}$ une matrice carrée d'ordre n . Si A est symétrique définie positive alors le système $Ax = b$ admet une solution unique.*

Théorème 1.2.4 *soit $A \in \mathbb{R}^{n \times n}$ une matrice symétrique. Alors A est définie positive si et seulement si une des propriétés suivantes est satisfaite :*

1. $(Ax, x) > 0 \forall x \neq 0$ avec $x \in \mathbb{R}^n$;
2. Les valeurs propres des sous - matrices principales de A sont toutes positives ;
3. Les mineures principaux dominants de A sont tous positifs (critère de Sylvestre) ;
4. Il existe une matrice inversible H telle que $A = HH^t$;

1.2.3 Classification des méthodes des résolutions

Les méthodes de résolutions des systèmes linéaires se classent en trois grands catégories :

1. Les méthodes directes : (Gauss, Factorisation LU et Cholesky)
2. Les méthodes itératives stationnaires (Jacobi, Gauss-seidel et Relaxation)
3. Les méthodes non stationnaires (projectives) : (Gradient, La plus profonde descente, Gradient conjugué et Gradient conjugué préconditionné)

1.3 Les normes

1.3.1 Normes vectorielles

Soit V un espace vectoriel sur K . on dit qu'une application $\|\cdot\|$ de V dans \mathbb{R} est une norme sur V si :

1. $\|v\| \geq 0 \forall v \in V$ et $\|v\| = 0$ si et seulement si $v = 0$;
2. $\|\alpha v\| = |\alpha| \|v\| \forall \alpha \in K, \forall v \in V$ (propriété d'homogénéité);
3. $\|v + w\| \leq \|v\| + \|w\| \forall v, w \in V$ (inégalité triangulaire), où $|\alpha|$ désigne la valeur absolue (resp le module) de α si $K = \mathbb{R}$ (resp $K = \mathbb{C}$). voir [2 p432]

On définit la p -norme (ou norme de Hölder) par :

$$\|x\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{1/p} \quad \text{pour } 1 < p < \infty,$$

où x_i sont les composantes du vecteur x .

Les normes les plus usuelles sont définies par les relations ($p = 1, 2, \infty$)

$$\|x\|_1 = \sum_{i=1}^n |x_i|$$

$$\|x\|_2 = \sqrt{\sum_{i=1}^n |x_i|^2}$$

$$\|x\|_\infty = \max_{1 \leq i \leq n} |x_i|$$

1.3.2 Normes matricielles,

Définition 1.3.1 une norme matricielle est une application $\|\cdot\| : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}$ telle que

1. $\|A\| \geq 0 \forall A \in \mathbb{R}^{m \times n}$ et $\|A\| = 0$ si et seulement si $A = 0$;
2. $\|\alpha A\| = |\alpha| \|A\| \forall \alpha \in \mathbb{R}, \forall A \in \mathbb{R}^{m \times n}$ (propriété d'homogénéité);

3. $\|A + B\| \leq \|A\| + \|B\| \quad \forall A, B \in \mathbb{R}^{m \times n}$ (inégalité triangulaire),

voir [2 p 438].

Définition 1.3.2 On dit que une norme matricielle $\|\cdot\|$ est compatible avec une norme vectorielle $\|\cdot\|$ si :

$$\|Ax\| \leq \|A\| \cdot \|x\| \quad \forall x \in \mathbb{R}^n$$

Définition 1.3.3 Soit $\|\cdot\|$ une norme vectorielle. La fonction

$$\|A\| = \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|}$$

est une norme matricielle. on l'appelle norme matricielle subordonnée associée à la norme vectorielle $\|\cdot\|$. On l'appelle aussi parfois norme matricielle naturel, ou encore norme matricielle induite par la norme vectorielle $\|\cdot\|$.

Remarque 1.3.1 On dit que : les p -normes sont définies comme suite :

$$\|A\|_p = \sup_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p}$$

La 1-norme et la norme infinie se calculent facilement :

$$\|A\|_1 = \max_{j=1, \dots, n} \sum_{i=1}^m |a_{ij}|, \quad (\text{somme des colonnes}).$$

$$\|A\|_\infty = \max_{i=1, \dots, m} \sum_{j=1}^n |a_{ij}|, \quad (\text{somme des lignes}).$$

$$\|A\|_2 = \sqrt{\rho(A^t, A)}.$$

On a de plus $\|A\|_1 = \|A^t\|_\infty$ et si A est autoadjointe ou symétrique réelle, $\|A\|_1 = \|A\|_\infty$.

Théorème 1.3.1 Soit $\|\cdot\|$ une norme matricielle subordonnée à une norme vectorielle $\|\cdot\|$. Alors :

1. $\|Ax\| \leq \|A\| \cdot \|x\|$, i.e. $\|\cdot\|$ est une norme compatible avec $\|\cdot\|$;
2. $\|I\| = 1$;
3. $\|A + B\| \leq \|A\| + \|B\|$, i.e. $\|\cdot\|$ est une sous-norme multiplicative.

1.4 Conditionnement d'une matrice

Le conditionnement mesure l'influence des erreurs d'arrondi sur la solution d'un problème donné. Il est mis en évidence par une légère perturbation des données initiales. voir [4 p 27]

Considérons le système linéaire $Ax = b$ suivant :

$$\begin{pmatrix} 23 & 9 & 12 \\ 12 & 10 & 1 \\ 14 & -12 & 25 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 44 \\ 23 \\ 27 \end{pmatrix} \quad (\text{P})$$

qui admet la solution $(1, 1, 1)$. (la matrice A est inversible).

Considérons le problème suivant dans lequel le vecteur b est légèrement perturbé

$$\begin{pmatrix} 23 & 9 & 12 \\ 12 & 10 & 1 \\ 14 & -12 & 25 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 44,44 \\ 23,77 \\ 27,73 \end{pmatrix} \quad (\text{P}_1)$$

La solution du système perturbé (P_1) est $(6,895, 5,565, 40)$.

Considérons le problème suivant dans laquelle la matrice A perturbée

$$\begin{pmatrix} 23,23 & 9,09 & 12,12 \\ 12,12 & 9,9 & 1,01 \\ 14,14 & -11,88 & 25,25 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 44 \\ 23 \\ 27 \end{pmatrix} \quad (\text{P}_2)$$

La solution du système perturbé (P_2) est (6, 23 4, 73 4, 69).

On remarque que : malgré que les systèmes (P_1) et (P) (ou (P_2) et (P)) soient voisins, leurs solutions sont très différents. (i.e. Une erreur de 1/100 sur les données provoque une grande variation sur la solution).

On étudier l'effet dans la solution x , si l'élément du système linéaire $Ax = b$ est perturbée.

soit A une matrice inversible, $b \neq 0$ un vecteur non nul, x la solution du système linéaire $Ax = b$ et $\|\cdot\|$ est une norme subordonnée.

1. Si $x + \delta x$ la solution du système perturbé $A(x + \delta x) = b + \delta b$ alors

$$\frac{\|\delta x\|}{\|x\|} \leq \|A\| \cdot \|A^{-1}\| \frac{\|\delta b\|}{\|b\|}. \quad (1)$$

2. Si $x + \delta x$ la solution du système perturbé $(A + \delta A)(x + \delta x) = b$ alors

$$\frac{\|\delta x\|}{\|x + \delta x\|} \leq \|A\| \cdot \|A^{-1}\| \frac{\|\delta A\|}{\|A\|}. \quad (2)$$

3. Si $x + \delta x$ la solution du système perturbé $(A + \delta A)(x + \delta x) = b + \delta b$ alors

$$\frac{\|\delta x\|}{\|x\|} \leq \|A\| \cdot \|A^{-1}\| \left(\frac{\|\delta b\|}{\|A\| \|x + \delta x\|} + \frac{\|\delta A\|}{\|A\|} \right). \quad (3)$$

Preuve.

- (a) on démontre l'inégalité (1) :

ona

$$A(x + \delta x) = Ax + A(\delta x) = b + A(\delta x) = b + \delta b,$$

donc

$$A(\delta x) = \delta b$$

Alors, $\sigma x = A^{-1}\delta b$, donc

$$\|\sigma x\| \leq \|A^{-1}\| \|\sigma b\| \quad (*)$$

Pour $Ax = b$, on trouve

$$\|b\| \leq \|A\| \|x\| \quad (**)$$

On multiplier les inégalités (*) et (**):

$$\|\sigma x\| \|b\| \leq \|A\| \|A^{-1}\| \|\sigma b\| \|x\|,$$

alors

$$\frac{\|\sigma x\|}{\|x\|} \leq \|A\| \cdot \|A^{-1}\| \frac{\|\sigma b\|}{\|b\|}$$

(b) on démontre l'inégalité (2) :

ona

$$(A + \sigma A)(x + \sigma x) = Ax + A(\sigma x) + \sigma A(x + \sigma x) = b.$$

alors

$$A(\sigma x) + \sigma A(x + \sigma x) = 0$$

On multiplier l'équation (*) par A^{-1} , on obtient

$$\sigma x = -A^{-1}\sigma A(x + \sigma x),$$

et alors

$$\|\sigma x\| \leq \|A^{-1}\| \|\sigma A\| \|x + \sigma x\|,$$

et

$$\frac{\|\sigma x\|}{\|x + \sigma x\|} \leq \|A^{-1}\| \|\sigma A\|.$$

On multiplier la partie à gauche par $\|A\| / \|A\|$, on obtient

$$\frac{\|\sigma x\|}{\|x + \sigma x\|} \leq \|A\| \cdot \|A^{-1}\| \frac{\|\sigma A\|}{\|A\|}.$$

(c) On démontre l'inégalité (3) :

On a

$$Ax = b \text{ et } (A + \sigma A)(x + \sigma x) = b + \sigma b,$$

alors

$$A(\sigma x) + \sigma A(x + \sigma x) = \sigma b.$$

On le multiplie par A^{-1} , on trouve

$$\sigma x = A^{-1}(\sigma b - \sigma A(x + \sigma x)),$$

Alors

$$\|\sigma x\| \leq \|A^{-1}\| \|\sigma b - \sigma A(x + \sigma x)\|.$$

On applique l'inégalité triangulaire pour $\|\sigma b - \sigma A(x + \sigma x)\|$,

$$\|\sigma x\| \leq \|A^{-1}\| (\|\sigma b\| + \|\sigma A\| \|(x + \sigma x)\|).$$

On divise les deux parties de l'inégalité précédente par $\|(x + \sigma x)\|$ et multiplie la partie droite par $\|A\| / \|A\|$, on obtient

$$\frac{\|\sigma x\|}{\|x\|} \leq \|A\| \cdot \|A^{-1}\| \left(\frac{\|\sigma b\|}{\|A\| \|x + \sigma x\|} + \frac{\|\sigma A\|}{\|A\|} \right).$$

■

Définition 1.4.1 On appelle conditionnement d'une matrice A , noter $K(A)$, le nombre

$$k(A) = \|A\| \cdot \|A^{-1}\|$$

Le conditionnement le plus utilisés en pratique sont : $\text{cond}_p(A) = \|A\|_p \cdot \|A^{-1}\|_p$ pour $p = 1, 2, \infty$, où les normes $\|A\|_p$ sont déjà définies.

Proposition 1.4.1 Pour une matrice carrée A d'ordre n ; le conditionnement a les propriétés suivantes :

1. $\text{cond}(I) = 1$.
2. $\text{cond}(A) = \text{cond}(A^{-1}) \geq 1$, et $\text{cond}(\alpha A) = \text{cond}(A) \quad \forall \alpha \neq 0$.
3. $\text{cond}_2(A) \geq 0$.
4. pour une matrice quelconque, $\text{cond}_2(A) = \frac{\mu_n(A)}{\mu_1(A)}$, où $\mu_1(A)$, $\mu_n(A)$ sont respectivement la plus petite et la plus grande valeur singulière.
5. pour une matrice **normal**, $\text{cond}_2(A) = \frac{|\lambda_n(A)|}{|\lambda_1(A)|}$, où $|\lambda_1(A)|$, $|\lambda_n(A)|$ sont respectivement la plus petite et la plus grande valeur propre en module.
6. pour toute matrice **unitaire** U , $\text{cond}_2(U) = 1$.
7. pour toute matrice **unitaire** U , $\text{cond}_2(AU) = \text{cond}_2(UA) = \text{cond}_2(A)$.

Chapitre 2

Résolutions des systèmes linéaires par méthodes itératives

L'objet de ce chapitre est d'étudier une classe important de méthode de résolution des systèmes linéaires est celle des méthodes itérative stationnaires et non stationnaires.

Définition 2.0.2 *Les méthodes itératives sont particulièrement intéressantes pour les très grandes matrices ou matrices creuse. L'idée principale dans les méthodes itératives pour la résolution de système $Ax = b$ est de construire une suite $(x^n)_n$ telle que cette suite doit converge vers la solution exacte x du système linéaire $Ax = b$.*

2.1 Comment construire une méthode itérative ?

Une théqniqne générale pour construire une méthode itérative est basée sur la décomposition de la matrice A sous la forme :

$$A = M - (M - A) = M - N \quad (2.1.1)$$

ou M est une matrice inversible (appelée préconditionnement de A)

Alors, le système

$$Ax = b \quad (2.1.2)$$

mis sous la forme

$$Mx = Nx + b \quad (2.1.3)$$

Lorsque la matrice M est inversible ,

$$x = (M^{-1}N)x + M^{-1}b \quad (2.1.4)$$

Remarquent que cette équation est équation de la forme $x = f(x)$. Par conséquent, les méthodes itératives sont des méthodes de point fixe. La détermination du point fixe repose sur la détermination de l'équation

$$x_{k+1} = (M^{-1}N)x_k + M^{-1}b \quad (2.1.5)$$

en notons x_k le vecteur de composantes $x_k = (x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)})$.

L'algorithme est initialisée par un vecteur arbitraire $x_0 = (x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)})$. et s'arrête quand :

$$\forall i \in \mathbb{N}, \quad \left| x_i^{(k)} - x_i^{(k-1)} \right| < \varepsilon \text{ pour un } \varepsilon \text{ donné}$$

En posant $B = M^{-1}N$, et $c = M^{-1}b$.

Alors la méthode itérative s'écrit sous la forme :

$$\begin{cases} x_0 \text{ donnée sur } \mathbb{R}^n. \\ Mx_{k+1} = Nx_k + c, \quad \forall k \geq 1 \end{cases}$$

Où B désigne une matrice carrée appelée matrice d'itération et c un vecteur dépendant de b (le second membre du système à résoudre).

On peut définir la méthode itérative correspondante :

$$M(x^{(k+1)} - x^{(k)}) = r^{(k)} \quad k \geq 0$$

Où

$$r^{(k)} = b - Ax^{(k)} \tag{2.1.6}$$

Désigne le résidu à l'itération k .

On peut généraliser cette méthode de la manière suivante ,voir [6 p 161]

$$M(x^{(k+1)} - x^{(k)}) = \alpha_k r^{(k)} \quad k \geq 0 \tag{2.1.7}$$

Où α_k , est un paramètre qui peut changer à chaque itération k et qui sera a priori utile pour améliorer les propriétés de convergence de la suite $(x^{(k)})$.

La méthode (2.1.7), appelée méthode de Richardson, conduite à chercher à chaque itération le résidu préconditionné $Z^{(k)}$, c'est à dire la solution du système linéaire :

$$MZ^{(k)} = r^{(k)} \quad (2.1.8)$$

Alors,

$$Z^{(k)} = M^{-1}r^{(k)}$$

La nouvelle itérée est alors définie par :

$$x^{(k+1)} = x^{(k)} + \alpha_k Z^{(k)}$$

Ainsi, la matrice M doit être choisie de telle manière que le coût de la résolution de (2.1.8) soit assez faible (par exemple une matrice M diagonale, ou triangulaire vérifiant à ce critère)

Considérons à présent une méthode peut être mise sous la forme (2.1.7)

La méthode est dite stationnaire, quand $\alpha_k = \alpha$ (une constante donnée pour tout $k \geq 0$), non stationnaire (dynamique) quand α_k peut varier au cours d'itérations.

Remarque 2.1.1 Les méthodes itératives peuvent être définies par :

$$\begin{cases} x_0 \text{ donnée sur } \mathbb{R}^n. \\ Mx_{k+1} = Nx_k + c, \forall k \geq 1 \end{cases} \quad (2.1.9)$$

Si la suite de solutions approchées x_k converge vers une limite x quand k tend vers l'infini, alors, par passage à la limite dans la relation de récurrence (2.1.9), on obtient

$$(M - N)x = Ax = b.$$

Par conséquent, si la suite des solutions approchées converge, sa limite est forcément la solution du système linéaire.

D'un point de vue pratique, il faut savoir quand on peut arrêter les itérations, c'est-à-dire à quel moment $x^{(k)}$ est suffisamment proche de la solution inconnue x . Comme on ne connaît pas x , on ne peut pas décider d'arrêter le calcul dès que $\|x - x^{(k)}\| \leq \varepsilon$ ou ε

est la précision désirée.

Par contre on connaît Ax (qui vaut b), est un critère d'arrêt fréquemment utilisé est :

$$\|b - Ax_k\| \leq \varepsilon.$$

Cependant, si la norme de A^{-1} est grande ce critère peut être trompeur car :

$$\|x - x_k\| \leq \|A^{-1}\| \|b - Ax_k\| \leq \varepsilon \|A^{-1}\|$$

qui peut ne pas être petit.

Dans la pratique, lorsque nous voulons calculer $x^{(k+1)}$ à partir de $x^{(k)}$, nous commençons par calculer le vecteur $c = Nx + b$, puis nous résolvons le système $Mx = c$. de façon évidente, ce dernier système doit être plus facile à résoudre que le système $Ax = b$ pour que nous gagnions quelque chose. Nous ferons donc en sorte que la décomposition (2.1.1) donne lieu à une matrice M telle que le système $Mx = c$ soit facile à résoudre, ce qui est par exemple le cas si M est diagonale ou triangulaire. Pour ce faire écrivons la matrice A sous la forme

$$A = D - (E + F).$$

où D la matrice diagonale formée des seuls éléments diagonaux de A .

$$D = \begin{pmatrix} a_{11} & 0 & \dots & \dots & 0 \\ 0 & a_{22} & \cdot & \cdot & \cdot \\ \dots & \cdot & \cdot & \cdot & \\ \dots & \cdot & \cdot & \cdot & 0 \\ 0 & \cdot & \cdot & 0 & a_{nn} \end{pmatrix}$$

$(-E)$ la matrice formée des a_{ij} si $i > j$.

$$-E = \begin{pmatrix} 0 & 0 & \cdot & \dots & 0 \\ a_{21} & 0 & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{n,1} & \cdot & \cdot & a_{n,n-1} & 0 \end{pmatrix}$$

($-F$) la matrice formée des a_{ij} si $i < j$.

$$-F = \begin{pmatrix} 0 & a_{12} & \dots & \dots & a_{1n} \\ 0 & 0 & \dots & \dots & \dots \\ \cdot & \cdot & \cdot & \dots & \dots \\ \cdot & \cdot & \cdot & \cdot & a_{n-1,n} \\ 0 & \dots & \cdot & \cdot & 0 \end{pmatrix}$$

Schématiquement nous avons donc

$$A = \begin{pmatrix} \cdot & & & & \\ & \cdot & & -F & \\ & & \cdot D & & \\ & -E & & \cdot & \\ & & & & \cdot \end{pmatrix}$$

2.2 Etude de la convergence des méthodes itératives

Considérons, les méthodes itératives de la forme

$$\begin{cases} x_0 \text{ donné} \\ x^{(k+1)} = Bx^{(k)} + c, \quad k \geq 0. \end{cases} \quad (2.2.10)$$

ou B désigne une matrice carrée appelée matrice d'itération et c un vecteur dépendant de b (le seconde membre du système à résoudre).

Théorème 2.2.1 *On dit qu'une méthode itérative est convergente si, quel que soit le choix du vecteur initial $\mathbf{x}_0 \in \mathbb{R}^n$, la suite de solutions approchées x_k converge vers la solution exacte x .*

2.2.1 Conditions de la convergence des méthodes itératives

Théorème 2.2.2 *voir [3 p 474]. Si la matrice d'itération $x^{(k+1)} = Bx^{(k)} + c$ vérifie que : $\|B\| < 1$ pour quelques normes subordonnées, alors l'itération $x^{(k+1)} = Bx^{(k)} + c$ converge pour quel que soit le choix du vecteur initial $\mathbf{x}_0 \in \mathbb{R}^n$.*

Preuve. voir [3 p 474]. Supposons $\|B\| < 1$ pour quelque norme subordonnée, alors

$$\|B^k\| \leq \|B\|^k \quad \text{et} \quad \|B\|^k \rightarrow 0 \text{ lorsque } k \rightarrow \infty$$

et

$$B^k \rightarrow 0 \text{ lorsque } k \rightarrow \infty$$

maintenant, on considère l'itération :

$$x^{(k+1)} = Bx^{(k)} + c \tag{1}$$

Si x la solution exact :

$$x = Bx + c \tag{2}$$

Par soustraction d'équation (1) de (2), on trouve :

$$x - x^{(k+1)} = B(x - x^{(k)}) \tag{3}$$

On définit l'erreur $e^{(k+1)} = x - x^{(k+1)}$ à l'itération k . L'équation (3) donne la relation

$$e^{(k+1)} = Be^{(k)} \tag{4}$$

On applique l'équation (3) :

$$\begin{aligned}
 e^{(1)} &= B e^{(0)}, \\
 e^{(2)} &= B e^{(1)} = B^2 e^{(0)} \\
 e^{(3)} &= B e^{(2)} = B^3 e^{(0)} \\
 &\dots \\
 &\dots \\
 e^{(k)} &= B^k e^{(0)}
 \end{aligned}$$

Si $B^k \rightarrow 0$ lorsque $k \rightarrow \infty$, alors $\lim_{k \rightarrow \infty} e^{(k)} = 0$, et la méthode est convergente. ■

Exemple 2.2.1 voir [3 p 437]

$$D = \begin{pmatrix} 5 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & -7 \end{pmatrix}, E = \begin{pmatrix} 0 & 0 & 0 \\ -1 & 0 & 0 \\ 1 & 6 & 0 \end{pmatrix}, \text{ et } F = \begin{pmatrix} 0 & -1 & 2 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

Méthode	B	$\ B\ _1$	$\ B\ _\infty$	$\ B\ _2$
Jacobi	$\begin{pmatrix} 0 & 0.2000 & -0.4000 \\ 0.2500 & 0 & -0.2500 \\ 0.1429 & 0.8571 & 0 \end{pmatrix}$	1.0571	1.0000	0.8997
Gauss-Seidel	$\begin{pmatrix} 0 & 0.2000 & -0.4000 \\ 0 & 0.0500 & -0.3500 \\ 0 & 0.0714 & 0.3571 \end{pmatrix}$	1.1071	0.6000	0.6692
SOR($\omega = 1.2$)	$\begin{pmatrix} -0.2000 & 0.2400 & -0.4800 \\ -0.0600 & -0.1280 & -0.4440 \\ -0.0690 & -0.0905 & -0.7390 \end{pmatrix}$	1.6630	0.9255	1.0063

La convergence des méthodes itératives

Remarque 2.2.1 La condition $\|B\| < 1$ pour quelques normes subordonnées assure la convergence. Mais il est possible que l'itérations converge si $\|B\| \geq 1$ pour Jacobi et Gauss-seidel. Alors, comme première vérification pour déterminer si la méthode converge, on calcule les normes subordonnées de B et on détermine si l'un à une valeur inférieure à 1. Si oui, alors, elle est converge, mais si les normes subordonnées sont tous supérieures ou égale à 1, la méthode peut converger. Donc on passe au deuxième condition qui depend du rayon spectral.

Théorème 2.2.3 voir[3 p475] La méthode itérative $x^{(k+1)} = Bx^{(k)} + c$ converge si et seulement si le rayon spectral de la matrice B vérifier

$$\rho(B) < 1$$

Dans la pratique le rayon spectrale est difficile à calculer (il faut calculer ses valeurs propres). C'est pourquoi on utilise d'autres conditions suffisantes de convergence comme indique ci-dessous.

Lemme 2.2.1 Soit A une matrice hermitienne, définie positive. Soit une décomposition régulière de A définie par $A = M - N$ avec M inversible. Alors la matrice $(M^* + N)$ est hermitienne. De plus, si $(M^* + N)$ est aussi définie positive, alors

$$\rho(M^{-1}N) < 1.$$

Proposition 2.2.1 Soit $A = M - N$ avec A et M symétriques définies positives. Si la matrice $2M - A$ est définie positive, alors la méthode (2.2.1) est convergent pour tout donnée initial $x^{(0)}$ et

$$\rho(B) = \|B\|_A = \|B\|_M < 1.$$

Proposition 2.2.2 Soit $A = M - N$ avec A symétrique définie positive. Si la matrice $M + M^t - A$ est définie positive, alors M est inversible, alors la méthode (2.2.1) est converge

de manière monotone pour la norme $\|\cdot\|_A$ et

$$\rho(B) \leq \|B\|_A < 1.$$

De plus, la convergence de la suite est monotone pour les normes $\|\cdot\|_M$ et $\|\cdot\|_A$. i.e. ($\|e^{(k+1)}\|_M \leq \|e^{(k)}\|_M$ et $\|e^{(k+1)}\|_A \leq \|e^{(k)}\|_A$, $k = 0, 1, \dots$).

2.3 Méthodes itératives stationnaires

2.3.1 Méthode de Jacobi :

L'idée de Jacobi : voir [3 p 369]

Pour la méthode de Jacobi, on écrit chaque inconnue en fonction des autres inconnues, nous illustrons cette idée par utilisée un système 4×4 .

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 = b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 = b_3 \\ a_{41}x_1 + a_{42}x_2 + a_{43}x_3 + a_{44}x_4 = b_4 \end{cases}$$

Alors

$$\begin{cases} x_1 = [b_1 - (a_{12}x_2 + a_{13}x_3 + a_{14}x_4)]/a_{11} \\ x_2 = [b_2 - (a_{21}x_1 + a_{23}x_3 + a_{24}x_4)]/a_{22} \\ x_3 = [b_3 - (a_{31}x_1 + a_{32}x_2 + a_{34}x_4)]/a_{33} \\ x_4 = [b_4 - (a_{41}x_1 + a_{42}x_2 + a_{43}x_3)]/a_{44} \end{cases}$$

On suppose une valeur initiale pour x_1, x_2, x_3 et x_4 . On l'insère dans la coté droite de l'égalité et calculer la coté seconde des valeurs approximés des inconnues. Les nouvelles

valeurs son substituée on la coté droite de l'égalité pour obtenu la troisième ensemble de la solution approchée, ...etc.

Pour le cas générale de n inconnues, L'itération est défini par :

$$x_i = 1/a_{ii} \left(b_i - \sum_{\substack{j=1 \\ i \neq j}}^n a_{ij}x_j \right), 1 \leq i \leq n$$

Alors, dans la méthode de Jacobi, et pour une donnée initiale x_0 , on calcule selon la formule :

$$x_i^{n+1} = 1/a_{ii} \left(b_i - \sum_{\substack{j=1 \\ i \neq j}}^n a_{ij}x_j^n \right), 1 \leq i \leq n \quad (*)$$

Cela revient à effectuer la décomposition de la matrice A suivant :

$$M = D, \text{ et } N = (D - A) = (E + F). \text{ on a bien } A = M - N$$

On note D la matrice formée des seuls éléments diagonaux de A .

$$D = \begin{pmatrix} a_{11} & 0 & \dots & \dots & 0 \\ 0 & a_{22} & & & \\ \dots & & \cdot & \cdot & \\ \dots & & & \cdot & 0 \\ 0 & & & 0 & a_{nn} \end{pmatrix}$$

$(-E)$ la matrice formée des a_{ij} si $i > j$.

$$-E = \begin{pmatrix} 0 & 0 & \cdot & \dots & 0 \\ a_{21} & 0 & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{n,1} & \cdot & \cdot & a_{n,n-1} & 0 \end{pmatrix}$$

($-F$) la matrice formée des a_{ij} si $i < j$.

$$-F = \begin{pmatrix} 0 & a_{12} & \dots & \dots & a_{1n} \\ 0 & 0 & \dots & \dots & \dots \\ \cdot & \cdot & \cdot & \dots & \dots \\ \cdot & \cdot & \cdot & \cdot & a_{n-1,n} \\ 0 & \dots & \cdot & \cdot & 0 \end{pmatrix}$$

de sorte que $A = D - (E + F)$.

la relation (*) permettent de calculer explicitement les composantes x_i^{n+1} , $1 \leq i \leq n$, du vecteur x^{n+1} à partir des composantes x_i^n , $1 \leq i \leq n$ du vecteur x^n .

Matrice d'itération de Jacobi

La relation (*) peut écrire sous forme matricielle

$$\begin{aligned} x^{(k+1)} &= B_J x^{(k)} + c. \\ B_J &= D^{-1}(E + F), c = D^{-1}. \end{aligned}$$

la matrice B_J est appelée : matrice d'itération de Jacobi.

2.3.2 Méthodes de Gauss-Seidel

La méthode de Gauss-Seidel c'est une amélioration de la méthode de Jacobi dans laquelle les valeurs calculées sont utilisées au fur et à mesure du calcul et non à l'issue d'une itération comme dans la méthode de Jacobi, et on améliore ainsi la vitesse de convergence.

L'itération est défini par :

$$x_i^{(n+1)} = 1/a_{ii} \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(n+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(n)} \right), 1 \leq i \leq n \quad (**)$$

Cela revient à effectuer la décomposition de la matrice A suivant :

$$M = D - E \quad \text{et} \quad N = F$$

Matrice d'itération de Gauss-Seidel

La relation (**) peut écrire sous forme matricielle

$$\begin{aligned} x^{(k+1)} &= B_{G-S}x^{(k)} + c. \\ B_{G-S} &= (D - E)^{-1}F, \quad c = (D - E)^{-1}b. \end{aligned}$$

La matrice B_{G-S} est appelée : matrice d'itération de Gauss-Seidel.

2.3.3 Méthode de relaxation

La convergence d'une méthode itérative ne dépend pas du choix du vecteur initial x_0 mais la rapidité de convergence en dépend. D'où l'idée d'introduire un facteur de relaxation ω non nul.

La méthode de relaxation a été introduite par D. Young dans la thèse de son doctorat en 1950, sous-titre "Résolution des EDP elliptique".

L'idée de Young :

$$Ax = b$$

$$(D - E - F)x = b$$

$$\left(\frac{D}{\omega} - \frac{D}{\omega} + D - E - F\right)x = b$$

$$\left[\left(\frac{D}{\omega} - E\right) - \left(\frac{D}{\omega} - D + F\right)\right]x = b$$

$$\left(\frac{D}{\omega} - E\right)x = \left(\frac{D}{\omega} - D + F\right)x + b$$

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

Définition 2.3.1 Soit $\omega \in \mathbb{R}^+$. On appelle méthode de relaxation, pour le paramètre ω , la méthode itérative associée à la décomposition :

$$M = \frac{1}{\omega} D - E, \quad N = \frac{1-\omega}{\omega} D + F$$

L'algorithme est fondé sur le calcul des itérées

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

Matrice d'itération de relaxation La relation (***) peut écrire sous forme matricielle

$$x^{(k+1)} = B_w x^{(k)} + c.$$

$$B_w = \left(\frac{1}{\omega}D - E\right)^{-1} \left(\frac{1-\omega}{\omega}D + F\right), \quad c = \left(\frac{1}{\omega}D - E\right)^{-1}.$$

Telle que B_w est appelée : matrice d'itération de relaxation.

Pour que la méthode de relaxation soit bien définie, il faut encore que la matrice D soit inversible.

Pour $\omega = 1$, on retrouve la méthode de Gauss-Seidel.

Lorsque $0 < \omega < 1$, on parle de sous relaxation.

Lorsque $1 < \omega < 2$, on parle de sur relaxation (SOR, Successive over Relaxation).

En général, il existe un paramètre ω_{opt} qui minimise le rayon spectral de la matrice d'itération $M^{-1}N$, et donc qui maximise la vitesse de convergence.

2.3.4 Résultats de convergence de Jacobi et Gauss-Seidel

voir [7 p123].

Théorème 2.3.1 *Si A est une matrice à diagonale dominante strict, les méthodes de Jacobi et de Gauss-Seidel sont convergentes.*

Théorème 2.3.2 *Si A et $2D - A$ sont symétriques définies positives, alors la méthode de Jacobi est convergent et*

$$\rho(B_J) = \|B_J\|_A = \|B_J\|_D.$$

Théorème 2.3.3 *Si A est symétrique définie positive, la méthode de Gauss-Seidel est converge de manière monotone pour la norme $\|\cdot\|_A$.*

Enfin, si A est tridiagonale symétrique définie positive, on peut montrer que la méthode de Jacobi est convergente et que

$$\rho(B_{GS}) = \rho^2(B_J).$$

Dans ce cas , la méthode de Gauss-Seidel converge plus rapidement que celle de Jacobi.

2.3.5 Résultats de convergence pour la méthode de relaxation

voir [7 p123].

Sans hypothèse particulière sur A , on peut déterminer les valeurs de ω pour lesquelles la méthode de SOR ne peut pas converger :

Théorème 2.3.4 *On a $\rho(B(\omega)) \geq |\omega - 1| \forall \omega \in \mathbb{R}$. la méthode SOR diverge donc si $\omega \leq 0$ ou $\omega \geq 2$.*

Preuve. si $\{\lambda_i\}$ désigne l'ensemble des valeurs propres de la matrice d'itération de SOR, alors

$$\left| \prod_{i=1}^n \lambda_i \right| = |\det[(1 - \omega)I + \omega D^{-1}F]| = |1 - \omega|^n.$$

Par conséquent, au moins une valeur propre λ_i est telle que $|\lambda_i| \geq |1 - \omega|$. Pour avoir convergence, il est donc nécessaire que $|1 - \omega| < 1$, c'est-à-dire $0 < \omega < 2$. ■

Si on suppose A symétrique définie positive, la condition nécessaire $0 < \omega < 2$ devient suffisante pour avoir convergence .On a en effet le résultat suivant :

Théorème 2.3.5 (Ostrowski-Reich) *Si A symétrique définie positive, alors la méthode SOR converge si et seulement si $0 < \omega < 2$. De plus, sa convergence est monotone pour $\|\cdot\|_A$.*

Enfin, si A est à diagonale dominante stricte, SOR converge si $0 < \omega < 1$.

En général, il existe un paramètre ω_{opt} qui minimise le rayon spectral de la matrice d'itération $M^{-1}N$, et donc qui maximise la vitesse de convergence.

Lorsque A est une matrice tridiagonale par blocs dont les blocs diagonaux sont inversibles, alors la valeur optimal du facteur de relaxation est donnée par

$$\omega_{opt} = \frac{2}{1 + \sqrt{1 - \rho(B_J)^2}}$$

Exemple 2.3.1 voir[7 p122].

Le **but de cet exemple** est de démontrer que : pour des matrices générales, on ne peut pas tirer aucune conclusion sur la convergence des méthodes de Jacobi et de Gauss-Seidel. Considérons les systèmes linéaires 3×3 de la forme $A_i x = b_i$. On choisi b_i de manière à ce que la solution du système soit le vecteur unité, et les matrices A_i sont données par

$$A_1 = \begin{pmatrix} 3 & 0 & 4 \\ 7 & 4 & 2 \\ -1 & 1 & 2 \end{pmatrix}, \quad A_2 = \begin{pmatrix} -3 & 3 & -6 \\ -4 & 7 & -8 \\ 5 & 7 & -9 \end{pmatrix},$$

$$A_3 = \begin{pmatrix} 4 & 1 & 1 \\ 2 & -9 & 0 \\ 0 & -8 & -6 \end{pmatrix}, \quad A_4 = \begin{pmatrix} 7 & 6 & 9 \\ 4 & 5 & -4 \\ -7 & -3 & 8 \end{pmatrix}$$

On peut vérifier que la méthode de Jacobi ne converge pas pour A_1 ($\rho(B_J) = 1.33$), contrairement à celle de Gauss-Seidel. C'est exactement le contraire qui se produit pour A_2 ($\rho(B_{GS}) = 1.1$). La méthode de Jacobi converge plus lentement que celle de Gauss-Seidel pour la matrice A_3 ($\rho(B_J) = 0.44$) et ($\rho(B_{GS}) = 0.018$). La méthode de Jacobi converge plus rapide que celle de Gauss-Seidel pour la matrice A_4 ($\rho(B_J) = 0.64$) et ($\rho(B_{GS}) = 0.77$).

2.4 Méthodes itératives non stationnaire

Dans cette section, nous supposons que la matrice A symétrique définie positive et nous étudions un autre type des méthodes itératives pour résoudre numériquement $Ax = b$, le vecteur b étant donné.

Commençons par définir, pour tout N -vecteur y , la quantité

$$\Phi(y) = \frac{1}{2} \langle y, Ay \rangle - \langle b, y \rangle = \frac{1}{2} y^t Ay - b^t y. \quad (1)$$

Il est clair que $\Phi(y)$ est un nombre réel. En fait, Φ est une application de \mathbb{R}^n dans \mathbb{R} . Nous avons le résultat suivant :

Théorème 2.4.1 voir [5 p92]. Si A est une $n \times n$ matrice symétrique définie positive et si x est solution de $Ax = b$ alors, pour tout n -vecteur y différent de x on a :

$$\Phi(x) < \Phi(y)$$

Preuve. voir [5 p92]. Soit x tel que $Ax = b$ et soit y un vecteur différent de x . Par hypothèse, le vecteur $z = x - y$ est donc différent de zéro. Par définition de Φ nous avons :

$$\begin{aligned} \Phi(y) &= \Phi(x - z) = \frac{1}{2} (x - z)^t A (x - z) - b^t (x - z) \\ &= \frac{1}{2} x^t Ax - b^t x - \frac{1}{2} z^t Ax - \frac{1}{2} x^t Az + \frac{1}{2} z^t Az + b^t z \end{aligned}$$

En tenant compte de la symétrie de A , nous avons $x^t Az = z^t Ax$ ainsi que $b^t z = z^t b$, si bien que

$$\begin{aligned} \Phi(y) &= \Phi(x) - z^t Ax + z^t b + \frac{1}{2} z^t Az \\ &= \Phi(x) - z^t (Ax - b) + \frac{1}{2} z^t Az \end{aligned}$$

et puisque $Ax = b$, nous obtenons

$$\Phi(y) = \Phi(x) + \frac{1}{2}z^tAz$$

La matrice A étant symétrique définie positive et le vecteur z étant non nul, nous avons $z^tAz > 0$ et ainsi $\Phi(y) > \Phi(x)$. ■

Le principe :

Puisque la solution du système $Ax = b$ réalise le minimum de Φ . Nous allons construire une méthode itérative qui cherche, à chaque itération, à diminuer Φ .

Supposons donc que l'on a obtenue $x^{(n)}$ et calculons $x^{(n+1)}$ de sorte à ce que :

$$\Phi(x^{(n+1)}) < \Phi(x^{(n)})$$

Ce qui revient à déterminer des directions de déplacement qui permettent de se rapprocher le plus possible de la solution.

Une idée légitime consiste à choisir un vecteur $w^{(n+1)}$ non nul et à poser

$$x^{(k+1)} = x^{(k)} + \alpha^{(k+1)}w^{(k+1)}. \quad (2)$$

Où α_k est un nombre réel qui minimise la quantité $f(\alpha)$ définie par :

$$f(\alpha) = \Phi(x^{(k)} + \alpha w^{(k+1)}). \quad (3)$$

Cette manière de calculer $x^{(n+1)}$ à partir de $x^{(n)}$ est appelée méthode de **descente**, le vecteur $w^{(k+1)}$ est appelé **direction de descente**.

Explicitons maintenant le calcul de α^{k+1} tel que $f(\alpha^{k+1}) \leq f(\alpha), \forall \alpha \in \mathbb{R}$. nous annulons la dérivée $f'(\alpha)$ En utilisant (1)

Nous avons

$$\Phi(y) = \frac{1}{2} \sum_{i,j=1}^N a_{ij}y_iy_j - \sum_{i=1}^N b_iy_i \quad (4)$$

et par suite, en tenant compte de la symétrie de A :

$$\frac{\partial}{\partial y_i} = \Phi(y) = \sum_{i,j=1}^N a_{ij}y_i - \sum_{i=1}^N b_i. \quad (5)$$

Il suffit d'utiliser la définition (3) pour montre que

$$\begin{aligned} f(\alpha) &= \sum_{i=1}^N w_i^{k+1} \frac{\partial}{\partial y_i} \Phi(x^{(k)} + \alpha w^{(k+1)}) \\ &= \sum_{i=1}^N w_i^{k+1} \left(\sum_{j=1}^N a_{ij} (x_j^k + \alpha w_j^{k+1} - b_i) \right) \\ &= (w^{k+1})^t (A(x^k + \alpha w^{k+1}) - b) \end{aligned} \quad (6)$$

Clairement, si nous voulons que α^{k+1} soit tel que $f'(\alpha^{k+1}) = 0$, nous déduisons de (5) que α^{k+1} est donner par :

$$\alpha^{k+1} = \frac{(w^{k+1})^t (b - Ax)}{(w^{k+1})^t Aw^{k+1}} \quad (7)$$

En définissant encore

$$r = b - Ax \quad (8)$$

qui es le **résidu** à l'étape n , la méthode de descente peut se résumer ainsi lorsqu'on veut calculer $x^{(k+1)}$ à partire de $x^{(k)}$:

- on choisit une direction de descente w^{k+1} ;
- on calcule

$$\alpha^{k+1} = \frac{(w^{k+1})^t r^{(k)}}{(w^{k+1})^t Aw^{k+1}} \quad (9)$$

- on calcule $x^{(k+1)} = x^{(k)} + \alpha^{(k+1)} w^{(k+1)}$.

Toute question est maintenant de bien choisir la direction de descente w^{k+1} de sorte que la méthode converge.

2.4.1 Méthode du Gradient

L'idée la plus naturelle est de prendre la direction de descente de pente maximal $\nabla\Phi(x^{(k)})$, c'est la méthode du **gradient** ou **méthode de la plus profonde descente**.

D'après (1)

$$\nabla\Phi(x^{(k)}) = Ax^{(k)} - b = -r^{(k)}.$$

La direction du Gradient de Φ coïncide donc avec le résidu et peut être immédiatement calculée en utilisant la valeur $x^{(k)}$. Ceci montre que la méthode du Gradient revient à se déplacer à chaque étape k le long de direction

$$p^{(k)} = r^{(k)} = -\nabla\Phi(x^{(k)}).$$

avec un paramètre α_k à déterminer.

Donc on a deux cas :

1. Le choix simple : α est constante, on dit qu'est une méthode à pas fixe.
2. Le choix le plus compliqué : α est variable, dans ce cas on dit qu'est une méthode à pas variable.

2.4.2 Méthode du Gradient à pas fixe

L'algorithme du Gradient à pas fixe est donnée par :

$$\left\{ \begin{array}{l} x^{(0)} \text{ donnée} \\ x^{(k+1)} = x^{(k)} - \alpha \nabla\Phi(x^{(k)}) \\ k = 1, 2, 3, \dots \end{array} \right.$$

Le réel α est un paramètre, qu'on doit le fixer par le lemme suivant :

Lemme 2.4.1 voir[6 p166]. Soit $A \in \mathbb{R}^{n \times n}$. pour toute matrice inversible $M \in \mathbb{R}^{n \times n}$.
La méthode de Gradient à pas fixe converge si et seulement si

$$|\lambda_i|^2 < \frac{2}{\alpha} \operatorname{Re} \lambda_i \quad \forall i = 1, \dots, n,$$

où les λ_i sont les valeurs propres de $M^{-1}A$.

Si ces dernière sont toutes réelles, alors la méthode converge si et seulement si

$$0 < \alpha \lambda_i < 2 \quad \forall i = 1, \dots, n.$$

Si la matrice M et A sont symétriques et définies positives, la méthode de Gradient à pas fixe converge pour ou $x^{(0)}$ si et seulement si $0 < \alpha < 2/\lambda_{\max}$, où $\lambda_{\max}(> 0)$ est la valeur propre maximale de $M^{-1}A$. De plus, le rayon spectrale $\rho(B_\alpha)$ de la matrice d'itération $B_\alpha = I - \alpha M^{-1}A$ est minimum quand $\alpha = \alpha_{opt}$, où $\alpha_{opt} = \frac{2}{\lambda_{\min} + \lambda_{\max}}$ où λ_{\min} étant la valeur propre minimale de $M^{-1}A$. Enfin, on a la résultat de convergence suivant

$$\|e^{(k)}\|_A = \left(\frac{k(M^{-1}A) - 1}{k(M^{-1}A) + 1} \right)^k \|e^{(0)}\|_A, \quad k \geq 0$$

où $\|v\|_A = \sqrt{v^t A v}$, $\forall v \in \mathbb{R}^n$, s'appelle norme de l'énergie associée à la matrice A .

2.4.3 Méthode de la plus profonde descente

La méthode de la plus profonde descente cherché à minimiser le résidu r_k . Elle se fonde sur l'algorithme suivant : on se donne un vecteur x_0 , puis on calcule successivement les quantités

$$\begin{cases} r_k = b - Ax_k \\ \alpha_k = \frac{\langle r_k, r_k \rangle}{\langle r_k, Ar_k \rangle} \\ x_{k+1} = x_k + \alpha_k r_k \end{cases}$$

ou $\langle r_k, r_k \rangle$ est le produit scalaire de r_k par lui-même. L'efficacité de la méthode

dépend du conditionnement de la matrice A .

Notons

$$\varepsilon(x_k) = \langle A(x_k - x), (x_k - x) \rangle^{1/2}$$

La norme "énergétique".

Soit A une matrice symétrique définie positive, et $k = \frac{\lambda_{\max}}{\lambda_{\min}}$ le conditionnement de la matrice A , k est le rapport de la plus grande valeur propre de A sur la plus petite.

La convergence de la méthode de la plus profonde descente est donnée par

$$\varepsilon(x_k) \leq \left(\frac{k-1}{k+1} \right)^k \varepsilon(x_0)$$

La méthode n'est pas toujours efficace : si la matrice A a un grand conditionnement, on voit dans l'expression précédente que si k est élevé, la norme énergétique n'évolue presque pas. Par conséquent, le vecteur résiduel $r^{(k)}$ ne change pas beaucoup d'une itération à l'autre, la convergence est très lente.

Pour éviter ce problème, Fox, Husky et Wilkinson ont proposé en 1949 de remplacer la minimisation de long du vecteur résiduel par une minimisation de long de la direction orthogonale : c'est la méthode des directions conjuguées. Hestenes et Stiefel ont montré qu'on pouvait choisir ces directions à chaque pas : c'est la méthode du gradient conjugué.

2.4.4 Méthode du gradient conjugué

La méthode du gradient conjugué est une amélioration de la méthode de la plus profonde descente, dans laquelle le calcul de $x_{k+1} = x_k + \alpha_{k+1} \rho_k$ se fait le long de nouvelles directions (ρ_0, \dots, ρ_k) .

On rappelle que la méthode itérative précédent i.e. La méthode du Gradient, la nouvelle itérée $x^{(k+1)}$ est obtenue en ajoutant à l'ancienne $x^{(k)}$ un vecteur appelé direction de descente, qui est le résidu.

On peut se demander s'il ne serait pas possible de construire d'autres directions de

descente, $p^{(k)}$, qui permettraient de converger plus vite.

Quand la matrice $A \in \mathbb{R}^{n \times n}$ est symétrique définie positive, la méthode du gradient conjugué (en abrégé GC) utilise une suite de directions de descente constituée par des vecteurs A -orthogonaux on dit aussi A -conjugués, c'est-à-dire vérifiant $\forall k \geq 1$,

$$(Ap^{(j)})^T p^{(j+1)} = 0, j = 0, 1, \dots, k.$$

L'algorithme

On suppose que A est une matrice symétrique, définie positive.

Les trois premières équations de l'algorithme correspondant à la minimisation de Φ sur l'espace $x_0 + \text{vect}(\rho_0, \dots, \rho_k)$, ou x_0 est un point arbitraire choisi comme point initial de l'algorithme.

Les deux dernières équations correspondent au calcul de la nouvelle direction.

Elle se fonde sur l'algorithme suivant :

On se donne un point x_0 de \mathbb{R}^n , $r_0 = b - Ax_0$ et $p_0 = r_0$ et on calcule les quantités suivants :

$$\left\{ \begin{array}{l} \alpha_{k+1} = \frac{\langle r_k, r_k \rangle}{\langle p_k, Ap_k \rangle} \\ x_{k+1} = x_k + \alpha_{k+1} p_k \\ r_{k+1} = r_k - \alpha_{k+1} Ap_k \\ \beta_{k+1} = \frac{\langle r_{k+1}, r_{k+1} \rangle}{\langle r_k, r_k \rangle} \\ p_{k+1} = r_{k+1} + \beta_{k+1} p_k \end{array} \right.$$

le paramètre α_k permet de minimiser l'erreur $\|e^{(k+1)}\|_A$ le long de la direction de descente $p^{(k)}$, et β_k est choisi pour que la nouvelle direction $p^{(k+1)}$ soit A -conjuguée avec $p^{(k)}$, c'est-à-dire $(Ap^{(k)})^T p^{(k+1)} = 0$.

Proposition 2.4.1 *Soit A une matrice symétrique définie positive. En arithmétique exacte, la méthode du gradient conjuguée pour résoudre un système $Ax = b$, converge en au plus n étapes.*

De plus, l'erreur $e^{(k)}$ à la k -ème itération (avec $k < n$) est orthogonal à $p^{(j)}$, pour $j = 0, \dots, k - 1$ et

$$\|e^{(k+1)}\|_A \leq \frac{2c^k}{1 + c^{2k}} \|e^{(0)}\|_A, \text{ avec } c = \frac{\sqrt{k(A)} - 1}{\sqrt{k(A)} + 1}.$$

Remarque 2.4.1 Ainsi, en l'absence d'erreur d'arrondi, on peut considérer La méthode du Gradient conjugué comme une méthode directe puisqu'elle fournit le résultat en un nombre fini d'étapes. Cependant, pour les matrices de grande taille, utilisé comme une méthode itérative, c'est-à-dire dont les itérations sont interrompues quand un estimateur de l'erreur devient inférieure à une tolérance donnée.

2.4.5 Méthode du gradient conjugué préconditionné

La méthode du gradient conjugué nécessite $n^3 + 5n^2 - 3n$ additions, $n^3 + 6n^2$ multiplications et $2n$ divisions.

Si la matrice A est mal conditionnée, la convergence de l'algorithme du gradient conjugué est lente. Dans ce cas, on cherchera à améliorer la vitesse de convergence ; c'est la méthode du gradient conjugué préconditionnée.

Si les valeurs propres de la matrice A sont dispersées, il faut procéder à un préconditionnement et remplacer le système $Ax = b$ par $MAx = Mb$ ou M est une matrice symétrique définie positive telle que $cond(MA) \ll cond(A)$.

On modifie l'algorithme précédent par les formules

$$\left\{ \begin{array}{l} r_0 = b - Ax_0 \\ p_0 = Mr_0 \\ \alpha_k = \frac{\langle r_k, Mr_k \rangle}{\langle p_k, Ap_k \rangle} \\ x_{k+1} = x_k + \alpha_k p_k \\ r_{k+1} = r_k - \alpha_k Ap_k \\ \beta_{k+1} = \frac{\langle r_{k+1}, Mr_{k+1} \rangle}{\langle r_k, Mr_k \rangle} \\ p_{k+1} = Mr_{k+1} + \beta_{k+1} p_k \end{array} \right.$$

Comment choisir un préconditionné est une question fort délicate qui à ce jour n'a pas de solution universelle.

En analyse numérique, il n'existe pas une meilleure stratégie de préconditionnement valable pour tout type de matrices, et leur efficacité varie fortement d'un système à autre.

Parmi les préconditionnements les plus importants :

1. Préconditionnement de Jacobi.
2. SSOR (symmetric successive over relaxation)
3. Méthode de cholesky incomplète : $P = LL^t$.
4. Préconditionnement par bande.

Chapitre 3

Application numérique par MATLAB

Dans ce chapitre on essaye de faire une comparaison entre quelques méthodes itératives stationnaires et non stationnaires à l'aide des exemples en matlab.

Commençons par l'étude des conditions pour assure la convergence de tous les méthodes :

-Conditions sur la matrice :

A Symétrique définie positive.

Le rayon spectrale inférieure de 1..

-Condition sur les méthodes :

Pour la méthode de relaxation : $0 < \omega < 2$.

3.1 Exemple 1

Soit le système $Ax = b$ telle que :

$$\begin{pmatrix} 0.2 & 0.1 & 1 & 1 & 0 \\ 0.1 & 4 & 1 & 1 & 1 \\ 1 & -1 & 60 & 0 & -2 \\ 1 & 1 & 0 & 8 & 4 \\ 0 & -1 & 2 & 4 & 700 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix}$$

* A est-elle Symétrique définie positive ?

La matrice est symétrique définie positive, car :

Ona A est symétrique (claire).

On calcule les valeurs propres de A :


```
A=[0.2 0.1 1 1 0;0.1 4 -1 1 -1;1 -1 600 -2;1 1 0 8 4;0 -1 -2 4 700];
```

```
>> Spectr = eig(A) %les valeurs propres de A.
```

```
Spectr =
```

```
0.0571
```

```
3.7453
```

```
8.3384
```

```
60.0284
```

```
700.0308
```

Toutes les valeurs propres sont strictement positives, alors la matrice A est définie positive.

```
* Programme : calcul le rayon spectrale de la matrice d'itération de Gauss-Seidel
```

```
A=[0.2 0.1 1 1 0;0.1 4 -1 1 -1;1 -1 600 -2;1 1 0 8 4;0 -1 -2 4 700];
```

```
P = tril(A); %
```

```
F = -triu(A,1);
```

```
Bg = inv(P) * F; % la matrice d'itération de Gauss-Seidel.
```

```
Spectr = eig(Bg); % les valeurs propres de Bg
```

```
ray - spc = max(abs(Spectr)) %le rayon spectral de Bg
```

```
>>ray - spc =
```

```
0.712
```

Programme : calcul le rayon spectrale de la matrice d'itération de Jacobi

```
A=[0.2 0.1 1 1 0;0.1 4 -1 1 -1;1 -1 600 -2;1 10 8 4;0 -1 -24 700];
```

```
P = diag(diag(A)); % matrice diagonal.
```

```
N = A - P;
```

```
Bj = inv(P) * N; % matrice d'itération de Jacobi.
```

```
Spectr = eig(Bj) % le spectr de A.
```

```
ray - spc = max(abs(Spectr)) % le rayon spectral de A.
```

```
>>
```

```
ray - spc =
```

```
0.8805
```

```
>>
```

On a : $\rho(B_J) < 1$ et $\rho(B_{G-S}) < 1$, alors les deux méthodes de Jacobi et Gauss-Seidel sont convergent.

La matrice A est DP et pour $\omega = 1.25$ alors, la méthode de relaxation est converge.

Programme : méthode de Jacobi

```
function[xvect,err,nit]=jacobi(x0,n_max,tol,A,b)
% [xvect,err,nit]=jacobi (x0,n-max,tol,A,b)
% Méthode de Jacobi pour la résolution du système linéaire Ax=b
% critère d'arrêt : ||b-A*x_n|| < tol ou it=n_max
% INPUT :
% x0      vecteur initial (vecteur colonne)
% n_max   Nombre maximal d'itérations
% tol     Tolérance sur le critère d'arrêt
% A, b    Matrice et terme de droite du système linéaire
% OUTPUT :
% xvect   Matrice contenant tous les vecteurs (colonne) x_i calculé ;
%         xvect( :,i)=i-eme vecteur calculé
% err     Vecteur contenant la norme du résidu
%         ||b-A*x_n|| à chaque itération
% nit     Nombre d'itérations effectuées
% NOTE : La solution calculée est la dernière colonne de la matrice xvect
A=[0.2 0.1 1 1 0;0.1 4 -1 1 -1;1 -1 60 0 -2;1 1 0 8 4;0 -1 -2 4 700];
b=[ 1;2;3;4;5];
x0=zeros(size(b));
n_max=50;
tol=10^-3;
P=diag(diag(A));% la matrice diagonal
N=P-A;
B=eye(size(A))-inv(P)*A;% La matrice d'itération de Jacobi
bb=P\b;
r=b-A*x0;
erreur=norm(r);
```

```
nit=1;
xvect(:,1)=x0;
err(1)=erreur;
while (nit <= n_max) & (erreur > tol)
xvect(:,nit+1)=B*xvect(:,nit)+bb;
r=b-A*xvect(:,nit+1);
erreur=norm(r)
err(nit+1)=erreur;
nit=nit+1
end
return
```

Programme 2 : méthode de Gauss-Seidel

```
function[xvect,err,nit]=Gauss-Seidel (x0,n_max,tol,A,b)
% [xvect,err,nit]=Gauss-Seidel (x0,n_max,tol,A,b)
% Méthode de Gauss-Seidel pour la résolution du système linéaire Ax=b
% critère d'arrêt : ||b-A*x_n|| < tol ou it=n_max
% INPUT :
% x0      Vecteur initial (vecteur colon)
% n_max   Nombre maximal d'itérations
% tol     Tolérance sur le critère d'arrêt
% A, b    Matrice et terme de droite du système linéaire
% OUTPUT :
% xvect   Matrice contenant tous les vecteurs (colonne) x_i calculé ;
%         xvect( :,i)=i-eme vecteur calculé
% err     Vecteur contenant la norme du résidu
%         ||b-A*x_n|| à chaque itération
% nit     Nombre d'itérations effectuées
% NOTE : La solution calculée est la dernière colonne de la matrice xvect
A=[0.2 0.1 1 1 0;0.1 4 -1 1 -1;1 -1 60 0 -2;1 1 0 8 4;0 -1 -2 4 700];
b=[ 1;2;3;4;5];
x0=zeros(size(b));
n_max=50;
tol=10^-3;
P=tril(A)
F=-triu(A,1)
B=inv(P)*F
bb=P\b;
r=b-A*x0;
erreur=norm(r);
```

```
nit=1;
xvect(:,1)=x0;
err(1)=erreur;
while (nit <= n_max) & (erreur > tol)
xvect(:,nit+1)=B*xvect(:,nit)+bb;
r=b-A*xvect(:,nit+1);
erreur=norm(r)
err(nit+1)=erreur;
nit=nit+1
end
return
```

Programme 2 : méthode de relaxation

```
function[xvect,err,nit]=relaxation (x0,n_max,tol,A,b)
% [xvect,err,nit]=relaxation(x0,n_max,tol,A,b)
% Méthode de jacobi pour la résolution du système linéaire Ax=b
% critère d'arrêt : ||b-A*x_n|| < tol ou it=n_max
% INPUT :
% x0      Vecleur initial (vecteur colonne)
% n_max   Nombre maximal d'itérations
% tol     Tolirance sur le critère d'arrêt
% A, b    Matrice et terme de droite du ssystème linéaire
% OUTPUT :
% xvect   Matrice contenant tous les vecteurs (colonne) x_i calculé ;
%         xvect( :,i)=i-eme vecteur calculé
% err     Vecteur contenant la norme du residu
%         ||b-A*x_n|| à chaque itération
% nit     Nombre d'itérations effectuées
% NOTE : La solution calculée est la dernière colonne de la matrice xvect
A=[0.2 0.1 1 1 0;0.1 4 -1 1 -1;1 -1 60 0 -2;1 1 0 8 4;0 -1 -2 4 700];
b=[ 1;2;3;4;5];
x0=zeros(size(b));
n_max=50;
tol=10^-3;
D=diag(diag(A));
P=tril(A)- D +(1/w)*D;
F=-triu(A,1);
M=((1-w)/w)*diag(diag(A))+F;
```

```
B=inv(P)*M;
bb=P\b;
r=b-A*x0;
erreur=norm(r);
nit=1;
xvect(:,1)=x0;
err(1)=erreur;
while (nit <= n_max) & (erreur > tol)
xvect(:,nit+1)=B*xvect(:,nit)+bb;
r=b-A*xvect(:,nit+1);
erreur=norm(r)
err(nit+1)=erreur;
nit=nit+1
end
return
```


Programme 3 : méthode de gradient conjugué

```
function[xvect,err,nit]= gradientconjugué (x0,n_max,tol,A,b)
% [x,err,nit]= gradientconjugué (x0,n_max,tol,A,b)
% Méthode de gradient conjugué pour la résolution du système linéaire Ax=b
% critère d'arrêt : ||b-A*x_n|| < tol ou it=n_max
% INPUT :
% x0      Vecleur initial (vecteur colonne)
% n_max   Nombre maximal d'itérations
% tol     Tolirence sur le critère d'arrêt
% A, b    Matrice et terme de droite du ssystème linéaire
% OUTPUT :
% err     Vecteur contenant la norme du residu
%         ||b-A*x_n|| à chaque itération
% nit     Nombre d'itérations effectuées
A=[0.2 0.1 1 1 0;0.1 4 -1 1 -1;1 -1 60 0 -2;1 1 0 8 4;0 -1 -2 4 700];
b=[ 1;2;3;4;5];
x0=zeros(size(b));
n_max=50;
tol=10^-3;
i=0;
r=b-A*x;
d=r
delta=r'*r
delta_0=delta
erreur=norm(r)
while (nit<n_max)& (erreur > tol)
q=A*d
alpha=delta/ (d'*q)
```

```
xold=x
x=x + alpha*d
r=r-alpha*q
%end
delta_old=delta
delta=r'*r
beta=delta/delta_old
erreur=norm(r)
d=r+beta*d
nit=nit+1
end
end
```

Méthode du gradient conjugué préconditionné :

```
>> A=[0.2 0.1 1 1 0;0.1 4 -1 1 -1;1 -1 600 -2;1 1 0 8 4;0 -1 -2 4 700];  
>> b = [1;2;3;4;5];  
>> tol = 10^-3;  
>> n_max = 50;  
>>pcg(A,b,tol,n_max) %C'est une fonction prédéfinie en matlab.  
pcg converged at iteration 5 to a solution with relative residual 3.5e - 008
```

On obtient :

méthode	Nombre d'itération	$\ Ax^{(k)} - b\ _2$
Jacobi	50	0.0126
Gauss-Seidel	17	9.0817×10^{-4}
SOR	9	5.4902×10^{-4}
Gradient conjugué	5	2.5069×10^{-6}
Gradient conjugué preconditionnée	5	3.5×10^{-8}

Le nombre d'itération et l'erreur décroît selon l'ordre des méthodes suivant : Jacobi, Gauss-Seidel, Relaxation puis Gradient conjugué.

La méthode de gradient conjugué et gradient conjugué préconditionnée ont le même nombre d'itération mais, l'erreur dans gradient conjugué est plus grand que celle de gradient conjugué préconditionnée.

3.2 Exemple 2

Exemple 3.2.1

Soit la matrice A suivante :

$$A = \begin{pmatrix} 1 + 2r & -r & 0 & \dots & \dots & 0 \\ -r & 1 + 2r & -r & 0 & \dots & 0 \\ 0 & -r & 1 + 2r & -r & 0 & 0 \\ 0 & 0 & \cdot & \dots & \cdot & 0 \\ 0 & 0 & 0 & \cdot & \dots & -r \\ 0 & 0 & 0 & 0 & -r & 1 + 2r \end{pmatrix}$$

Cette matrice est diagonale dominante stricte, alors les deux méthodes de Jacobi et Gauss-Seidel sont convergent.

Pour $n = 10$: $\rho(B_{G-S}) = 0.1023$ et pour $n = 100$ $\rho(B_{G-S}) = 0.1110$.

On prend $r = 0.25$.

toutes les termes diagonaux sont strictement positive.

Alors, A est une matrice symétrique à diagonale strictement dominant et strictement positive, alors elle est symétrique définie positive.

A est définie positive et avec $\omega = 1.25$ alors la méthode de relaxation est converge.

On applique les trois méthodes précédent dans le système $Ax = b$, avec matlab pour $n = 10$ et $n = 100$, avec $b = [1, 1, \dots, 1, 1]$, $tol = 10^{-14}$, $n_max = 100$, et $x_0 = 0$.

Prenons $n = 10$:

Ce programme génère une matrice tridiagonal :

```
n = 10
r = 0.25;
A = zeros(n,n);
for i = 1 : n - 1
    A(i,i) = 1 + 2 * r;
    A(i,i + 1) = -r;
    A(i + 1,i) = -r;
end
A(n,n) = 1 + 2 * r;
A = full(A)
```

```
A =
    1.5000   -0.2500         0         0         0         0         0         0         0         0
   -0.2500    1.5000   -0.2500         0         0         0         0         0         0         0
         0   -0.2500    1.5000   -0.2500         0         0         0         0         0         0
         0         0   -0.2500    1.5000   -0.2500         0         0         0         0         0
         0         0         0   -0.2500    1.5000   -0.2500         0         0         0         0
         0         0         0         0   -0.2500    1.5000   -0.2500         0         0         0
         0         0         0         0         0   -0.2500    1.5000   -0.2500         0         0
         0         0         0         0         0         0   -0.2500    1.5000   -0.2500         0
         0         0         0         0         0         0         0   -0.2500    1.5000   -0.2500
         0         0         0         0         0         0         0         0   -0.2500    1.5000
```

Et

$$b = \begin{pmatrix} 1 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{pmatrix}$$

Méthode	Nombre d'itération	$\ Ax - b\ _2$
Jacobi	30	2.6709×10^{-15}
Gauss-Seidel	18	2.91×10^{-15}
SOR	19	7.7119×10^{-15}
Gradient conjugué	5	8.0905×10^{-19}
Gradient conjugué preconditionnée	5	7×10^{-17}

Le nombre d'itération dans Jacobi plus grand que celle de Gauss-Seidel, contrairement pour l'erreur.

Le nombre d'itération et l'erreur dans relaxation sont plus grand que celle de Gauss-Seidel.

Le nombre d'itération et l'erreur dans gradient conjugué sont plus petit que celle de les trois méthodes précédents..

La méthode de gradient conjugué et gradient conjugué preconditionnée ont le même nombre d'itération mais, l'erreur dans gradient conjugué est plus petit que celle de gradient conjugué preconditionnée.

Prenon $n = 100$:

Méthode	Nombre d'itération	$\ Ax - b\ _2$
Jacobi	32	4.5627×10^{-15}
Gauss-Seidel	22	3.3748×10^{-15}
SOR	22	7.733×10^{-15}
Gradient conjugué	19	$7.7334/5.57 \times 10^{-15}$
Gradient conjugué preconditionnée	18	1.7×10^{-17}

on remarque que :

La méthode de Gauss-Seidel et relaxation ont même nombre d'itération mais, l'erreur dans Gauss-Seidel est plus petit que celle de relaxation.

Le nombre d'itération et l'erreur dans Jacobi sont plus grand que celle de Gauss-Seidel.

nombre d'itération dans gradient conjugué est plus petit que celle de Gauss-Seidel ,
contrairement pour l'erreur.

nombre d'itération et l'erreur dans la méthode gradient conjugué est plus grand que
celle de gradient conjugué préconditionnée

3.3 Exemple 3

```
%ce programme génère une marice tridiagonal
%SDP de taille n
n = 10
D = sparse(1 : n, 1 : n, 1/2 * ones(1, n), n, n);
E = sparse(2 : n, 1 : n - 1, 1/8 * ones(1, n - 1), n, n);
F = sparse(1 : n - 1, 2 : n, 1/8 * ones(n - 1, 1), n, n);
A = E + D + F; %A est symétrique
eig(A) % les valeurs propres doivent être toutes positives
A = full(A)
```

```
A =
    0.5000    0.1250         0         0         0         0         0         0         0         0
    0.1250    0.5000    0.1250         0         0         0         0         0         0         0
         0    0.1250    0.5000    0.1250         0         0         0         0         0         0
         0         0    0.1250    0.5000    0.1250         0         0         0         0         0
         0         0         0    0.1250    0.5000    0.1250         0         0         0         0
         0         0         0         0    0.1250    0.5000    0.1250         0         0         0
         0         0         0         0         0    0.1250    0.5000    0.1250         0         0
         0         0         0         0         0         0    0.1250    0.5000    0.1250         0
         0         0         0         0         0         0         0    0.1250    0.5000    0.1250
         0         0         0         0         0         0         0         0    0.1250    0.5000
```



```
>> b=A*ones(n,1)
```

```
b =
```

```
0.6250  
0.7500  
0.7500  
0.7500  
0.7500  
0.7500  
0.7500  
0.7500  
0.7500  
0.7500  
0.6250
```

Cette matrice est diagonale dominante stricte, alors les deux méthodes de Jacobi et Gauss-Seidel sont convergent.

Pour $n = 10$: $\rho(B_{G-S}) = 0.2302$ et pour $n = 100$ $\rho(B_{G-S}) = 0.2498$.

A est définie positive et avec $\omega = 1.25$ alors la méthode de relaxation est converge.

On applique les trois méthodes précédent dans le système $Ax = b$, avec matlab pour $n = 10$ et $n = 100$, avec $tol = 10^{-2}$, $n_max = 20$, $x_0 = 0$ et une solution exact $x = [1, 1 \quad 1, 1]$.

Prenons $n = 10$:

Méthode	Nombre d'itération	$\ Ax - b\ _2$
Jacobi	8	0.0083
Gauss-Seidel	4	0.0042
SOR	7	0.0056
Gradient conjugué	4	0.0033
Gradient conjugué preconditionnée	3	0.0044

Prenons $n = 100$:

Méthode	Nombre d'itération	$\ Ax - b\ _2$
Jacobi	10	0.0095
Gauss-Seidel	5	0.0032
SOR	9	0.0047
Gradient conjugué	4	0.0042
Gradient conjugué preconditionnée	2	0.0059

On remarque que (pour $n = 10$ et $n = 100$) :

Le nombre d'itération et l'erreur sont décroissent dans les méthodes selon l'arrangement suivant : Jacobi, relaxation, Gauss-Seidel puis gradient conjugué.

nombre d'itération dans la méthode gradient conjugué est plus grand que gradient conjugué preconditionnée , contrairement pour l'erreur.

Conclusion

Dans ce mémoire, on essaye d'étudier une classe importante de méthodes de résolution des systèmes linéaires est celle des méthodes itératives, et on expose ces deux classes : stationnaires et non stationnaires et on les compare les uns aux autres au sens de rapidité.

D'après les exemples précédents :

on remarque que la méthode de Jacobi est très lente, c'est pour cette raison on a amélioré par la méthode de Gauss-Seidel.

Dans la méthode de Gauss-Seidel, si le rayon spectral est très proche de 1, alors la convergence devient lente; On a donc introduit le paramètre ω qui accélère la convergence. (i.e le nombre d'itération diminue)/

En revanche dans les méthodes projectives : la méthode de gradient conjugué possède un taux de convergence valable. Le seul inconvénient dans cette méthode est quand la taille de la matrice est très grande. Ce qui nous conduit à un mauvais conditionnement. Si la matrice A est mal conditionnée, la convergence de l'algorithme du gradient conjugué est lente. Dans ce cas, on recourt à améliorer la vitesse de convergence; c'est la méthode du gradient conjugué préconditionnée. Le seul problème est dans le choix de la matrice de préconditionnement. Comment choisir un préconditionneur est une question très délicate qui à ce jour n'a pas de solution universelle.

En analyse numérique, il n'existe pas une meilleure stratégie de préconditionnement valable pour tout type de matrices, et leur efficacité varie fortement d'un système à l'autre.

Enfin, On ne peut pas conclure sur le meilleur choix des méthodes, car chaque méthode a des inconvénients et des avantages et il reste à ce jour un problème de recherche.

Bibliographie

- [1] Åke Björck "Numerical Methods in Matrix Computations" Springer International Publishing Switzerland 2015
- [2] R. L. Burden et J.D. Faires "Numerical analysis".unge-Kutta, Université de Grenoble ,Master
- [3] W. Ford "Numerical lineare algebra with applications" Ac. Press 2010.
- [4] F. Jedrzejewski "Introduction aux méthodes numériques".Deuxième édition. Springer-verlage france, paris 2005.
- [5] J. Rappaz et M. Picasso "Introduction à l'analyse numérique" Presses polytechnique et univeritaires romandes, 2004.
- [6] A. Quarteroni, R. Sacco et Fausto Saleri "méthodes numériques" Springer- Verlag Italia,Milano 2007.
- [7] A. Quarteroni, Riccardo Sacco et Paola Gervasio "Calcul Scientifique - Deuxième édition" Springer-Verlag Italia 2010.

ملخص :

في هذا العمل حاولنا أن نعطي لمحة عامة عن نوع مهم من أنواع طرق حلول جمل معادلات خطية والذي تناولنا من خلاله الطرق التكرارية (المستقرة و غير المستقرة و حاولنا المقارنة بينهم). في البداية قدمنا تذكير و مفاهيم أساسية حول المصفوفات، جمل المعادلات الخطية، الطويلة و غيرها ثم قدمنا لمحة عن الطرق التكرارية و شروط تقاربها، و من ثم تطرقنا للطرق التكرارية المستقرة و غير المستقرة. و أخيرا حاولنا إجراء مقارنة لحلول بعض جمل المعادلات الخطية باستعمال بعض الطرق التكرارية وذلك من خلال أمثلة.

كما هو معروف طرق حل جمل المعادلات الخطية تلعب جورا هاما في شتى المجالات، ولا تزال حتى يومنا هذا موضوع بحث.

الكلمات المفتاحية :

جملة معادلات خطية، مصفوفة، طويلة، طريقة تكرارية، مستقرة، غير مستقرة، تقارب، سرعة التقارب

Résumé :

Dans ce mémoire, on essayer d'étudier une classe important de méthode de résolution des systèmes linéaires est celle des méthodes itératives stationnaires et non stationnaires et on les comparant les uns aux autres.

Au début, nous allons donner des rappelles et des concepts de base : les matrices, systèmes linéaires, normes, conditionnement. Ensuite, nous allons donner un aperçu sur les méthodes itératives avec des conditions de convergences. De plus, on exposée ces deux classes : stationnaires et non stationnaires.

Enfin, on donne une comparaison, entre quelques méthodes itératives pour la résolution des systèmes linéaires.

Mots clés :

Système linéaire, matrice, norme, définie positive, Méthode itérative, stationnaire, non-stationnaire, convergence, vitesse de convergence

Abstract:

In this memory, we try to study an important class of method of solving linear systems is that of the iterative stationary and non-stationary methods and comparing them with each other.

At the beginning, we will give reminders and basic concepts: the matrices, linear systems, standards, conditioning. Next, we will give an overview on the iterative methods with convergence conditions. In addition, these two classes are exposed: stationary and non-stationary.

As is known, the linears systems play a very important role in the scientific, they rest until our present a research area.

Key words :

Linear systems, matric, norm, definit positiv, conditioning, Iterativs methods,, stationary non-stationary, ,convergence conditioning

