Analyse Numérique 2

Abstract

Chapter 1

Introduction

L'analyse numérique a pour objectif de résoudre des problèmes issus de la physique, de l'industrie, de la finance, de la biologie... à utilisant la capacité qu'ont les ordinateurs d'effectuer des milliards d'opérations en une seconde. Pour cela, il faut convevoir des algorithmes permettant de transformer un problème mathématique en un programme capable de le résoudre (ou du moins d'en trouver une solution approchée).
Même si les ordinateurs sont de plus en plus puissants, il est toujours nécessaire de développer des algorithmes performant permettant de limiter les temps de calculs, suffisament précis et robustes (sensibilité aux erreurs sur les données, aux erreurs d'arrondis...), et utilisant le moins de mémoire possible.

Si on résoud le système linéaire $Ax = b$ par une méthode de Cramer qui consite à dire que $x_i = det\ A_i / det\ A$ où x_i est la ième composante du vecteur x et A_i est la matrice A dans laquelle on a remplacé la ième colonne par le vecteur b. Si le système est de taille n, pour une résolution on a

- n divisions
- $(n+1)$ déterminants à calculer soit $(n+1) * n * n!$

soit au total $(n+1)n! + n$ opérations. Si le système est de taille $n = 25$, le nombre d'opérations nécessaire sera de $4x10^{26}$. Sur une machine effectuant 10^9 opérations par seconde, soit $3x10^25$ opérations par milliard d'année, il faudrait plus de 10 milliards d'années pour résoudre ce problème.

Quelques domaines d'application:

- Energie: nucléaire (fusion et fission), hydrocarbure, énergies renouvelables
- Transport: Aéronautique, automobile, spatial
- Télécommunication: sattelites
- Environement: météorologie, géophysique, climatologie...
- Finance et assurance

Chapter 2

Rappels d'algèbre linéaire

2.1 Premiers rappels

Considérons u un vecteur colonne à n composantes à valeurs dans un corps $\mathbb{K}$ ($\mathbb{R}$ ou $\mathbb{C}$), On note A la matrice à m lignes et n colonnes et B la matrice à n lignes et p colonnes. On note $A \in \mathcal{M}_{m,n}(\mathbb{K})$ et $B \in \mathcal{M}_{n,p}(\mathbb{K})$.

On rappelle que le produit de la matrice A et du vecteur u est le vecteur v de $\mathbb{K}^m$ défini par

$$v_i = \sum_{j=1}^{n} a_{i,j} u_{j,i} = 1, ..., m$$

où les $a_{i,j}$, $1 \leq i \leq m$, $1 \leq j \leq n$ sont les composantes de la matrice A et les u_i, $1 \leq i \leq n$ sont les composantes du vecteur u.

De la même façon, le produit de la matrice A et B est la matrice $C \in \mathcal{M}_{m,p}$ dont les composantes $c_{i,j}$ sont définies par

$$c_{i,j} = \sum_{k=1}^{n} a_{i,k} b_{k,j},\ 1 \leq i \leq m,\ 1 \leq j \leq p$$

Définition 1.

- *On appelle matrice transposée et on note $A^T \in \mathcal{M}_{n,m}$ la matrice définie par*
$$a^T_{i,j} = a_{j,i}, 1 \leq i \leq m,\ 1 \leq j \leq n$$
- *On appelle matrice adjointe et on note $A^* \in \mathcal{M}_{n,m}$ la matrice définie par*
$$a^*_{i,j} = \overline{a}_{j,i}, 1 \leq i \leq m,\ 1 \leq j \leq n$$

Soient $u, v \in \mathbb{K}^n$, on note $(\,.\,,\,.\,)$ le produit scalaire usuel défini par

$$(u, v) = \sum_{i=1}^{n} u_i v_i^*$$

2.2 Matrices carrées

Dans ce paragraphe, on suppose que $A \in \mathcal{M}_{n,n}(\mathbb{K})$ ou $\mathcal{M}_n(\mathbb{K})$ est une matrice carrée.

2.2.1 Matrices particulières

Définition 2.

- *A est dite inversible s'il existe une matrice $B \in \mathcal{M}_n(\mathbb{K})$ telle que $AB = BA = I_n$ où I_n est la matrice identité de taille n. La matrice B est appelée matrice inverse de A et est notée A^{-1}.*

Proposition 1.

- $(A^{-1})^{-1} = A$
- *Soit $k \in \mathbb{K}$,* $(kA)^{-1} = k^{-1}A^{-1}$
- $(AB)^{-1} = B^{-1}A^{-1}$

Définition 3. *Soit $A \in \mathcal{M}_n(\mathbb{K})$,*

- *A est symétrique si $A^T = A$*
- *A est hermitienne si $A^* = A$*
- *A est orthogonale si $A^{-1} = A^T$*

La résolution de problèmes par des méthodes numériques se ramène généralement à la résolution d'un (ou plusieurs) systèmes linéaires creux, c'est à dire avec une matrice comportant beaucoup de coefficients nuls. Présentons quelques cas particuliers de matrices creuses

Définition 4. *Soit $A \in \mathcal{M}_n(\mathbb{K})$ de coefficients $a_{i,j}$, $1 \leq i, j \leq n$.*

- *A est dite triangulaire inférieure ssi*

$$a_{i,j} = 0,\ 1 \leq i < j \leq n.$$

- *A est dite triangulaire supérieure ssi*

$$a_{i,j} = 0,\ 1 \leq j < i \leq n.$$

- *A est dite diagonale si*

$$a_{i,j} = 0,\ i \neq j$$

Les systèmes linéaires triangulaires ou diagonaux présentent l'avantage d'être très simple à résoudre. Elles sont par conséquent très utiles en analyse numérique (cf chap suivant).

Théorème 1. *L'inverse d'une matrice triangulaire inférieure (resp. supérieure) est triangulaire aussi une matrice inférieure (resp. supérieure).*

Proof: Soit L une matrice triangulaire inférieure de taille n inversible. Notons $A = (a_{i,j})_{1 \leq i,j \leq n}$ son inverse. On a

$$\delta_{i,j} = \sum_{k=1}^{n} l_{i,k} a_{k,j} = \sum_{k=1}^{i} l_{i,k} a_{k,j}$$

Pour $i = 1$, et $j > 1$ on a

$$l_{1,1} a_{1,j} = 0$$

d'où $a_{1,j} = 0$.
Pour $i = 2$ et $j > i$,

$$l_{2,1} a_{1,j} + l_{2,2} a_{2,j} = 0$$

d'où $a_{2,j} = 0$. Par récurrence, on montre que pour tout $i > j$, $a_{i,j} = 0$.
La matrice A est donc bien triangulaire inférieure.

2.2.2 Valeurs propres et vecteurs propres

Définition 5. *Soit A une matrice carrée*

- $\lambda \in \mathbb{K}$ *est une valeur propre de A si $det(A - \lambda I_n) = 0$.*
- *L'ensemble des valeurs propres de A est appelé spectre de A et noté $Sp(A)$*
- *Enfin, on appelle rayon spectral de A le nombre $\rho(A)$ defini par*

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

Proposition 2. *Soit A une matrice carrée et $\lambda \in \mathbb{K}$ est une valeur propre de A. Il existe au moins un vecteur $\boldsymbol{u}$ non nul vérifiant*

$$A\boldsymbol{u} = \lambda \boldsymbol{u}.$$

On appelle $\boldsymbol{u}$ vecteur propre.

Définition 6. *Une matrice $A \in \mathcal{M}_n(\mathbb{R})$ est diagonalisable s'il existe une matrice P inversible et une matrice D diagonale telle que*

$$A = PDP^{-1}$$

Proposition 3. *La diagonale de la matrice D est constituée des n valeurs propres de A et les colones de matrice inversible P sont constitué des vecteur propres associés qui forment une base.*

Proposition 4. *Une matrice carrée A d'ordre n qui a exactement n valeurs propres distinctes est diagonalisable.*

Proposition 5. *Si A est réelle symetrique, alors A est diagonalisable dans $\mathbb{R}$ dans une base orthonormée, c'est a dire qu'il existe une matrice orthogonale P et une matrice diagonale $D \in M_n(\mathbb{R})$ telle que*

$$A = PDP^T$$

Si la matrice A est hermitienne ($A^* = A$), alors il existe une matrice unitaire U ($U^* = U^{-1}$) telle que

$$A = UDU^*$$

2.3 Normes matricielles

On considère $\mathcal{M}_n(\mathbb{K})$ l'espace des matrices carrées d'ordre n.

Définition 7. *L'application* $\|.\| : \mathcal{M}_n(\mathbb{K}) \to \mathbb{R}^+$ *est une norme matricielle si elle satisfait les 4 propriétés suivantes:*

1. $\forall A \in \mathcal{M}_n(\mathbb{K}),\ \|A\| = 0 \Rightarrow A = 0$
2. $\forall A \in \mathcal{M}_n(\mathbb{K}),\ \forall \alpha \in \mathbb{K},\ \|\alpha A\| = |\alpha|\|A\|$
3. $\forall A, B \in \mathcal{M}_n(\mathbb{K}),\ \|A + B\| \leq \|A\| + \|B\|$
4. $\forall A, B \in \mathcal{M}_n(\mathbb{K}),\ \|AB\| \leq \|A\|.\|B\|$

Comme dans le cas des normes vectorielles, nous allons définir des normes usuelles pour le cas des normes matricielles.

Définition 8. *Soit* $\|.\|$ *une norme vectorielle sur* $\mathbb{K}^n$*, on définit la norme matricielle subordonnée à la norme vectorielle* $\|.\|$ *que l'on note également* $\|.\|$ *par*

$$\forall A \in \mathcal{M}_n(\mathbb{K}), \|A\| = \sup_{v \in \mathbb{K}^n} \frac{\|Av\|}{\|v\|}$$

Remarque 1. *Pour une norme subordonnée,* $\forall v \in \mathbb{K}^n,\ \|Av\| \leq \|A\|\|v\|$

Condidérons les normes vectorielles

$$\forall v \in \mathbb{R}^n,\ \|v\|_p = \left(\sum_{k=1}^{n} |v_i|\right)^{1/p},\ p \in \mathbb{N}^*$$

et

$$\forall v \in \mathbb{R}^n, \|v\|_\infty = \max_{1 \leq i \leq n} |v_i|,$$

on définit pour toute matrice $A \in \mathcal{M}_n(\mathbb{K})$

$$\|A\|_p = \sup_{v \in \mathbb{K}^n} \frac{\|Av\|_p}{\|v\|_p},\ 1 \leq p \leq \infty$$

Proposition 6. *Soit* $A \in M_n(\mathbb{K})$*, alors*

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

et

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

Proof: Soit $v \in \mathbb{K}^n$, on a

$$\|Av\|_1 = \sum_{i=1}^{n} |\sum_{j=1}^{n} a_{i,j} v_j| \leq \sum_{i=1}^{n} \sum_{j=1}^{n} |a_{i,j}||v_j|,$$

d'où

$$\|Av\|_1 \leq \sum_{j=1}^{n} \left(\sum_{i=1}^{n} |a_{i,j}| \right) |v_j| \leq \max_{1\leq j\leq n} \sum_{i=1}^{n} |a_{i,j}| \|v\|_1.$$

Par conséquent,

$$\|A\|_1 \leq \max_{1\leq j\leq n} \sum_{i=1}^{n} |a_{i,j}|.$$

D'autre part, on peut montrer qu'il existe un vecteur $u \in \mathbb{K}^n$ unitaire tel que

$$\|Au\|_1 = max_{1\leq j\leq n} \sum_{i=1}^{n} |a_{i,j}|.$$

En choisissant $j_0 \in \{1, ..., n\}$ tel quel

$$\sum_{i=1}^{n} |a_{i,j_0}| = \max_{1\leq j\leq n} \sum_{i=1}^{n} |a_{i,j}|$$

et en penant u tel que $u_i = \delta_{i,j_0}$, on a

$$\|Au\|_1 = \sum_{i=1}^{n} |(Au)_i| = \sum_{i=1}^{n} |a_{i,j_0}| = \max_{1\leq j\leq n} \sum_{i=1}^{n} |a_{i,j}|$$

d'où le résultat.

2.4 Conditionnement d'une matrice

Considérons la matrice

$$\begin{pmatrix} 1 & 1/2 & 1/3 & 1/4 \\ 1/2 & 1/3 & 1/4 & 1/5 \\ 1/3 & 1/4 & 1/5 & 1/6 \\ 1/4 & 1/5 & 1/6 & 1/7 \end{pmatrix}$$

et le vecteur

$$b^T = (\frac{25}{12}, \frac{77}{60}, \frac{57}{60}, \frac{319}{420}) \simeq (2.0833, 1.2833, 0.9500, 0.7599).$$

En résolvant $Ax = b$, on obtient

$$x^T = (1, 1, 1, 1).$$

Maintenant remplaçons b par

$$\tilde{b}^T = (2.1, 1.3, 1, 0.8).$$

En résolvant, $A\tilde{x} = \tilde{b}$ La solution obtenu est

$$\tilde{x}^T = (5.6, -48, 114, -70).$$

On remarque donc qu'une très légère modification du terme de droite b affecte de manière significative la solution x du système.

De manière plus générale, si on considère une matrice réelle A inversible de taille n et un vecteur b de taille n, la solution x du problème $ax = b$ est donnée par

$$x = A^{-1}b.$$

En perturbant de δb le vecteur b, on peut écrire la solution du système perturbé sous la forme $x + \delta x$:

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

Si on considère une norme vectorielle $\|.\|$ sur $\mathbb{R}^n$, nous allons controler l'erreur relative $\|\delta x\|/\|x\|$ en fonction de $\|\delta b\|/\|b\|$ et de la norme subordonnée de la matrice A.
Par linéarité, on a

$$A\delta x = \delta b \Rightarrow \|\delta x\| \leq \|A^-1\|\|\delta b\|$$

et

$$Ax = b \Rightarrow \|b\| \leq \|A\|\|x\|.$$

On obtient alors

$$\frac{\|\delta x\|}{\|x\|} \leq \|A^{-1}\|\|A\|\frac{\|\delta b\|}{\|b\|}.$$

Définition 9. *On appelle conditionnement de la matrice A relativement à la norme matricielle $\|.\|$ subordonnée à la norme vectorielle $\|.\|$ la quantité*

$$cond(A) = \|A\|.\|A^{-1}\|.$$

- *un système linéaire est bien conditionné si $cond(A)$ n'est "pas trop grand".*
- *un système linéaire est mal conditionné si $cond(A)$ est "grand".*

Proposition 7. *Soit $A \in \mathcal{M}_n(\mathbb{K})$ une matrice inversible et $\|.\|$ une norme matricielle subordonnée à la norme vectorielle $\|.\|$, on a*

- $cond(I_n) = 1$
- $cond(A) \geq 1$

Proof: Le premier point est évident. on sait que $cond(AA^{-1}) = cond(I_n) = 1$.
De plus,

$$cond(AA^{-1}) = \|AA^{-1}\|\|AA^{-1}\| \leq (\|A\|\|A^{-1}\|)^2 = cond(A)^2$$

d'où $cond(A) \geq 1$.

Remarque 2. *Le conditionnement est un moyen de mesurer la sensibilité d'un système linéaire aux petites perturbations. Il n'est cependant pas toujours facile de le calculer étant donné qu'il nécessite la connaissance de l'inverse de la matrice.*

Proposition 8. *Soit A est hermitienne (en particulier symetrique réelle) alors*

$$\|A\|_2 = \rho(A).$$

De plus, si $\mathbb{K} = \mathbb{R}$ et A est une matrice symétrique inversible, alors

$$cond_2(A) = \frac{max\{|\lambda_i|, i = 1, ..., n\}}{\min\{|\lambda_i|, i = 1, ..., n\}}$$

Proof: Comme A est une matrice hermitienne, il existe une matrice U telle que $UU^* = I_n$ et

$$U^*AU = diag(\lambda_i)$$

où les (λ_i) sont les valeurs propres de A. De plus

$$\|A\|_2^2 = \sup_{v\in\mathbb{K}^n} \frac{\|Av\|_2^2}{\|v\|_2^2} = \sup_{v\in\mathbb{K}^n} \frac{(Av)^*(Av)}{v^*v}$$

Or,

$$\begin{aligned} v^*AAv &= v^*UU^*A^*UU^*AUU^*v \\ &= (U^*v)^*(diag(\lambda_i))^*(diag(\lambda_i))U^*v. \end{aligned}$$

D'autre part, $U^*AU = (U^*AU)^*$, et en posant $w = U^*v$ on a

$$v^*A^*Av = w^*diag(\overline{\lambda}_i)diag(\lambda_i)w.$$

Ainsi,

$$\|A\|_2^2 = sup_{w\in K^n} \frac{w^*diag(|\lambda_i|^2)w}{w^*w} = \rho(A)^2.$$

Si on se place dans le cas réel, A est une matrice inversible. Comme A est inversible, les valeurs propres de A sont non nulles, et $1/\lambda_i$ est valeur propre de A^-1. On peut aplliquer le résultat précédent à A^-1 et on a

$$\|A^{-1}\|_2 = \rho(A^{-1}) = \frac{1}{\min\{\lambda_i, 1 \le i \le n\}}$$

Comme $cond_2(A) = \|A\|_2\|A^{-1}\|_2$, on en déduit le résultat.

2.5 Préconditionnement d'un système linéaire

Afin de résoudre $Ax = b$, on multiplie à gauche par une matrice inversible P

$$PAx = Pb$$

avec P choisie de manière que PA soit bien (ou mieux) conditionnée (la meilleure solution serait $P = A^{-1}$.
Dans ce cas, on parle de préconditionneur à gauche. De la mêm façon, on peut définir un préconditionneur à droite:

$$APz = b,\ x = Pz$$

Il existe d'autre moyen de préconditionner un système. Nous en verrons quelques exemple en TD.

Chapter 3

Methode de Gauss - Decomposition LU

3.1 Resolution systeme triangulaire

Les systemes triangulaires sont simples a resoudre.

$$x - y = 2 \tag{3.1}$$
$$3y + 2z = -1 \tag{3.2}$$
$$z = 2 \tag{3.3}$$

Eq(3.3) donne $z = 2$.
En injectant $z = 2$ dans (3.2), on trouve $y = 1$.
En injectant $y = 1$ dans (3.1), on trouve $x = 3$

Les systemes des taille n se resolvent de maniere similaire (par recurrence).

3.2 Methode de Gauss sans Pivot

Supposons maintenant qu'on ai un systeme d'equation non triangulaire. La methode du Pivot de Gauss permet de ramener la resolution de ce systeme generale a la resolution d'un systeme triangulaire.

Proposition 9. *operations autorisees:*

- $l_i \leftrightarrow l_j$ *: echanger deux lignes*
- $l_i \leftarrow \alpha l_i$ *: multiplier une ligne par* α *non nul*
- $l_i \leftarrow l_i + \alpha l_j$ *ajouter une ligne a une autre pour* $i \neq j$

3.2.1 Algorithme

But: rendre le systeme triangulaire.
Etape k:

- pour $k < i \leq n$, $l_i \leftarrow l_i - \frac{a_{k,k}}{a_{i,k}} l_k$

3.2.2 Exemple

$$x - y = 2 \tag{3.4}$$
$$2x + y + 2z = 1 \tag{3.5}$$
$$x + 2y + 3z = 1 \tag{3.6}$$

Operations: $l_2 \leftarrow l_2 - 2l_1$ puis $l_3 \leftarrow l_3 - l_1$

$$x - y = 2 \tag{3.7}$$
$$3y + 2z = -3 \tag{3.8}$$
$$3y + 3z = -1 \tag{3.9}$$

Operation: $l_3 \leftarrow l_3 - l_2$

$$x - y = 2 \tag{3.10}$$
$$3y + 2z = 1 \tag{3.11}$$
$$z = 2 \tag{3.12}$$

3.3 Decomposition LU

La methode de gauss permet de resoudre les systemes de type $Ax = b$. Comment resoudre le systeme $Ax = b'$? Il faut a priori recommencer depuis le debut. Cependant, on remarque que les operations de la methode de Gauss ne dependent pas du membre de droite b.

Pour pouvoir resoudre tout systeme $Ax = b$ (avec A fixe), on introduit la decomposition LU de A.

Définition 10. *Soit $A \in M_n(\mathbb{K})$.*
S'il existe une matrice L triangulaire inferieure et une matrice U triangulaire superieur tel que

$$A = LU$$

alors on dit que A admet une decomposition LU.

Si L est inversible, resoudre $Ax = b$ revient a resoudre $Ux = L^{-1}b$ ou U est triangulaire supperieur, ce qui est facile.

L'algorithme du pivot de Gauss permet de calculer U et L^{-1}. L'example de la section precedante s'ecrit matricielement

$$Ax = b$$

avec $A = \begin{pmatrix} 1 & -1 & 0 \\ 2 & 1 & 2 \\ 1 & 2 & 3 \end{pmatrix}$ et $b = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}$

Définition 11. *On note*

$$E_{i,j} = (e_{i,j})_{i,j}$$

avec $e_{i,j} = \delta_{i,j}$

$E_{i,j}$ a un seul coefficient non nul, celui en position (i, j).

Proposition 10. *$B = E_{i,j}A$ est une matrice ou la i-eme ligne contient la j-eme ligne de A, les autres lignes etant vides.*

Proposition 11. *Soient i, j avec $i \neq j$*

- *$l_i \leftarrow l_i + \alpha l_j$: on multiplie a gauche A et b par $(I_n + \alpha E_{i,j})$ ie*

$$(I_n + \alpha E_{i,j})A = (I_n + \alpha E_{i,j})b$$

- *$l_i \leftrightarrow l_j$: on multiplie a gauche A et b par une matrice de permutation P.*

Proposition 12. *Pour tout $j < n$:*

$$\left(I_n + \alpha_{j+1}E_{i_{j+1},j}\right) \dots (I_n + \alpha_n E_{i_n,j}) = (I_n + \alpha_{j+1}E_{i_{j+1},j} + \dots + \alpha_n E_{i_n,j})$$

On note

$$B^{(j)}(\alpha) = \alpha_{j+1}E_{i_{j+1},j} + \dots + \alpha_n E_{i_n,j}$$
$$L^{(j)}(\alpha) = (I_n + B^{(j)}(\alpha))$$

Proposition 13. *Pour tout $\alpha^{(1)}, \dots, \alpha^{(n-1)}$:*

$$L^1(\alpha^{(1)}) \dots L^{n-1}(\alpha^{(n-1)}) = (I_n + B^{(1)}(\alpha^{(1)}) + \dots B^{(n-1)}(\alpha^{(n-1)})$$

Autrement dit, on peut se passer de multiplier les matrices et faire une simple somme.

Théorème 2. *Soit A une matrice d'ordre n tel que toute ses sous matrices soient inversibles. Alors il existe L triangulaire inferieur avec $l_{i,i} = 1$ et U triangulaire superieur tel que*

$$A = LU$$

De plus, cette decomposition est unique.

Théorème 3. *Si A est symetrique (ou hermitienne) definie positive, alors A admet une decomposition LU.*

3.3.1 Exemple

Utilisons maintenant la methode de Gauss pour calculer la decomposition LU associe au systeme $Ax = b$ avec

$$A = \begin{pmatrix} 1 & -1 & 0 \\ 2 & 1 & 2 \\ 1 & 2 & 3 \end{pmatrix} \text{ et } b = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}$$

- $l_2 \leftarrow l_2 - 2l_1$ puis $l_3 \leftarrow l_3 - l_1$ correspond donc a multiplier le systeme par $I_3 - 2E_{2,1}$ puis par $I_n - E_{3,1}$.

$$\begin{aligned} Ax = b &\iff (I_n - E_{3,1})(I_3 - 2E_{2,1})A = (I_n - E_{3,1})(I_3 - 2E_{2,1})b \\ &\iff (I_n - 2E_{2,1} - E_{3,1})A = (I_n - E_{3,1})(I_3 - 2E_{2,1})b \end{aligned}$$

- $l_3 \leftarrow l_3 - l_2$: multiplication par $I_n - E_{3,2}$ donc

$$\begin{aligned} Ax = b &\iff (I_n - E_{3,2})(I_n - 2E_{2,1} - E_{3,1})A = (I_n - E_{3,2})(I_n - 2E_{2,1} - E_{3,1})b \\ &\iff (I_n - 2E_{2,1} - E_{3,1} - E_{3,2})A = (I_n - E_{3,2})(I_n - E_{3,1})(I_3 - 2E_{2,1})b \end{aligned}$$

Il est maintenant simple d'obtenir la decomposition LU de A. On pose

$$\begin{aligned} U &= (I_n - 2E_{2,1} - E_{3,1} - E_{3,2})A \\ L &= ((I_n - E_{3,2})(I_n - E_{3,1})(I_3 - 2E_{2,1}))^{-1} \\ &= (I_3 - 2E_{2,1})^{-1}(I_n - E_{3,1})^{-1}(I_n - E_{3,2})^{-1} \\ &= (I_3 + 2E_{2,1})(I_n + E_{3,1})(I_n + E_{3,2}) \\ &= (I_n + 2E_{2,1} + E_{3,1} + E_{3,2}) \end{aligned}$$

U est bien triangulaire superieur et L triangulaire inferieur.

3.3.2 Algorithme

- Initialisation: $U = A$, $L = I_n$
- Pour $1 \leq k < n$:
  - Sur la matrice U, on effectue les operations

$$l_i \leftarrow l_i - u_{i,k}/u_{k,k} l_k$$

    pour $k < i \leq n$ ou l_i designe la i-eme ligne de la matice U
  - Sur la matrice L,

$$l_{i,k} = u_{i,k}/u_{k,k}$$

    pour $k < i \leq n$
    (attention, ici $l_{i,j}$ designe le coef en position (i, j) de la matrice L, alors que precedement, l_i designait la i-eme ligne de U)

3.4 Methode de Gauss avec pivot

Il n'est pas toujours possible de faire la methode de Gauss sans pivot, en effet, si a l'etape i, $a_{i,i} = 0$, la methode de Gauss sans pivot ne fonctionne pas. Pour pallier a ce probleme, on utilise la methode de Gauss avec pivot.

Dans cette methode, on ne cherche pas a construire un systeme triangulaire, mais un systeme triangulaire a permutation pres.

3.4.1 Algorithme

a l'etape k:

- on cherche le pivot p_k tq $a_{p_k,k} \neq 0$ et $p_k \neq p_j$, $j < k$
- Pour tout i tq $i \neq p_j$, $j \leq k$ (ie une ligne n'ayant pas ete pivot)

$$l_i \leftarrow l_i - \frac{a_{i,k}}{a_{p_k,k}} l_{p_k}$$

a l'etape finale, on permute les lignes pour obtenire un systeme triangulaire.

3.4.2 Exemple

$$y + z = 1 \tag{3.13}$$
$$x + y + z = 1 \tag{3.14}$$
$$x + y - z = 1 \tag{3.15}$$

On utilise donc la deuxieme ligne comme pivot: $l_3 \leftarrow l_3 - l_2$

$$y + z = 1 \tag{3.16}$$
$$x + y + z = 1 \tag{3.17}$$
$$-z = 0 \tag{3.18}$$

puis on permute la ligne 1 et 2 : $l_1 \leftrightarrow l_2$

$$x + y + z = 1 \tag{3.19}$$
$$y + z = 1 \tag{3.20}$$
$$-z = 0 \tag{3.21}$$

La permutation des lignes correspond a la multiplication par une matrice de permuation P. On a donc ici une decomposition PLU plutot que LU.

3.5 Decomposition PLU

Théorème 4. *Soit A inversible. Alors A admet une decomposition PLU.*

Proposition 14. *Erreur numerique:*
avoir un pivot proche de 0 entraine des erreurs numeriques. On choisit donc quand c'est possible un pivot plus adapte quite a introduire une permutation.

3.5.1 Algorithme

- Initialisation: $U = A$, $L = 0$, $P^T = I_n$
- Pour $1 \leq k < n$:
  - On cherche $j \geq k$ tq $u_{k,j} = \max\{u_{k,l} | l \geq k\}$
  - sur les matrices U, L et P^T on echange les lignes k et j
  - Sur la matrice U, on effectue les operations
    $$l_i \leftarrow l_i - u_{i,k}/u_{k,k} l_k$$
    pour $k < i \leq n$ ou l_i designe la i-eme ligne de la matice U
  - Sur la matrice L,
    $$l_{i,k} = u_{i,k}/u_{k,k}$$
    pour $k < i \leq n$
    (attention, ici $l_{i,j}$ designe le coef en position (i, j) de la matrice L, alors que precedement, l_i designait la i-eme ligne de U)
- etape finale: $L = L + I_n$

3.6 Complexite

Théorème 5. *la complexite de la methode du pivot de Gauss est* $O(n^3)$

a l'etape k, on fait $(n-k)$ additions de lignes ie $n(n-k)$ operations. d'ou complexite $\sum_{k=1..n} n(n-k) = O(n^3)$

Chapter 4

Résolution de système linéaires par des méthodes itératives

L'utilisation de méthodes directes pour la résolution des systèmes linéaires permettent d'obtenir des solutions exactes du système (aux erreurs d'arrondis près). Cependant, ces méthodes présentent l'inconvénient d'occuper beaucoup d'espace en mémoire. D'autre part, le nombre d'opérations nécessaire pour la résolution d'un système augmente rapidement avec la taille du système.
Le but de ce chapitre est d'étudier des méthodes de résolution de systèmes linéaires basés sur la construction de suites vectorielles convergeant vers la solution exacte du système.

4.1 Theoreme de Schur

Théoreme 1 (Théorème de Schur)**.** *Soit $A \in \mathcal{M}_n(\mathbb{C})$ une matrice quelconque, alors il existe une matrice unitaire $U \in \mathcal{M}_n(\mathbb{C})$ (c'est a dire $U^* = U^{-1}$) telle que*

$$T = U^* A U,$$

où T est une matrice triangulaire inférieure dont la diagonale est composée de l'ensemble des valeurs propres de A.

4.2 Principe général

On considère un système linéraire de la forme

$$Ax = b.$$

L'objectif des méthodes itératives est de construire une suite $(x^{(k)})_{k \in N}$ qui converge vers la solution exacte du système qui sera un point fixe. En se donnant un vecteur d'initialisation $x^{(0)}$, on considère la suite

$$x^{(k+1)} = F(x^{(k)}),\ k \in \mathbb{N}$$

où la fonction F sera exprimé dans la suite.
En décomposant la matrice A sous la forme

$$A = M - N,$$

où M est une matrice inversible, la résolution du système de ré-écrit

$$Mx = Nx + b$$

ou encore

$$x = M^{-1}Nx + M^{-1}b = F(x)$$

où F est une fonction affine.
On c'est donc ramene a un probleme de point fixe:

$$F(x) = x$$

Pour ce type de probleme, la suite $F(...F(x_0)...)$ converge frequement vers la solution. Pour $x^{(0)}$ donné, la suite $(x^{(k)})_{k\in N}$ est donc définie par

$$x^{(k+1)} = F(x^{(k)}) = M^{-1}Nx^{(k)} + M^{-1}b.$$

Tout le problème de la construction de méthodes itératives réside dans le choix des matrices M et N de manière à assurer la convergence de la méthode et obtenir une convergence rapide. Une dernière difficulté consiste à choisir un critère d'arrêt permettant d'obtenir une solution approchée suffisament proche de la solution exacte.

Définition 12. *Soit $A = M - N \in \mathcal{M}_n(\mathbb{K})$, où M est une matrice inversible. On dit que la méthode itérative est convergente si pour tout $b \in \mathbb{K}^n$ et pour tout $x^{(0)} \in \mathbb{K}^n$, la suite $(x^{(k)})_{k\in\mathbb{N}}$ converge vers la solution $x = A^{-1}b$ dans $\mathbb{K}^n$, càd que pour une norme de $\mathbb{K}^n$ donnée on a*

$$\lim_{k\to+\infty} \|x^{(k)} - x\| = 0.$$

Cette définition étant très générale, il est nécessaire d'établir un cirtère permettant d'établir la convergence de la méthode. Pour cela, on introduit l'erreur $e^{(k)}$ entre la solution approchée $x^{(k)}$ et la solution exacte x:

$$e^{(k)} = x(k) - x.$$

Proposition 15. *Soit x la solution exacte du systeme $Ax = b$. Alors:*

$$e^{(k)} = x^{(k)} - x = (M^{-1}N)^k(x^{(0)} - x)$$

Proof:

$$\begin{aligned} e^{(k)} &= x^{(k)} - x \\ &= M^{-1}Nx^{(k-1)} + M^{-1}b - M^{-1}(Nx - b) \\ &= M^{-1}Ne^{(k-1)} \end{aligned}$$

On a le resultat par recurrence.

Par conséquent, la méthode itérative sera convergente dès que

$$\lim_{k\to+\infty} \|(M^{-1}N)^k\| = 0.$$

Pour cela, la matrice $M^{-1}N$ devra vérifier le théorème suivant:

Théoreme 2. *Soit $B \in \mathcal{M}_n(\mathbb{C})$ alors les 4 propriétés suivantes sont équivalentes:*

1. $\lim_{k\to+\infty} \|B^k\| = 0$ *pour n'importe quelle norme*
2. $\lim_{k\to+\infty} B^k v = 0_{\mathbb{C}^n}$ *pour tout* $v \in \mathbb{C}^n$.
3. *Le rayon spectral de B vérifie*
$$\rho(B) < 1.$$
4. *Il existe une norme matricielle* $\|.\|_B$ *(dont le choix dépend de B) telle que* $\|B\|_B < 1$.

Proof:
$1 \Rightarrow 2$
Supposons que $\lim_{k\to+\infty} \|B^k\| = 0$ pour une norme quelconque. On a donc
$$0 \leq \|B^k v\| \leq \|B^k\|\|v\|.$$
D'où
$$\lim_{k\to+\infty} \|B^k v\| = 0_{\mathbb{K}^n}$$
$2 \Rightarrow 3$
Soit $\lambda \in Sp(B)$ et w un vecteur propre associé à la valeur propre λ, ie $Bw = \lambda w$.
On a
$$B^k w = B^{k-1} w(Bw) = B^{k-1}(\lambda w) = ... = \lambda^k w.$$
Comme $\lim_{k\to+\infty} B^k w = 0_{\mathbb{K}^n}$, on a
$$\lim_{k\to+\infty} \lambda^k w = 0$$
où w est indépendant de k et contient au moins une composante non nulle d'où
$$\lim_{k\to+\infty} \lambda^k = 0$$
d'où $|\lambda| < 1$, donc $\rho(B) < 1$.

$3 \Rightarrow 4$ Supposons que pour toute valeur propre λ de B vérifie $|\lambda_i| < 1$, $i = 1, ..., n$. D'après le théorème de Shur, il existe une matrice unitaire U telle que
$$T = U^* B U = \begin{pmatrix} \lambda_1 & t_{1,2} & \cdots & t_{1,n} \\ & \ddots & \ddots & \vdots \\ 0 & & \lambda_{n-1} & t_{n-1,n} \\ & & & \lambda_n \end{pmatrix}$$
On introduit la matrice diagonale
$$D = diag(1, \delta, ..., \delta^{n-1}).$$
où $\delta > 0$ est supposé petit. On a
$$D^{-1} = diag(1, \delta^{-1}, ..., \delta^{1-n}).$$

et

$$D^{-1}TD = \begin{pmatrix} \lambda_1 & \delta t_{1,2} & \cdots & \delta^n t_{1,n} \\ & \ddots & \ddots & \vdots \\ 0 & & \lambda_{n-1} & \delta t_{n-1,n} \\ & & & \lambda_n \end{pmatrix}$$

En choisissant δ suffisament petit, on a $\|D^{-1}TD\|_1 < 1$, càd

$$\|D^{-1}U^*BUD\|_1 < 1.$$

On note $\|.\|_B$ l'application qui a une matrice $A \in \mathcal{M}_n(\mathbb{R})$ associe

$$\|A\|_B = \|D^{-1}U^*AUD\|_1$$

On peut vérifier que $\|.\|_B$ est une norme matricielle:

- $\|A\|_B = 0 \Rightarrow D^{-1}U^*AUD = 0 \Rightarrow A = 0$ car D et U sont inversibles.
- $\|\alpha A\| = |\alpha|\|A\|$
- $\|A_1 + A_2\|_B = \|D^{-1}U^*A_1UD + D^{-1}U^*A_2UD\|_1 \leq \|A_1\|_B + \|A_2\|_B$
- $\|A_1A_2\|_B = \|(D^{-1}U^*A_1UD)(D^{-1}U^*A_2UD)\|_1 \leq \|A_1\|_B\|A_2\|_B$

De plus, $\|B\|_B < 1$.
On peut noter que $\|.\|_B$ est associee a la norme vectoriel $\|x\|_B = \|D^{-1}U^*x\|_1$.
$4 \Rightarrow 1$ Supposont que pour une norme donnée $\|.\|_B$ nous avons $\|B\|_B < 1$, alors comme en dimension finie toutes les normes sont équivalentes, il existe deux constantes $C_1 > 0$ et $C_2 > 0$ telles que pour toute norme subordonnée quelconque $\|.\|$ et pour tout $k > 0$

$$C_1\|B^k\| < \|B^k\|_B \leq C_2\|B^k\|$$

D'où

$$\|B^k\| \leq \frac{\|B^k\|_B}{C_1} \leq \frac{\|B\|_B^k}{C_1}$$

Comme $\|B\|_B < 1$,

$$\lim_{k\to+\infty} \|B^k\| = 0.$$

Théorème 6. *Soit $B \in \mathcal{M}_n(\mathbb{C})$ tel que $\rho(B) < 1$. Alors pour une norme quelconque $\|.\|$,*

$$\|B^k v\| \leq C\rho(B)^k$$

ou C est une constante dependant de B et de la norme $\|.\|$ choisie.

Proof: cf TD

4.3 Méthode de Jacobi

La méthode de Jacobi consiste à décomposer la matrice A de la manière suivante

$$A = D - E - F$$

où

- D est la diagonale de A
- $-E$ la partie inférieure de la matrice
- $-F$ la partie supérieure

et on pose $M = D$, et $N = E + F$. L'algorithme itératif s'écrit

$$\left\{ \begin{array}{l} x^{(0)} \in \mathbb{K}^n \\ Dx^{(k+1)} = (E+F)x^{(k)} + b \end{array} \right.$$

que l'on peut réécrire sous la forme

$$\left\{ \begin{array}{l} x^{(0)} \in \mathbb{K}^n \\ a_{i,i}x_i^{(k+1)} = -\sum_{j<i} a_{i,j}x_j^{(k)} - \sum_{j>i} a_{i,j}x_j^{(k)} + b_i. \end{array} \right.$$

On a le résultat de convergence suivant

Théoreme 3. *Soit $A \in \mathcal{M}_n(\mathbb{K})$ une matrice à diagonale strictement dominante,*

$$|a_{i,i}| > \sum_{i \neq j} |a_{i,j}|, \forall i \in \{1, ..., n\}$$

alors pour tout $x^{(0)}$, la méthode de Jacobi converge vers la solution x du système linéaire.

Proof: D'après le résultat de convergence du paragraphe précédent, il suffit de montrer qu'il existe une norme matricielle subordonnée $\|B\|$ telle que la matrice

$$B = D^{-1}(E+F)$$

vérifie $\|B\| < 1$. Or on a

$$\|B\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^{n} |b_{i,j}|.$$

De plus, on sait que

$$b_{i,j} = \left\{ \begin{array}{ll} 0 & \text{si } i = j \\ -\frac{a_{i,j}}{a_{i,i}} & \text{si } i \neq j \end{array} \right.$$

Comme A est à diagonale strictement dominante, on vérifie que

$$\sum_{j=1}^{n} |b_{i,j}| = \frac{\sum_{j=1, j\neq i}^{n} |a_{i,j}|}{a_{i,i}} < 1$$

donc $\|B\|_\infty < 1$ et la méthode de Jacobi est convergente.

4.4 Méthode de Gauss-Seidel

Proposition 16. *Soit $A \in M_n(\mathbb{R})$ une matrice symetrique. Il y a equivalence entre:*

- *A est definie positive*
- *pour tout x non nul, $x^T Ax > 0$*
- *les valeurs propres de A sont strictement positives (elles sont forcement reeles car A est symetrique)*
- *$(x, y) \to (Ax, y) = x^T Ay$ est un produit scalaire*

Proposition 17. *Si $A \in M_n(\mathbb{R}) = (a_{i,j})_{i,j}$ est symetrique definie positive alors pour tout i:*

$$a_{i,i} > 0$$

On utilise la même décomposition que pour la méthode de Jacobi, càd

$$A = D - E - F.$$

On pose $M = D - E$ et $N = F$. La méthode de Gauss-Seidel s'écrit

$$(D - E)x^{(k+1)} = Fx^{(k)} + b.$$

Dans ce cas, $D - E$ est une matrice triangulaire inférieure et est facilement inversible en utilisant un algorithme de descente. On a

$$\begin{cases} x^{(0)} \in \mathbb{K}^n \\ a_{i,i}x_i^{(k+1)} = -\sum_{j<i} a_{i,j}x_j^{(k+1)} - \sum_{j>i} a_{i,j}x_j^{(k)} + b_i. \end{cases}$$

On a le résultat de convergence suivant

Théoreme 4. *Soit $A \in \mathcal{M}_n(\mathbb{R})$ une matrice symétrique définie positive. Alors:*

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

Proof: Il suffit de montrer qu'il existe une norme matricielle subordonnée $\|.\|$ telle que

$$\|M^{-1}N\| < 1.$$

Comme A est une matrice symétrique définie positive, on peut considérer la norme vectorielle

$$\|x\| = \sqrt{(Ax, x)}$$

et on peut par conséquent calculer la norme matricielle subordonnée de $M^{-1}N$

$$\|M^{-1}N\| = \sup_{v \in \mathbb{R}^n} \frac{\|M^{-1}Nx\|^2}{\|x\|^2} = \sup_{v \in \mathbb{R}^n} \frac{(AM^{-1}Nx, M^{-1}Nx)}{(Ax, x)}$$

En écrivant $N = M - A$ et $y = M^{-1}Ax$,

$$\begin{aligned} (AM^{-1}Nx, M^{-1}Nx) &= (A(I_n - M^{-1}A)x, (I_n - M^{-1}A)x) \\ &= (Ax, x) - (Ax, y) - (Ay, x) + (Ay, y) \end{aligned}$$

Comme A est symétrique

$$\|M^{-1}Nx\|^2 = (Ax, x) - 2(Ax, y) + (Ay, y) \geq 0$$

Pour montrer que

$$\sup_x \frac{\|M^{-1}Nx\|^2}{\|x\|^2} < 1$$

il nous faut prouver $\sup_x -2(Ax, y) + (Ay, y) < 0$.
Puisque $Ax = My$

$$-2(Ax, y) + (Ay, y) = -2(My, y) + (Ay, y).$$

Or

$$\begin{aligned}(My, y) &= (My)^T y \\ &= y^T M^T y \\ &= y^T (M^T y) \\ &= (y, M^T y) \\ &= (M^T y, y)\end{aligned}$$

et donc

$$\begin{aligned}-2(My, y) + (Ay, y) &= -2(My, y) + ((M - N)y, y) \\ &= -(My, y) - (Ny, y) \\ &= -(M^T y, y) - (Ny, y) \\ &= -((M^T + N)y, y)\end{aligned}$$

On $M^T + N = D$ car A est symetrique. De plus, les elements diagonaux de A sont strictement positifs car A est definie positive. Donc D est définie positive et $(Dy, y) > 0$.

$$\begin{aligned}\frac{-2(My, y) + (Ay, y)}{\|x\|^2} &= \frac{-(Dy, y)}{\|x\|^2} \\ &\leq \frac{-\min_i(d_{i,i})\|y\|_2^2}{\|A^{-1}My\|^2} \\ &\leq \frac{-\min_i(d_{i,i})\|y\|_2^2}{\|A^{-1}M\|^2\|y\|^2}\end{aligned}$$

Comme les normes sont equivalentes, il existe $C > 0$ tel que $\|y\|_2 \leq C\|y\|$.

$$\begin{aligned}\frac{-2(My, y) + (Ay, y)}{\|x\|^2} &\leq \frac{-\min_i(d_{i,i})C^2\|y\|^2}{\|A^{-1}M\|^2\|y\|^2} \\ &\leq \frac{-\min_i(d_{i,i})C^2}{\|A^{-1}M\|^2} \\ &< 0\end{aligned}$$

donc

$$\sup_x \frac{\|M^{-1}Nx\|^2}{\|x\|^2} < 1 - \frac{\min_i(d_{i,i})C^2}{\|A^{-1}M\|^2}$$

et finalement

$$\|M^{-1}N\| \leq 1 - \frac{\min_i(d_{i,i})C^2}{\|A^{-1}M\|^2} < 1$$

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

Théoreme 5. *Si A est une matrice symetrique définie positive, alors pour tout $x^{(0)}$ la méthode de Gauss Seidel est bien définie et converge vers la solution x du système $Ax = b$.*

Proof: D'après le theoreme précédent $\rho(M^{-1}N) < 1$. Il reste donc a verifier uniquement l'inversibilite de M.
puisque M est une matrice triangulaire, nous avons

$$det(M) = det(D) = \prod_{i=1}^{n} a_{i,i}$$

et puisque A est définie positive, tous les termes diagonaux sont positifs.

4.5 Gradient Conjugue

L'algorithme du gradient conjugue s'applique aux matrice symetrique definie positive et est generalement considere comme le plus rapide pour ces matrices. Il est frequement utilise en conjonction avec Gauss-Siedel.

4.5.1 preliminaires

Dans toute la suite, A designe une matrice symetrique definie positive. On definie le produit scalaire associe a A:

$$< x, y >_A = x^T Ay$$

Soit $p_1, ...p_n$ une base orthogonale pour $< ., . >_A$. On note x^* la solution exacte du systeme. il existe $\alpha_1, ..., \alpha_n \in \mathbb{R}^n$ tel que

$$x^* = \alpha_1 p_1 + ... + \alpha_n p_n$$

Et comme la base est orthogonale:

$$\alpha_k = \frac{< x^*, p_k >_A}{< p_k, p_k >_A}$$

Or

$$< x^*, p_k >_A = < p_k, x^* >_A = p_k^T Ax^* = p_k^T b = < p_k, b >$$

Donc

$$\alpha_k = \frac{< p_k, b >}{< p_k, p_k >_A}$$

4.5.2 Methode

On va donc chercher a construire iterativement une tel base p_k, donnant si possible une bonne approximation de x^* en calculant uniquement les k premiers termes ($x = \alpha_1 p_1 + ... + \alpha_k p_k$).

Principe: pour resoudre $Ax = b$ on introduit

$$f(x) = 1/2 < Ax, x > - < b, x >$$

On remarque que $grad_f(x) = Ax - b$ donc le minimum de f est atteint pour la solution de $Ax = b$. La resolution du systeme se transforme donc en un probleme de minimisation de f. Cette fonction f sera donc utilisee pour choisir p_k assurant une convergence rapide.

On construit ici en meme temps

- une suite x_k qui converge vers x^*
- une suite $r_k = -grad_f(x_k)$
- une suite $p_1, ..., p_k$ de vecteurs orthogonaux 2 a 2 (on dit aussi qu'ils sont conjugues) construite grace aux r_k et tel que p_k soit orthogonale a r_{k-1}

Au pas k, on suppose que l'on a calcule $x_0, .., x_k$, $r_0, ...r_k$ et $p_0, ...p_{k-1}$. Si $r_k = 0$ alors $grad_f(x_k) = 0$ et donc $x_k = x^*$. Sinon on calcule x_{k+1}, r_{k+1} et p_k de la maniere suivante:

- $x_{k+1} = x_k + \rho_k p_k$ avec $\rho_k = \frac{(r_k, p_k)}{(Ap_k, p_k)}$
- $r_{k+1} = -grad_f(x_{k+1}) = b - Ax_{k+1}$
- $p_k = r_k + \alpha_{k-1} p_{k-1}$ et $\alpha_{k-1} = -\frac{(r_k, Ap_{k-1})}{(p_{k-1}, Ap_{k-1})}$

Lemme 1. *On peut alors prouver par recurrence:*

1. $(r_k, r_j) < 0$ *pour* $0 \leq j < k$
2. $(r_k, p_j) = 0$ *pour* $0 \leq j < k$
3. (p_{k-1}, Ap_j) *pour* $0 \leq j < k - 1$

Proof: Supposons que 1), 2) et 3) soient vrais pour k. Montrons que cela reste vrais pour $k + 1$:
3)Prouvons pour commencer $(p_k, Ap_{k-1}) = 0$.

$$\begin{aligned}(p_k, Ap_{k-1}) &= (r_k + \alpha_{k-1} p_{k-1}, Ap_{k-1}) \\ &= (r_k, Ap_{k-1}) - \frac{(r_k, Ap_{k-1})}{(p_{k-1}, Ap_{k-1})}(p_{k-1}, Ap_{k-1}) \\ &= 0\end{aligned}$$

Soit $0 \leq j < k-1$.

$$\begin{aligned}(p_k, Ap_j) &= (r_k + \alpha_{k-1}p_{k-1}, Ap_j) \\ &= (r_k, Ap_j)\end{aligned}$$

Or

$$\begin{aligned}r_{j+1} - r_j &= A(x_{j+2} - x_{j+1}) \\ &= \rho_j Ap_j\end{aligned}$$

Donc

$$\begin{aligned}(p_k, Ap_j) &= \rho_j(r_k, r_{j+1} - r_j) \\ &= 0\end{aligned}$$

2) Soit $0 \leq j < k+1$
Comme $r_{k+1} = r_k - \rho_k Ap_k$,

$$(r_{k+1}, p_j) = (r_k - \rho_k Ap_k, p_j)$$

D'apres l'hypothese de recurrence, si $j < k$, $(r_k, p_j) = 0$ et en utilisant ce qui vient d'etre montre en **3)** :$(Ap_k, p_j) = 0$ on obtient $(r_{k+1}, p_j) = 0$.
Si $j = k$,

$$\begin{aligned}(r_{k+1}, p_k) &= (r_k, p_k) - \frac{(r_k, p_k)}{(Ap_k, p_k)}(Ap_k, p_k) \\ &= 0\end{aligned}$$

1)
Soit $0 \leq j < k+1$. Comme $r_j = p_j - \alpha_{j-1}p_{j-1}$,

$$(r_{k+1}, r_j) = (r_{k+1}, p_j - \alpha_{j-1}p_{j-1})$$

En utilisant **2)** qui vient d'etre demontre, on obtient $(r_{k+1}, r_j) = 0$ ce qui conclut la preuve.

Théorème 7. *Le gradient conjugue converge en au plus n iterations (ou n est la dimension du systeme).*

Proof: En effet tant que $r_k \neq 0$ les $p_1, ..., p_k$ sont ortogonaux et non nuls, donc libres. donc Dans le pire des cas, on arrive a $p_0, ..., p_{n-1}$ qui forme alors une base de $\mathbb{R}^n$.
Si r_n est non nul, d'apres le lemme precedant, r_n est orthogonal aux $p_0, ..., p_{n-1}$ qui forment une base ce qui implique $r_n = 0$. contradiction.

Chapter 5

Résolution numérique de problèmes non linéaires

Soit $f : \Omega \subset \mathbb{R}^n \to \mathbb{R}^n$ une fonction non linéaire. L'objectif de ce chapitre est de résoudre numériquement le problème

$$f(x) = 0$$

par un algorithme itérative. En d'autre termes, nous allons construire une suite convergeant vers la solution du problème.
En 1D, pour calculer la racine carré d'un nombre positif α on pourra résoudre

$$x^2 - \alpha = 0.$$

5.1 Métodes itératives dans $\mathbb{R}$

Soit $f : [a, b] \to \mathbb{R}$ une fonction continue. On s'intéresse au problème

$$(P) \begin{cases} \text{Trouver } x^* \in [a, b] \\ f(x^*) = 0. \end{cases}$$

Définition 13. *Soit $(x_k)_{k\in\mathbb{N}}$ une suite approchant une solution du problème (P) et obtenue par une méthode itérative. On dit que la suite $(x_k)_{k\in\mathbb{N}}$ converge vers x^* si*

$$\lim_{k\to\infty} |x_k - x^*| = 0.$$

La méthode itérative est dite d'ordre p s'il existe $\gamma > 0$ tel que

$$\lim_{k\to+\infty} \frac{|x^{k+1} - x^*|}{|x^k - x^*|^p} = \gamma > 0.$$

On parlera de convergence quadratique lorsque $p = 2$.

La notion de vitesse de convergence est très importante puisqu'elle va en parti déterminer le temps de calcul global de la méthode utilisée. On essaiera de privilégier les méthodes d'ordre le plus élevé possible qui permettent cependant de garder une difficulté de mise en oeuvre acceptable.

Définition 14. *Soit $(x_k)_{k\in\mathbb{N}}$ une suite d'approximation de la solution x^* avec x_0 donné. On dira que*

- *la suite converge globalement vers x^* si pour tout $x_0 \in [a,b]$ la suite $(x_k)_{k\in\mathbb{N}}$ converge vers x^**
- *la suite $(x_k)_{k\in\mathbb{N}}$ converge localement vers x^* s'il existe un voisinage V de x^* tel que pour tout $x_0 \in V$ la suite x_k converge vers x^*.*

L'avantage des méthodes convergeant globalement est qu'elles ne nécessitent pas d'avoir une idée sur la solution du problème.

5.1.1 Méthode de dichotomie

La méthode de dichotomie est la plus simple et la plus intuitive. Elle n'est par contre pas la plus performante.
Soit $f : [a,b] \to \mathbb{R}$ une fonction continue telle que

$$f(a)f(b) < 0.$$

D'après le théorème des valeurs intermédiaires, il existe $x^* \in [a,b]$ tel que $f(x^*) = 0$.
On pose $m = \dfrac{a+b}{2}$, alors

- si $f(m) = 0$, $x^* = m$
- si $f(a)f(m) < 0$, $b = m$
- si $f(m)f(b) < 0$, $a = m$.

On réitère le processus jusqu'à que l'une des conditions d'arrêt suivantes soit satisfaite:

- $|b-a| < \varepsilon$
- $|f(m)| < \varepsilon$.

function x=Dichotomie(f,a,b,eps,Nmax)
if $f(a) * f(b) \geq 0$ **then**
  disp('Error: $f(a)f(b) \geq 0$')
end if
k=0;
c=(a+b)/2;
fc=f(c);
while $abs(fc) > eps$ and $k < Nmax$ **do**
  k=k+1;
  if $f(a) * fc < 0$ **then**
    $b = c$;
  else
    $a = c$;
  end if
  $c = (a+b)/2$;
  $fc = f(c)$;

end while
endfunction

On peut montrer que cette méthode converge bien vers la solution x^*. De plus, elle présente l'avantage de ne nécessiter que la continuité de f.

5.1.2 Méthode du point fixe

L'équation $f(x) = 0$ peut être ré-écrite sous la forme

$$F(x) = x$$

en posant $F(x) = f(x) + x$.

Exemple 1. *$x^2 + x + 2 = 0$ peut s'écrire $x^2 + 2x - 2 = x$ ou $\sqrt{2 - 2x} = x$ ou $-2 + 2/x = x$. Le deuxième choix élimite la solution $x = -1$.*

Le processus itératif du point fixe est donné par

$$\begin{cases} x_0 \in \mathbb{R} \text{ donné} \\ x_{n+1} = F(x_n) \end{cases}$$

L'algorithme s'écrit:

function x=PointFixe(F,a,b,x0,eps,Nmax)
x=x0;
k=0;
fx=F(x)-x;
while $abs(fc) > eps$ and $k < Nmax$ **do**
k=k+1;
x=F(x);
fx=F(X)-X;
end while
endfunction

Cette algorithme très simple à mettre en oeuvre ne converge pas tout le temps. Pour s'assurer de la convergence, la fonction F doit vérifier des conditions particulières.

Définition 15.

- *On dit que x est un point fixe de F si et seulement si $F(x) = x$.*
- *Soient U une partie de $\mathbb{R}$ et $F : U \to \mathbb{R}$ une fonction. On dit que F est contractante sur U si et seulement s'il existe un réel $\lambda \in]0, 1[$ tel que*

$$|F(x) - F(y)| \leq \lambda |x - y|, \forall x, y \in U$$

On a alors

Théoreme 6. *Soit $F : [a, b] \to [a, b]$ une application contractante sur $[a, b]$, alors*

1. *F admet un unique point fixe $x \in [a, b]$*
2. *pour tout $x_0 \in [a, b]$, la suite $x_{n+1} = F(x_n)$ converge vers x.*

Théoreme 7. *Si F admet un point fixe x, si F est de classe $\mathcal{C}^1$ au voisinage de x et si $|F'(x)| < 1$ alors il existe un voisinage V de x tel que pour tout $x_0 \in V$, la suite $x_{n+1} = F(x_n)$ converge vers x.*

Remarque 3. *La convergence de la méthode du point fixe est linéaire.*
Cette méthode se généralise très facilement au cas des fonction à plusieurs variable $f : \mathbb{R}^n \to \mathbb{R}^n$.

5.1.3 Méthode de Newton

Soit $f : [a, b] \to \mathbb{R}$ une fonction de classe $\mathcal{C}^1(\mathbb{R})$. On note $x_0 \in [a, b]$ une valeur approchée de la solution du problème $f(x) = 0$. En utilisant un développement de Taylor, on a

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2}f"(\eta)(x - x_0)^2$$

avec $\eta \in [a, b]$.
En négligeant le terme d'ordre 2 et en choisissant x_0 suffisament proche de la solurion x, une nouvelle approximation x_1 est donnée par

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

En réitérant ce procédé, on construit une suite $(x_k)_{k\in\mathbb{N}}$ définie par

$$\begin{cases} x_0 \in [a, b] \text{ donné} \\ x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, k \geq 0. \end{cases}$$

On a le résultat de convergence suivant:

Théorème 8. *Soit $f : [a, b] \to \mathbb{R}$ une fonction de classe $\mathcal{C}^2$. Supposons qu'il existe $x \in [a, b]$ tel que $f(x) = 0$ et $f'(x) \neq 0$. Alors il existe $\delta > 0$ tel que pour tout $x_0 \in [x - \delta, x + \delta]$, la suite $(x_k)_{k\in\mathbb{N}}$ est bien définie et converge vers la solution $x \in [a, b]$. De plus il existe une constante $C > 0$ telle que*

$$|x_{k+1} - x| \leq C|x_k - x|^2.$$

On dira que la méthode de Newton converge au moins quadratiquement.

Exemple 2. *Considérons la fonction $f(x) = xe^{-x^2}$. On a $f'(x) = (1-2x^2)e^{-x^2}$ et la méthode de Newton s'écrit*

$$x_{n+1} = x_n - \frac{x_n}{1 - 2x_n^2}$$

En prenant $x_0 = 0.3$, on a

n	x_n	$f(x_n)$
0	0.3	0.274
1	$-6.58.10e^{-2}$	$-6.56.10^{-2}$
2	$5.76.10e^{-4}$	$5.76.10^{-4}$
3	$-3.82.10^{-10}$	$-3.82.10^{-10}$

En choisissant $x_0 = 0.5$, on a

n	x_n	$f(x_n)$
0	0.5	0.3894
1	-0.5	-0.3894
2	0.5	0.3894

```
function x=Newton(f,fprime,a,b,x0,eps,Nmax)
x=x0;
k=0;
fx=f(x);
dfx=dfx(x);
while abs(fc) > eps and k < Nmax do
  k=k+1;
  x=x-fx/dfx;
  fx=f(x);
  dfx=df(x);
end while
endfunction
```

La méthode de Newton est très efficace. Elle présente cepedant l'inconvenient d'avoir une idée de la solution pour initialiser x_0 et de connaitre la dérivée de la fonction f.
Pour le premier point, il est possible de se rapprocher de la solution en utilisant une méthode de Dichotomie ou une méthode de point fixe sur une ou deux itérations puis d'utiliser la méthode de Newton.
Si on ne connait pas une expression exacte de la dérifée de f, on peut écrire que

$$f'(x_n) \approx \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}.$$

On obtient alors la méthode de la sécante définie par

$$x_{n+1} = x_n - f(x_n)\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}.$$

Dans ce cas là, il est nécessaire de donner x_0 et x_1 pour initialiser l'algorithme.

5.2 Méthodes itératives dans $\mathbb{R}^d$

5.2.1 Méthode du point fixe

Soit $f : \mathbb{R}^d \to \mathbb{R}^d$ une fonction continue. La méthode du point fixe vu en 1D se généralise très facilement au cas multiD pour résoudre l'équation

$$f(x) = 0_{\mathbb{R}^d}.$$

En effet, si on se donne $x_0 \in \mathbb{R}^d$ et si on pose $F(x) = f(x) + x$, on construit la suite

$$x_{n+1} = F(x_n)$$

et on peut montrer que cette suite converge vers x solution de $f(x) = 0$.

Théorème 9. *Soit E un espace métrique complet, d une distance sur E et $F : E \to E$ une fonction strictement contractante, càd qu'il existe $k \in]0,1[$ tel que pour tout $x, y \in E$*

$$d(F(x), F(y)) < kd(x, y)$$

alors F admet un unique point fixe x et la suite définie par

$$\begin{cases} x_0 \in E \\ x_{n+1} = F(x_n) \end{cases}$$

converge vers x.

5.2.2 Méthode de Newton

Soit $f : \Omega \subset \mathbb{R}^n \to \mathbb{R}^n$ avec

$$f(x) = \begin{pmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_n(x) \end{pmatrix}$$

pour tout $x = (x_1, x_2, ..., x_n)^T \in \Omega$. On dit que f est différentiable ou dérivable s'il existe une application linéare $Df(x) \in \mathcal{L}(\mathbb{R}^n, \mathbb{R}^n)$ tel qu'on ait

$$f(x + h) = f(x) + Df(x)h + \|x\|\varepsilon(x), \text{ avec } \lim_{h \to 0} \varepsilon(h) = 0.$$

On peut montrer que $DF(x)$ peut être identifiée à la matrice Jacobienne

$$Df(x) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \vdots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}$$

Considérons le problème: Trouver $x \in \Omega$

$$f(x) = 0,$$

alors comme dans le cas 1D, on se donne $x_0 \in \Omega$ une approximation de x et en utilisant un développement de f au point x_0 on construit la suite

$$\begin{cases} x_0 \in \Omega \\ f(x_k) + Df(x_k)(x_{k+1} - x_k) = 0_{\mathbb{R}^n}, \ k \geq 0. \end{cases}$$

Pour tout k, nous devons

- calculer $Df(x_k)$
- résoudre le système linéaire

$$Df(x_{k+1}) = -f(x_k) + Df(x_k)x_k$$

qui se réécrit

$$x_{k+1} = x_n - [Df(x_k)]^{-1}x_k.$$

Comme dans le cas 1D, nous avons le résultat de convergence suivant:

Théorème 10. *Soit $f : \Omega \subset \mathbb{R}^n \to \mathbb{R}^n$ une fonction deux fois dérivable, dont la deuxième dérivée est continue. Supposons qu'il existe $x \in \Omega$ tel que $f(x) = 0$, et que $Df(x)$ est inversible alors*

1. *il existe $\delta > 0$ tel que pour tout $x_0 \in B(x, \delta)$, la suite $(x_k)_{k\in\mathbb{N}}$ est bien définie*
2. *la suite $(x_k)_{k\in\mathbb{N}}$ converge vers la solution $x \in \Omega$*
3. *il existe une constante $C > 0$ telle que*

$$\|x_{k+1} - x\| \leq C\|x_k - x\|^2$$

c"est à dire la convergence de la méthode est quadratique.

Chapter 6

décomposition QR

6.1 Factorisation de Cholesky

Théorème 11. *Soit $A \in M_n(\mathbb{R})$ symétrique définie positive. Alors il existe une unique matrice réel L triangulaire inférieur dont tous les éléments diagonaux sont strictement positifs tel que*

$$A = LL^t$$

Proof. A est symétrique définie positive donc admet une décomposition LU:

$$A = LU$$

Prouvons par récurrence $u_{ii} > 0$ pour tout i:
En examinant le produit LU, on trouve $a_{1,1} = u_{1,1}l_{1,1} = u_{1,1}$ (car $l_{i,i} = 1$ pour tout i). Comme A est symétrique définie positive, $a_{1,1} > 0$ donc $u_{1,1} > 0$.
On regarde A comme une matrice par bloques:
DEF A
A_k est donc dans $M_k(\mathbb{R})$ et est symétrique définie positive, donc $det(A_k) > 0$.
En utilisant la meme decomposition pour L et U, on trouve $A_k = L_kU_k$. Comme L et U sont triangulaire et que $l_{i,i} = 1$ pour tout i,

$$det(U_k) = det(A_k) > 0$$

Or

$$det(U_k) = u_{1,1}...u_{k,k}$$

et $u_{1,1}, ..., u_{k-1,k-1} > 0$ donc

$$u_{k,k} > 0$$

.

On pose $D = diag(\sqrt{u_{11}}, ..., \sqrt{u_{nn}})$ et $\tilde{L} = LD$ et $\tilde{U} = D^{-1}U$.
On a $A = \tilde{L}\tilde{U}$, et on cherche a prouver $\tilde{L}^t = \tilde{U}$.
Comme A est symétrique, $\tilde{L}\tilde{U} = \tilde{U}^t\tilde{L}^t$ et donc

$$(\tilde{U}^t)^{-1}\tilde{L} = \tilde{L}^t\tilde{U}^{-1}$$

Or $\tilde{U}^t$ est triangulaire inférieur donc $(\tilde{U}^t)^{-1}$ l'est aussi, et comme $\tilde{L}$ est aussi triangulaire inférieur, on déduit que $(\tilde{U}^t)^{-1}\tilde{L}$ est triangulaire inférieur.

Par un raisonnement similaire, $\tilde{L}^t\tilde{U}^{-1}$ est triangulaire supérieur.
Donc $\tilde{L}^t\tilde{U}^{-1}$ est a la fois triangulaire supérieur et inférieur, donc diagonale. De plus,

$$\tilde{L}^t\tilde{U}^{-1} = L(DU^{-1}D)$$

or la matrice $DU^{-1}D$ ne possède que des 1 sur ça diagonale, tout comme L. Donc

$$\tilde{L}^t\tilde{U}^{-1} = I_n$$

Donc

$$\tilde{L}^t = \tilde{U}$$

et donc

$$A = \tilde{L}\tilde{L}^t$$

Unicité:
Soient L_1 et L_2 triangulaires inférieur dont tous les éléments diagonaux sont positifs tel que

$$A = L_1L_1^t = L_2L_2^t$$

Alors

$$L_2^{-1}L_1 = L_2^t(L_1^t)^{-1}$$

Par un argument similaire, $L_2^{-1}L_1$ est diagonale, ie $\exists D$ tel que

$$L_1 = L_2D$$

Or

$$A = L_1L_1^t = L_2DD^tL_2^t = L_2D^2L_2^t$$

et $A = L_2L_2^t$, donc $L_2L_2^t = L_2D^2L_2^t$ et comme L_2 inversible,

$$D^2 = I_n$$

donc les coefficients de D sont soit 1 soit -1. Or les coefficients diagonaux de L_1 et L_2 sont positifs, et comme $L_1 = L_2D$, les coefficients de D sont donc tous égaux a 1, donc $D = I_n$ donc $L_1 = L_2$. □

Calculer la factorisation de Cholesky
Soit A une matrice symétrique définie positive et soit $A = LL^t$ ça factorisation de Cholesky. On a alors

$$a_{ij} = \sum_k l_{ik}l_{kj}^t = \sum_k l_{ik}l_{jk}$$

et comme L est triangulaire inférieur,

$$a_{ij} = \sum_{k\leq\min(i,j)} l_{ik}l_{jk}$$

On commence par déterminer la première colonne de L en fixant $i = 1$.

- $a_{11} = l_{11}^2$, et comme $l_{ii} > 0$, on trouve

$$l_{11} = \sqrt{a_{11}}$$

- $a_{1,j} = l_{11} l_{j1}$ donc $l_{j1} = \frac{a_{1j}}{l_{11}}$ pour $j > i$

Si les colonnes $1, ..., i-1$ de L ont été déterminées:

- $a_{ii} = \sum_{k=1}^{i} l_{ik}^2$ donc $l_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2}$
- $a_{ij} = \sum_{k=1}^{i} l_{ik} l_{jk}$ donc $l_{jk} = \frac{a_{ij} - \sum_{k=1}^{i-1} l_{ik} l_{jk}}{l_{ii}}$

6.2 Décomposition QR

Soit $A \in M_n(\mathbb{R})$ inversible. On cherche une décomposition

$$A = QR$$

avec Q orthogonale ($Q^t = Q^{-1}$) et R une matrice triangulaire supérieur dont les éléments diagonaux sont strictement positifs.
Pour cela, on considère la matrice $A^t A$. Celle-ci est symétrique, et de plus pour tout x non nul,

$$x^t (A^t A) x = (Ax)^t (Ax) > 0$$

car $Ax \neq 0$ (car A inversible). Donc $A^t A$ est symétrique définie positive, et possède donc une factorisation de Cholesky. Il existe L tel que

$$A^t A = L L^t$$

On a

$$A = ((A^t)^{-1} L) L^t$$

On pose donc $R = L^t$ et $Q = (A^t)^{-1} L$.
R est clairement triangulaire inférieur dont les éléments diagonaux sont strictement positifs grâce au propriétés de la factorisation de Cholesky.
Il reste donc a prouver Q orthogonale.

$$\begin{aligned} QQ^t &= (A^t)^{-1} L ((A^t)^{-1} L)^t \\ &= (A^t)^{-1} L L^t ((A^t)^{-1})^t \\ &= (A^t)^{-1} A^t A A^{-1} \\ &= I_n \end{aligned}$$

car pour toute matrice, $(A^t)^{-1} = (A^{-1})^t$. Donc Q est orthogonale. Cependant l'inversion de A^t est couteuse en calcule, alors que celle de L l'est bien moins car triangulaire. On exprime donc Q de la manière suivante:

$$\begin{aligned} Q &= (Q^{-1})^t \\ &= (((A^t)^{-1} L)^{-1})^t \\ &= (L^{-1} A^t)^t \\ &= A (L^{-1})^t \end{aligned}$$

Calculer décomposition QR

- calculer la factorisation de Cholesky de $A^t A = L L^t$
- prendre $Q = A(L^{-1})^t$ et $R = L^t$

6.3 Décomposition QR - matrice rectangulaire

Théorème 12. *Soit $A \in M_{p,n}(\mathbb{R})$ avec $p \geq n$. Il existe $Q \in M_{p,p}(\mathbb{R})$ orthogonale et $R \in M_{p,n}(\mathbb{R})$ triangulaire supérieur dont les $p-n$ dernières lignes sont nulles tel que*

$$A = QR = Q \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$$

Proof. Cf TD. □

Chapter 7

Méthode des moindres carrés linéaire

7.1 Introduction

FIGURE

Supposons que l'on ai des points de donne $(x_1, y_1), ...(x_n, y_n)$ et que l'on cherche la droite passant par ces points de donne. On pourrait essayer de résoudre le système pour les inconnus a et b

$$ax_1 + b = y_1$$
$$...$$
$$ax_n + b = y_n$$

Cependant ce système ne possède dans la plupart des cas pas de solution. Pour contourner ce problème, on cherche la droite la plus 'proche' de ces points de donne.
Pour définir proche, on utilise

$$\sum_{i=1}^{n} ((ax_i + b) - y_i)^2$$

on cherche donc la droite $y = ax + b$ la plus proche:

$$(a, b) = argmin_{a,b} \sum_{i=1}^{n} ((ax_i + b) - y_i)^2$$

En générale, la méthode des moindres carres s'applique aux problèmes surconditionnes, c'est a dire ou il y a plus d'équations que d'inconnus et donc a priori pas de solution. Ce type de problème s'écrit de la manière suivante:

$$Ax = b$$

avec $A \in M_{p,n}(\mathbb{R})$ et $p \geq n$ (p représente le nombre d'équations, n le nombre de degrés de liberté).
On cherche alors

$$x = argmin_{x \in \mathbb{R}^n} \|Ax - b\|^2$$

avec $\|x\|^2 = \sum_{i=1}^{n} x_i^2$ la norme euclidienne.

Définition 16. *Soient $A \in M_{p,n}(\mathbb{R})$ et $b \in \mathbb{R}^p$ avec $p \geq n$. On appelle problème des moindre carres le problème:*

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|$$

7.2 Équation normale

Soient $A \in M_{p,n}(\mathbb{R})$ et $b \in \mathbb{R}^p$

Définition 17. *On appelle equation normale l'équation*

$$A^t Ax = A^t b$$

Lemme 2. *x est solution du problème au moindre carres si et seulement si x est solution de l'équation normale.*

Proof.

$$\begin{aligned}
&\|Ax - b\|^2 \leq \|Ay - b\|^2 \, \forall y \\
&\Leftrightarrow \|Ax - b\|^2 \leq \|A(x + tz) - b\|^2 \, \forall z, \forall t \geq 0 \\
&\Leftrightarrow \|Ax - b\|^2 \leq \|Ax - b\|^2 + t^2\|Az\|^2 + 2t(Ax - b, Az) \, \forall z, \forall t \geq 0 \\
&\Leftrightarrow 0 \leq t^2\|Az\|^2 + 2t(Ax - b, Az) \, \forall z, \forall t > 0 \\
&\Leftrightarrow 0 \leq t\|Az\|^2 + 2(Ax - b, Az) \, \forall z, \forall t > 0 \\
&\Leftrightarrow (Ax - b, Az) \geq 0 \, \forall z \\
&\Leftrightarrow (Ax - b, Az) = 0 \, \forall z \text{ (on prend } z \text{ et } -z) \\
&\Leftrightarrow (A^t Ax - A^t b, z) = 0 \, \forall z \\
&\Leftrightarrow A^t Ax - A^t b = 0
\end{aligned}$$

□

Théorème 13. *L'équation normale admet une solution, et celle-ci est unique ssi $Ker(A) = 0$*

Proof. admis. □

7.3 Projection orthogonale

Théorème 14 (Théorème de la projection orthogonale)**.** *Soit E un espace vectoriel complet munit d'un produit scalaire et F un sous espace vectoriel de E de dimension finie. Soit $x \in E$.*
Il existe un unique $y^* \in F$ tel que

$$\|x - y^*\| = \inf_{y \in F} \|x - y\|$$

De plus y^* est la projection orthogonale de x sur F qui est tel que $(y^* - x, y) = 0$ pour tout $y \in F$.

Proof. cf TD □

Pour un problème de moindre carres, on pose $F = Im(A)$. On a alors

$$\inf_{x \in E} \|Ax - b\|^2 = \inf_{y \in F} \|y - b\|^2$$

Et l'on retrouve l'unicité quand $Ker(A) = 0$.

Décomposition QR
On utilise la décomposition QR de A:

$$A = QR = Q \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$$

Comme Q est orthogonale, Q^t ne change pas la norme:

$$\|Q^t y\| = \|y\|$$

pour tout y. Donc

$$\begin{aligned} \|Ax - b\|^2 &= \|Q^t(Ax - b)\|^2 \\ &= \|Rx - Q^t b\|^2 \end{aligned}$$

on pose $c = Q^t b$

$$\begin{aligned} \|Ax - b\|^2 &= \| \begin{bmatrix} R_1 x - c_1 \\ -c_2 \end{bmatrix} \|^2 \\ &= \|R_1 x - c_1\|^2 + \|c_2\|^2 \end{aligned}$$

Le vecteur minimisant $\|Ax - b\|^2$ est donc la solution de $R_1 x = c_1$

Algorithme

- Calculer la décomposition QR de A:

$$A = QR = Q \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$$

- Résoudre $R_1 x = c_1$

7.3.1 Choix de la méthode

- résoudre l'équation normale est plus rapide
- la décomposition QR est plus stable numériquement