



Faculté des Sciences Exactes et Informatique  
Département de Mathématiques

N° d'ordre : .....

N° de séries : .....

**Mémoire de fin d'étude**  
Présenté pour l'obtention du diplôme de

**Master**

**Spécialité : Mathématiques**  
**Option : EDP et Applications**

**Thème**

---

*Résolution d'un Problème de Minimisation  
Quadratique Convexe*

---

**Présenté par :**

M<sup>elle</sup> Rima BENREBAI.  
M<sup>elle</sup> Madjda MIMOUNE.

**Devant le jury composé de :**

Président	kECIS Ilyas	MCB	Université de Jijel
Encadreur	DJEMAI Samia	MCB	Université de Jijel
Examineur	KHELLAF Wahiba	MCB	Université de Jijel

Promotion **2017/2018**

✧ *Remerciements* ✧

*Nous tenons à remercier en premier lieu Dieu le Tout Puissant, qui nous a donné le courage et la volonté pour pouvoir réaliser ce modeste travail.*

*Nous exprimons notre gratitude à **M<sup>me</sup> DJEMAI Samia** pour avoir encadré et dirigé ce travail avec une grande patience, ses conseils, sa confiance et ses encouragements tout au long de ce travail.*

*Nous remercions également **M<sup>r</sup> KECIS Ilyas** et **M<sup>me</sup> KHELLAF Wahiba** d'avoir accepté de juger notre travail.*

*Nous remercions vivement nos familles pour leur patience, confiance et leur soutien moral et matérielle.*

*Merci à ceux qui n'ont pas hésité les moment critiques à nous apporter leur aide et encouragement.*

✧ *Dédicaces* ✧

*Je dédie ce modeste travail :*

✧ *À mes très chers parents pour tous leurs sacrifices et leurs soutien*

✧ *À mes frères : Samir, Amir, Mohamed, Djalal et Bilal*

✧ *À ma sœur : Sabra*

✧ *À mes oncles et tantes*

✧ *À ma meilleure amie et binôme Rima et sa famille*

✧ *À mes chères amies et toutes les personnes qui me chères*

Merci infiniment

***Madja***

✧ *Dédicaces* ✧

*Je dédie ce modeste travail :*

✧ *À mes très chers parents pour tous leurs sacrifices et leurs soutien*

✧ *À mes frères : Halim, Fouad, Fatah et Hamza*

✧ *À mes sœurs : Fatima, hanan, Dounia, Radia et Akila*

✧ *À les femmes de mes frères Sabrina et Amira*

✧ *À les enfants de mes frères et sœurs*

✧ *À mes oncles et tantes*

✧ *À ma meilleure amie et binôme Madjda et sa famille*

✧ *À mes chères amies et toutes les personnes qui me chères*

Merci infiniment

*Rima*

# Table des matières

<b>Table des matières</b>	<b>3</b>
<b>Liste des figures</b>	<b>4</b>
<b>Introduction Générale</b>	<b>5</b>
<b>1 Rappels mathématiques</b>	<b>7</b>
1.1 Introduction . . . . .	7
1.2 Les formes quadratiques . . . . .	7
1.2.1 Représentation matricielle d'une forme quadratique . . . . .	8
1.2.2 Gradient et Hessien d'une forme quadratique . . . . .	8
1.2.3 Propriétés des matrices définies positives et non négatives . . . . .	9
1.2.4 Noyau, image et rang d'une matrice . . . . .	10
1.2.5 La matrice Jacobienne . . . . .	10
1.2.6 L'inverse des matrices . . . . .	11
1.2.7 Formes quadratiques définies et non définies . . . . .	12
1.3 Convexité . . . . .	12
1.3.1 Ensemble convexe . . . . .	12
1.3.2 Fonction convexe . . . . .	13
<b>2 La programmation quadratique convexe (PQC)</b>	<b>16</b>
2.1 Introduction . . . . .	16

---

2.2	Programmation non linéaire avec contraintes . . . . .	16
2.2.1	Formulation du problème . . . . .	16
2.2.2	La méthode de Newton pour la minimisation d'une fonction différentiable . . . . .	18
2.2.3	Conditions d'optimalité pour le cas des contraintes linéaires de type égalités . . . . .	19
2.2.4	Conditions d'optimalité pour le cas des contraintes linéaires de type inégalités . . . . .	20
2.3	Programmation convexe . . . . .	21
2.3.1	Problème quadratique convexe (PQC) . . . . .	23
2.4	Dualité en programmation convexe . . . . .	23
<b>3</b>	<b>Méthodes de résolution en programmation quadratique convexe</b>	<b>25</b>
3.1	Introduction . . . . .	25
3.2	La méthode de points intérieurs . . . . .	25
3.2.1	Position du problème . . . . .	26
3.2.2	Problème dual . . . . .	26
3.2.3	Technique de la méthode . . . . .	27
3.2.4	La technique de chemin central . . . . .	29
3.2.5	Schéma de l'algorithme . . . . .	31
3.3	La méthode d'activation des contraintes . . . . .	31
3.3.1	Critère d'optimalité . . . . .	32
3.3.2	Itération de la méthode . . . . .	34
3.3.3	Schéma de l'algorithme . . . . .	35
3.3.4	Exemple numérique . . . . .	36
<b>4</b>	<b>Mise en œuvre</b>	<b>40</b>
4.1	Introduction . . . . .	40
4.2	Choix du langage . . . . .	40

---

4.2.1	Généralités sur le langage . . . . .	41
4.2.2	Programmation avec Matlab . . . . .	41
4.2.3	Générateur d'exemples . . . . .	42
4.3	La méthode d'activation des contraintes et La méthode de points intérieurs en Matlab . . . . .	42
4.3.1	Exemple d'exécution . . . . .	43
4.4	Comparaison entre les deux méthodes . . . . .	45
4.4.1	Discussion des résultats . . . . .	45
	<b>Conclusion Générale</b>	<b>46</b>
	<b>Bibliographie</b>	<b>46</b>

# Table des figures

1.1	Exemple d'ensemble convexe et non convexe . . . . .	13
1.2	Illustration d'une fonction convexe . . . . .	14



# Introduction Générale

L'OPTIMISATION QUADRATIQUE est l'une des théories de la programmation mathématique la plus utilisée pour modéliser des problèmes pratiques. C'est une théorie autonome de l'optimisation non linéaire dans lequel on minimise (ou maximise) une forme quadratique sous des contraintes linéaires.

Cette branche est très importante d'un point de vue pratique : de nombreux domaines d'application ont été touchés par cette théorie, notamment la recherche opérationnelle, l'économie, les sciences de l'ingénieur, et la physique.

L'optimisation quadratique est très intéressante du point de vue théorique, car elle est une transition naturelle entre la programmation linéaire (PL) et non linéaire. En effet, la majorité des méthodes développées pour le cas quadratique sont des extensions directes de celles de la PL. De plus, les algorithmes élaborés pour le cas de l'optimisation non linéaire reposent essentiellement sur l'approche quadratique.

La résolution numérique d'un problème quadratique nécessite l'utilisation des méthodes spécifiques. La plupart de ces méthodes sont itératives et génèrent une suite de vecteurs  $x_i$ . Les itérations successives sont construites de façon à faire converger cette suite vers la solution du problème  $x^*$ . On peut alors utiliser plusieurs méthodes pour la résolution de ces problèmes quadratiques. Dans ce travail, nous nous intéressons à deux méthodes de résolution d'un problème quadratique sous contraintes linéaires de type inégalité :

✓ *La méthode de points intérieurs* [15] : Après leur succès en programmation linéaire, elle a été généralisée pour le cas quadratique. Cette dernière est connue par une bonne complexité polynômiale et l'efficacité dans la résolution des problèmes quadratiques de grande taille.

✓ *La méthode d'activation des contraintes* en programmation quadratique a vu le jour grâce à *Fletcher* en 1971[7]. Par la suite, d'autres auteurs ont fait des raffinements numériques à cette dernière tels que *Gill et Al*, en 1978, ainsi que *Gould* en 1991. Elle est une généralisation de la méthode du simplexe et s'applique pour le cas des programmes

quadratiques convexes et non convexes.

Ce mémoire commence par une introduction qui s'articule autour des quatre chapitres. Dans le premier chapitre nous faisons quelques rappels mathématiques, certains résultats classiques concernant les fonctions quadratiques, ainsi que les notions de la convexité. Le deuxième chapitre comprend des définitions et des théorèmes sur la programmation quadratique convexe et la notion de la dualité. Le troisième chapitre consacre les méthodes de résolution d'un problème de programmation quadratique telles que la méthode d'activation des contraintes et celle de points intérieurs où on présente d'une manière détaillée le principe de ces deux méthodes. Dans le dernier chapitre, nous avons fait une étude numérique des deux méthodes, et ce, après avoir appliqué ces dernières sous le langage Matlab, exécutées sur des exemples du test. Enfin, ce mémoire s'achève par une conclusion générale.

# Chapitre 1

## Rappels mathématiques

### 1.1 Introduction

Dans ce chapitre, nous rappelons quelques définitions et propriétés des fonctions quadratiques, des matrices semi définies positives et des notions d'ensembles et fonctions convexes. Ceci nous sera utile dans l'étude ultérieure de la programmation quadratique convexe.

### 1.2 Les formes quadratiques

**Définition 1.1.** Une fonction  $F : \mathbb{R}^n \rightarrow \mathbb{R}$  est dite forme quadratique de  $n$  variables  $x_1, x_2, \dots, x_n$  si elle s'écrit sous la forme suivante :

$$F(x) = \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j + \sum_{j=1}^n c_j x_j = x'Ax + c'x. \quad (1.1)$$

Où  $x = (x_1, x_2, \dots, x_n)'$  et  $c$  sont des  $n$ -vecteurs colonnes ;  $A = (a_{ij}, 1 \leq i, j \leq n)$  est une matrice carrée d'ordre  $n$ .

Le symbole  $(')$  est l'opérateur de transposition vectorielle et matricielle.

Pour  $i \neq j$ , le coefficient du terme  $x_i x_j$  s'écrit  $a_{ij} + a_{ji}$ . En vertu de cela, la matrice  $A$  peut être supposée symétrique. En effet, en définissant de nouveaux coefficients

$$d_{ij} = d_{ji} = \frac{a_{ij} + a_{ji}}{2}, \quad 1 \leq i, j \leq n.$$

On obtient une nouvelle matrice  $D = (d_{ij}, 1 \leq i, j \leq n)$  symétrique. Il est clair qu'après une redéfinition des coefficients, la valeur de la forme quadratique  $F(x)$  reste inchangée :

$$F(x) = x'Ax + c'x = x'Dx + c'x, \quad \forall x \in \mathbb{R}^n.$$

Pour cela, il est naturel de considérer que la matrice d'une forme quadratique est toujours symétrique.

### 1.2.1 Représentation matricielle d'une forme quadratique

Soient  $x = (x_1, x_2, \dots, x_n)$  et  $D = (d_{ij}, 1 \leq i \leq n, 1 \leq j \leq n)$ .  
Pour  $n = 2$  et d'après la formule (1.1), on a

$$\begin{aligned} F(x) &= \sum_{i=1}^2 \sum_{j=1}^2 a_{ij} x_i x_j + \sum_{j=1}^2 c_j x_j \\ &= a_{11} x_1^2 + a_{12} x_1 x_2 + a_{22} x_2^2 + a_{21} x_2 x_1 + c_1 x_1 + c_2 x_2 \\ &= x_1 (a_{11} x_1 + a_{12} x_2) + x_2 (a_{21} x_1 + a_{22} x_2) + c_1 x_1 + c_2 x_2 \\ &= (x_1 \ x_2) \begin{pmatrix} a_{11} x_1 + a_{12} x_2 \\ a_{21} x_1 + a_{22} x_2 \end{pmatrix} + (c_1 \ c_2) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ &= (x_1 \ x_2) \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + (c_1 \ c_2) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ &= x' D x + c' x. \end{aligned}$$

**Exemple 1.2.** *Considérons la forme quadratique suivante :*

$$F(x) = 2x_1^2 + x_1 x_2 + x_2^2 - 12x_1 - 10x_2$$

La matrice symétrique associée est  $D = \begin{pmatrix} 4 & 1 \\ 1 & 2 \end{pmatrix}$  et  $c = \begin{pmatrix} -12 \\ -10 \end{pmatrix}$

### 1.2.2 Gradient et Hessien d'une forme quadratique

**Définition 1.3.** *Soit une fonction de classe  $C^1$   $F : \mathbb{R}^n \rightarrow \mathbb{R}$ . Le gradient de la fonction  $F$  est défini par :*

$$g(x) = \nabla F = \begin{bmatrix} \frac{\partial F}{\partial x_1} \\ \frac{\partial F}{\partial x_2} \\ \cdot \\ \cdot \\ \cdot \\ \frac{\partial F}{\partial x_n} \end{bmatrix} = 2Dx + c.$$

Où  $\frac{\partial F}{\partial x_i}$  est la dérivée partielle de  $F$  par rapport à  $x_i$ ,  $i = 1, 2, \dots, n$ .

**Définition 1.4.** Soit une fonction de classe  $C^2 F : \mathbb{R}^n \longrightarrow \mathbb{R}$ . Le hessien de la fonction  $F$  est défini par :

$$H(x) = \nabla^2 F(x) = \begin{bmatrix} \frac{\partial^2 F}{\partial x_1^2} & \frac{\partial^2 F}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 F}{\partial x_1 \partial x_n} \\ \frac{\partial^2 F}{\partial x_2 \partial x_1} & \frac{\partial^2 F}{\partial x_2^2} & \cdots & \frac{\partial^2 F}{\partial x_2 \partial x_n} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \frac{\partial^2 F}{\partial x_n \partial x_1} & \frac{\partial^2 F}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 F}{\partial x_n^2} \end{bmatrix}.$$

Où  $\frac{\partial^2 F}{\partial x_i^2}$  est la dérivée partielle d'ordre 2 de  $F$ .

### 1.2.3 Propriétés des matrices définies positives et non négatives

**Définition 1.5.** [17] Soit  $D$  une matrice carrée, on dit que  $D$  est symétrique si :  $d_{ij} = d_{ji}$  pour tout  $1 \leq i, j \leq n$  et  $i \neq j$ .

Les matrices symétriques définies ont des propriétés très intéressantes. En voici quelques unes :

**Propriétés 1.6.** Soit  $D$  une matrice symétrique  $n \times n$ . On note  $\{\lambda_i\}_{i=1, \dots, n}$  ses valeurs propres réelles. On a les équivalences suivantes :

- $D \geq 0 \Leftrightarrow \lambda_i \geq 0, i = 1, \dots, n$  ;
- $D > 0 \Leftrightarrow \lambda_i > 0, i = 1, \dots, n$  ;
- $D$  est indéfinie si et seulement si certains  $\lambda_i$  sont positifs et d'autres négatifs.

Lorsque la matrice  $D$  est définie positive (resp. semi-définie positive), on dira que  $F(x)$  est une forme quadratique définie positive (resp. semi-définie positive).

**Proposition 1.7.** Soit  $D$  une matrice symétrique  $n \times n$ . On dit que  $D$  est semi-définie positive ou définie non négative et on note  $D \geq 0$ , quand

$$xDx' \geq 0, \forall x \in \mathbb{R}^n \quad \text{et} \quad x \neq 0.$$

On dit que  $D$  est définie positive et on note  $D > 0$ , quand

$$xDx' > 0, \forall x \in \mathbb{R}^n \quad \text{et} \quad x \neq 0.$$

**Propriétés 1.8.** [12] Un élément de la diagonale d'une matrice symétrique  $D$  définie non négative ne peut s'annuler que si les autres éléments de la même ligne et colonne s'annulent aussi.

**Propriétés 1.9.** Soit la matrice  $D$  partitionnée de la manière suivante :

$$D = \begin{pmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{pmatrix}$$

Si  $D > 0$  ( $D \geq 0$ ), alors les sous matrices principales  $D_{11}$  et  $D_{22}$  sont aussi définies positives (non négatives). D'une manière générale, toute sous matrice principale d'une matrice définie positive (non négative) est définie positive (non négative).

**Propriétés 1.10.** [19] Soit  $D$  une matrice symétrique définie non négative. Si  $x$  est un point quelconque mais fixé de  $\mathbb{R}^n$  tel que  $x'Dx = 0$ , on a alors  $Dx = 0$ .

## 1.2.4 Noyau, image et rang d'une matrice

**Définition 1.11.** Soit  $A$  une matrice d'ordre  $m \times n$ .

Le noyau de  $A$  est l'ensemble :

$$N(A) = \{x \in \mathbb{R}^n : Ax = 0\}.$$

L'image ou l'espace image de  $A$  est l'ensemble :

$$R(A) = \{y \in \mathbb{R}^m : \exists x \in \mathbb{R}^n, Ax = y\}.$$

**Définition 1.12.** On appelle rang d'une matrice, la dimension de l'image de la matrice  $A$  :

$$\text{rang}(A) = \dim R(A)$$

## 1.2.5 La matrice Jacobienne

**Définition 1.13.** Soit une fonction  $F : (x_1, x_2, \dots, x_n) \longrightarrow \begin{pmatrix} y_1 \\ \cdot \\ \cdot \\ \cdot \\ y_m \end{pmatrix} = \begin{pmatrix} F_1(x) \\ \cdot \\ \cdot \\ \cdot \\ F_m(x) \end{pmatrix}$

On définit la matrice jacobienne associée à  $F$  de la manière suivante :

$$\mathcal{J}_F = \begin{pmatrix} \frac{\partial F_1}{\partial x_1} & \cdot & \cdot & \cdot & \cdot & \frac{\partial F_1}{\partial x_n} \\ \frac{\partial F_2}{\partial x_1} & \cdot & \cdot & \cdot & \cdot & \frac{\partial F_2}{\partial x_n} \\ \cdot & & & & & \\ \cdot & & & & & \\ \cdot & & & & & \\ \frac{\partial F_n}{\partial x_1} & \cdot & \cdot & \cdot & \cdot & \frac{\partial F_n}{\partial x_m} \end{pmatrix}$$

Si  $m = n$ , on appelle Jacobien de  $F$  le déterminant de sa matrice Jacobienne :

$$J = \det(\mathcal{J}_F).$$

### 1.2.6 L'inverse des matrices

**Définition 1.14.** *Une matrice carrée  $D$  est inversible si son déterminant est non nul, sinon elle est dite singulière.*

**Propriétés 1.15.**

1. *Pour chaque matrice carrée  $A$  d'ordre  $n$ , la matrice  $D = A'A$  est symétrique et définie non négative. La matrice  $D$  est définie positive si et seulement si  $A$  est inversible.*
2. *L'inverse d'une matrice symétrique définie positive est symétrique et définie positive.*

*Démonstration.* [13]

1. Soit  $A$  une matrice carrée d'ordre  $n$ . La matrice  $D = A'A$  est symétrique et définie non négative, car on a

$$D' = (A'A)' = D \quad \text{et} \quad F(x) = x'Dx = x'A'Ax = (Ax)'Ax = \|Ax\|^2 \geq 0.$$

La forme semi-définie  $F$  est définie positive si  $F(x) = 0 \Leftrightarrow x = 0$ .

En effet, on a

$$F(x) = \|Ax\|^2 = 0 \Leftrightarrow Ax = 0 \Leftrightarrow x = 0,$$

car  $A$  est inversible.

2. Comme  $x'Dx > 0, \forall x \neq 0$ , alors  $Dx = 0 \Rightarrow x = 0$ . Ce qui signifie que les colonnes de  $D$  sont indépendantes, donc le rang de  $D$  est complet et son inverse existe. Pour montrer que  $D^{-1}$  est définie positive, considérons la transformation  $y = Dx$  ou  $x = D^{-1}y$ . Chaque vecteur  $x$  correspond à un seul vecteur  $y$  et vice versa. De plus,  $y = 0$  si et seulement si  $x = 0$ . Pour tout vecteur  $y \neq 0$ , on a alors

$$y'D^{-1}y = x'DD^{-1}Dx = x'Dx > 0, \quad \text{car} \quad x = D^{-1}y \neq 0.$$

Par conséquent,  $D^{-1}$  est aussi définie positive.

□

### 1.2.7 Formes quadratiques définies et non définies

Soit la forme quadratique  $F(x) = x'Dx$ .

**Définition 1.16.**  $F(x)$  est dite définie positive si  $x'Dx > 0, \forall x \in \mathbb{R}^n$  et  $x \neq 0$ . Elle est dite semi-définie positive ou définie non négative si  $x'Dx \geq 0, \forall x \in \mathbb{R}^n$  et  $x \neq 0$ .

**Définition 1.17.**  $F(x)$  est dite définie négative si  $x'Dx < 0, \forall x \in \mathbb{R}^n$  et  $x \neq 0$ . Elle est définie non positive si  $x'Dx \leq 0, \forall x \in \mathbb{R}^n$  et  $x \neq 0$ .

**Définition 1.18.** Une matrice symétrique  $D$  est dite matrice définie positive (non négative) et se note  $D > 0, (D \geq 0)$ , si elle est associée à une forme quadratique définie positive (non négative).

**Définition 1.19.** Une forme quadratique  $F(x)$  est dite non définie si  $F(x)$  est positive pour certaines valeurs de  $x$  et négative pour d'autres.

## 1.3 Convexité

Dans les problèmes de minimisation avec ou sans contraintes, la convexité joue un rôle très important, nous allons rappeler brièvement les définitions de quelques notions importantes auxquelles nous ferons appel par la suite, ainsi que quelques propriétés.

### 1.3.1 Ensemble convexe

**Définition 1.20.** [6] Un ensemble  $S \subset \mathbb{R}^n$  est dit convexe si et seulement si pour deux points quelconques  $x$  et  $y$  on a :

$$\forall x \in S, \forall y \in S, \forall \lambda \in [0, 1] \Rightarrow \lambda x + (1 - \lambda)y \in S.$$

Géométriquement, pour tout segment reliant deux points quelconques  $x$  et  $y$  de  $S$ , le segment  $[x, y]$  doit être aussi dans  $S$ .

**Définition 1.21.** Un ensemble de points de  $\mathbb{R}^n$  est convexe si et seulement si toute combinaison convexe de points de l'ensemble est encore un point de l'ensemble.

**Théorème 1.22.** Soit  $S_1$  et  $S_2$  deux ensembles convexes de  $\mathbb{R}^n$ . Alors :

- $S_1 \cap S_2$  est convexe ;
- $S_1 \pm S_2 = \{x_1 \pm x_2 | x_1 \in S_1, x_2 \in S_2\}$  est convexe.



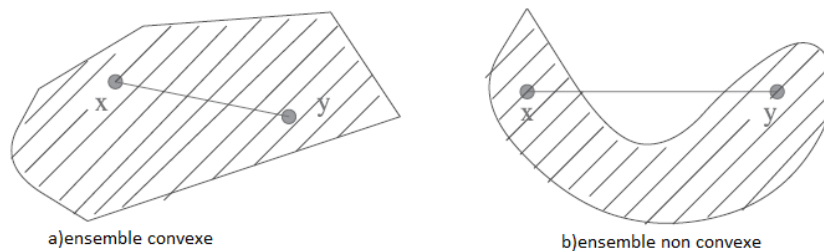


FIGURE 1.1 – Exemple d'ensemble convexe et non convexe

**Théorème 1.23.** Soit  $S \subset \mathbb{R}^n$  un ensemble convexe. Alors :

- L'intérieur de  $S$ , noté  $\text{Int}S$ , est un ensemble convexe ;
- La clôture  $\bar{S}$  de  $S$  est un ensemble convexe.

*Démonstration.*

- Soient  $x_1$  et  $x_2$  dans  $\text{Int}S$ , et  $x_3 = \lambda x_1 + (1 - \lambda)x_2$ ,  $\lambda \in [0, 1]$ .  
On choisit  $\delta > 0$  tel que  $B(x_2, \delta) \subset S$ , où  $B(x_2, \delta)$  est le  $\delta$ -voisinage de  $x_2$ .  
On a  $\|x_3 - x_1\| / \|x_2 - x_1\| = 1 - \lambda$ .

On sait que  $B(x_3, (1 - \lambda)\delta)$  est l'ensemble  $\lambda x_1 + (1 - \lambda)B(x_2, \delta) \subset S$ .

Donc  $B(x_3, (1 - \lambda)\delta) \subset S$ , d'où  $x_3 \in \text{Int}S$

- Prenons  $x_1, x_2 \in \bar{S}$ . Soient  $\{x_{k1}\}$  et  $\{x_{k2}\}$  deux suites convergentes vers  $x_1$  et  $x_2$  respectivement. Donc, pour  $\lambda \in [0, 1]$ , nous avons

$$\begin{aligned} \|\lambda x_{k1} + (1 - \lambda)x_{k2}\| &= \|\lambda(x_{k1} - x_1) + (1 - \lambda)(x_{k2} - x_2)\| \\ &\leq \lambda \|x_{k1} - x_1\| + (1 - \lambda) \|x_{k2} - x_2\| \end{aligned}$$

En passant aux limites

$$\lim \|\lambda x_{k1} + (1 - \lambda)x_{k2} - [\lambda x_1 + (1 - \lambda)x_2]\| = 0.$$

ce qui montre que  $\lambda x_1 + (1 - \lambda)x_2 \in \bar{S}$ .

□

### 1.3.2 Fonction convexe

**Définition 1.24.** On dit qu'une fonction  $F : S \subset \mathbb{R}^n \rightarrow \mathbb{R}$ , définie sur un ensemble convexe  $S$ , est convexe si elle vérifie :

$$\forall x \in S, \forall y \in S, \forall \lambda \in [0, 1], \text{ on a } F(\lambda x + (1 - \lambda)y) \leq \lambda F(x) + (1 - \lambda)F(y).$$

$F$  est dite strictement convexe si l'inégalité stricte est toujours vérifiée pour  $x \neq y$  et  $\lambda \in ]0, 1[$ .

Une fonction  $F$  est concave si  $-F$  est convexe.

L'interprétation géométrique de cette définition est que le graphe de la fonction est toujours en dessous du segment reliant les points  $(x, F(x))$  et  $(y, F(y))$ .

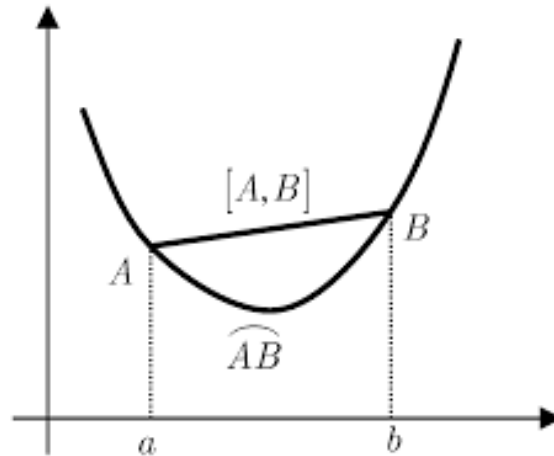


FIGURE 1.2 – Illustration d'une fonction convexe

**Propriétés 1.25.** Soit  $F$  une fonction réelle définie sur un ensemble convexe  $S \subset \mathbb{R}^n$ . Alors,  $F$  est convexe si et seulement si son épigraphe :

$$\text{epi}(F) = \{(x, t) \in \mathbb{R}^n \times \mathbb{R} : x \in S, F(x) \leq t\}$$

est un ensemble convexe.

*Démonstration.* [10]

Si  $F$  est convexe alors :

pour deux couple  $(x_1, t_1), (x_2, t_2) \in \text{epi}(F)$  et  $\lambda \in [0, 1]$ , on a :

$$F(x_i) \leq t_i, i = 1 : 2,$$

$$\lambda(x_1, t_1) + (1 - \lambda)(x_2, t_2) = (\lambda x_1 + (1 - \lambda)x_2, \lambda t_1 + (1 - \lambda)t_2),$$

comme

$$F(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda F(x_1) + (1 - \lambda)F(x_2),$$

alors

$$F(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda t_1 + (1 - \lambda)t_2.$$

Si  $\text{epi}F$  est convexe alors :

pour tout  $x_1, x_2 \in \text{dom}F$  et  $\lambda \in [0, 1]$ , on a

$$F(x_1) < \infty \quad \text{et} \quad F(x_2) < \infty, \quad (x, F(x)) \in \text{epi}F, \quad i = 1 : 2,$$

comme  $\text{epi}(F)$  est convexe on a :

$$\lambda(x_1, F(x_1)) + (1 - \lambda)(x_2, F(x_2)) = (\lambda x_1 + (1 - \lambda)x_2, \lambda F(x_1) + (1 - \lambda)F(x_2)) \in \text{epi}F.$$

D'où la convexité de  $F$ .

□

**Propriétés 1.26.** *Soit  $F$  une fonction réelle définie sur un ensemble convexe  $S \subset \mathbb{R}^n$ . Alors,  $F$  est convexe si et seulement si :*

$$F\left(\sum_{i=1}^n \lambda_i s_i\right) \leq \sum_{i=1}^n \lambda_i F(s_i);$$

Où  $s_i \in S$ ,  $i = 1, \dots, n$ ,  $\lambda_i \geq 0$  et  $\sum_{i=1}^n \lambda_i = 1$ .

**Théorème 1.27.**

1. *Soit  $F$  une fonction convexe dans un ensemble  $S \subset \mathbb{R}^n$  convexe et un nombre réel  $\alpha \geq 0$ . Alors,  $\alpha F$  est aussi une fonction convexe sur  $S$ .*
2. *Soient  $F_1, F_2$  deux fonctions convexes sur un ensemble  $S$  convexe. Alors,  $F_1 + F_2$  une fonction convexe sur  $S$ .*
3. *Soient  $F_1, F_2, \dots, F_m$  des fonctions convexes sur l'ensemble  $S$  convexe et  $\alpha_1, \alpha_2, \dots, \alpha_m \geq 0$  des nombre réels. Alors,  $\sum_{i=1}^m \alpha_i F_i$  est aussi une fonction convexe sur  $S$ .*

**Théorème 1.28.** [14] *Si  $F$  est continûment différentiable, les conditions (1) et (2) ci-dessous sont équivalentes. De plus, si  $F$  est deux fois continûment différentiable, les conditions (1), (2) et (3) ci-dessous sont équivalentes :*

1.  *$F$  est convexe ;*
2.  *$\forall x \in S, \forall y \in S, F(y) \geq F(x) + \nabla F(x)'(y - x)$  ;*
3.  *$\forall x \in S$ , le hessien  $\nabla^2 F(x)$  est une matrice semi-définie positive.*

Alors, d'après le théorème précédent, on déduit facilement qu'une fonction quadratique  $F(x) = x'Dx + c'x$  est convexe si et seulement si sa matrice associée  $D$  est semi-définie positive.

## Conclusion

Dans ce chapitre, nous avons présenté les notions et les propriétés des formes quadratiques. En plus, nous avons vu la notion de la convexité et les propriétés des fonctions convexes.

# Chapitre 2

## La programmation quadratique convexe (PQC)

### 2.1 Introduction

Le programme quadratique est le plus simple problème d'optimisation non linéaire avec contraintes. Dans ce chapitre nous verrons les définitions des problèmes quadratiques, les conditions nécessaires et les conditions suffisantes pour l'optimalité des problèmes quadratiques convexes.

### 2.2 Programmation non linéaire avec contraintes

#### 2.2.1 Formulation du problème

Un problème de programmation non linéaire avec contraintes se formule de la manière suivante :

$$\begin{cases} \min F(x) \\ h_i(x) = A_i'x - b_i = 0 & i \in \varepsilon \\ h_i(x) \leq 0 & i \in \mathcal{I} \end{cases} \quad (2.1)$$

- $F$  est la fonction objectif,
- $h_i, (i = 1, \dots, m)$  représentent les fonctions des contraintes, définies et dérivables sur un sous-ensemble de  $\mathbb{R}^n$ ,
- $x$  est un  $n$ -vecteurs,
- $b = (b_i, i \in \varepsilon \cup \mathcal{I})$  est un  $m$ -vecteurs,
- $A = (A_i, i \in \varepsilon \cup \mathcal{I})$  est une matrice d'ordre  $m \times n$ , notée la matrice des contraintes,
- l'ensemble  $\varepsilon = \{1, 2, \dots, m_e\}$  est l'ensemble des contraintes égalités et  $\mathcal{I} = \{m_e +$

$1, \dots, m\}$  celui des variables de décision.

- Les nombres  $m_e$  et  $m$  sont des entiers non négatifs, avec  $0 \leq m_e \leq m$ ,
- le signe ( $'$ ) représente celui de la transposition.

- Si  $m = 0$ , le problème (2.1) est un problème d'optimisation sans contraintes.
- Pour  $m_e = m \neq 0$ , il est dit problème d'optimisation avec contraintes d'égalités, si toutes les contraintes  $h_i(x)$  ( $i = 1, \dots, m$ ) sont linéaires, alors (2.1) est un problème d'optimisation avec contraintes linéaires.
- Si de plus la fonction  $F(x)$  est quadratique, alors (2.1) est un problème d'optimisation quadratique qui est l'objet de ce mémoire.

L'ensemble des solutions réalisables, noté  $X$ , est formé de tous les points satisfaisant les contraintes du problème (2.1), i.e.

$$X = \{x | h_i(x) = 0, i \in \varepsilon; h_i(x) \leq 0, i \in \mathcal{I}\} \quad (2.2)$$

### Définition 2.1.

- Un vecteur  $x \in \mathbb{R}^n$  est appelé solution réalisable ou plan du problème (2.1) s'il vérifie toutes les contraintes du problème, c'est-à-dire, que  $x \in X$ .
- Une solution réalisable  $x^*$  est appelée solution optimale du problème (2.1) si :

$$F(x^*) \leq F(x), \quad \forall x \in X;$$

et on note  $\min_{x \in X} F(x) = F(x^*)$ .

- $x^* \in X$  est appelé minimum local du problème (2.1) s'il existe un réel  $\varepsilon > 0$ , tel que :

$$F(x^*) \leq F(x), \quad \forall x \in X \cap B(x^*, \varepsilon).$$

où  $B(x^*, \varepsilon) = \{x \in \mathbb{R}^n, \|x - x^*\| < \varepsilon\}$  est la boule de centre  $x^*$  et de rayon  $\varepsilon$ .

- $x^*$  est un minimum local strict du problème (2.1), si

$$F(x^*) < F(x), \quad \forall x \in X \cap B(x^*, \varepsilon), x \neq x^*.$$

**Définition 2.2.** Il est clair qu'un minimum global est aussi un minimum local, mais sa recherche est plus difficile car il faut calculer tous les minimums locaux et puis prendre celui qui a la meilleure valeur de la fonction objectif comme minimum global.

**Définition 2.3.** Soit  $x \in X$ . On dit qu'un vecteur  $d \in \mathbb{R}^n$  est une direction admissible en  $x$  s'il existe un nombre réel  $\alpha > 0$ , tel que :

$$x + \theta d \in X, \quad \forall \theta \in [0, \alpha].$$

Si  $x$  est un point intérieur, alors toutes les directions sont admissibles.

**Théorème 2.4.** Soit la fonction  $F : \mathbb{R}^n \rightarrow \mathbb{R}$  de classe  $C^1$ . Si  $x^*$  est un point minimum local (ou global) du problème (2.1), alors pour toute direction admissible  $d \in \mathbb{R}^n$  en  $x^*$ ,

$$d' \nabla F(x^*) \geq 0$$

### 2.2.2 La méthode de Newton pour la minimisation d'une fonction différentiable

Lors de la minimisation d'une fonction  $F$  sur  $\mathbb{R}^n$ , il s'agit de chercher un point  $x^*$  tel que

$$\nabla F(x^*) = g(x^*) = 0, \quad g : \mathbb{R}^n \longrightarrow \mathbb{R}.$$

La méthode de Newton appliquée à l'équation  $g(x^*) = \nabla F(x^*) = 0$  construit alors les approximations successives de la manière suivante :

$$x^{(k+1)} = x^k - \left(\frac{\partial g}{\partial x}(x^k)\right)^{-1} g(x^k) \quad , \quad k = 0, 1, 2, \dots \quad (2.3)$$

Où

$$g = \begin{pmatrix} g_1 \\ g_2 \\ \cdot \\ \cdot \\ g_i \\ \cdot \\ \cdot \\ g_n \end{pmatrix}, \quad \frac{\partial g}{\partial x}(x^k) = \left(\frac{\partial g_i}{\partial x_j}(x_j), 1 \leq i, j \leq n\right)$$

En tenant compte le fait que  $\frac{\partial g}{\partial x}(x^k) = \nabla^2 F(x^k) = H(x^k)$ , alors on peut écrire :

$$x^{k+1} = x^k - H^{-1}(x^k) \nabla F(x^k), \quad k = 0, 1, 2, \dots$$

Le point  $x^{k+1}$  peut être interprété comme la racine de l'approximation linéaire de la fonction  $g(x) = \nabla F(x)$  au voisinage de  $x^k$ .

Une autre interprétation de la méthode de Newton est la suivante : elle consiste à minimiser l'approximation quadratique  $q(x)$  de  $F(x)$  au voisinage de  $x^k$  :

$$F(x) \simeq q(x) = F(x^k) + \nabla F'(x^k)(x - x^k) + \frac{1}{2}(x - x^k)' \nabla^2 F(x^k)(x - x^k).$$

Si le hessien  $H(x^k) = \nabla^2 F(x^k)$  est défini positif, le point  $x^{k+1}$  de la méthode de Newton est alors pris comme étant le minimum de la fonction  $q(x)$  :

$$\nabla q(x^{k+1}) = \nabla F(x^k) + \nabla^2 F(x^k)(x^{k+1} - x^k) = 0. \quad (2.4)$$

D'où la formule itérative :

$$x^{k+1} = x^k - [\nabla^2 F(x^k)]^{-1} \nabla F(x^k). \quad (2.5)$$

On remarque que dans la méthode de Newton, la direction ainsi que le pas sont fixés :

$$d^k = -[\nabla^2 F(x^k)]^{-1} \nabla F(x^k), \theta_k = 1.$$

La méthode de Newton possède une vitesse de convergence quadratique, mais la convergence ne peut être assurée que localement.

**Remarque.** Lorsque le hessien  $\nabla^2 F(x^k)$  n'est pas défini positif, la direction de déplacement  $d^k$  dans la méthode de Newton peut ne pas être une direction de descente. Dans ce cas, la méthode de Newton peut fournir aussi des minima locaux ou des points-selles.

**Remarque.** Dans le cas d'une fonction quadratique convexe, la méthode de Newton donne la solution  $x^*$  en une seule itération :

$$x^1 = x^0 - D^{-1}(Dx^0 + c) = x^0 - x^0 - D^{-1}c = -D^{-1}c = x^*$$

Cette unique itération, n'est que la solution du système linéaire  $Dx - c$ .

### 2.2.3 Conditions d'optimalité pour le cas des contraintes linéaires de type égalités

Le problème se formule sous la forme suivante :

$$\begin{cases} \min F(x) \\ A'_i x - b_i = 0, \forall i \in \varepsilon = 1, 2, \dots, m_e. \end{cases} \quad (2.6)$$

où  $F : \mathbb{R}^n \rightarrow \mathbb{R}$  est une fonction continûment différentiable,  $b$  est un vecteur de  $\mathbb{R}^m$  et  $A$  une matrice d'ordre  $m \times n$ .

**Remarque.** Pour que l'ensemble des solutions réalisables  $X$  ne soit pas vide ou ne soit pas réduit à un point isolé, on considérera que  $\text{rang} A = m < n$ .

**Définition 2.5.** La fonction  $\mathcal{L}(x, \lambda) = F(x) + \sum_{i=1}^m \lambda_i h_i(x)$  est appelée fonction de Lagrange associée au problème (2.6).

Le vecteur  $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_m) \in \mathbb{R}^m$  formé des multiplicateurs de Lagrange  $\lambda_i$ , est unique grâce à la remarque précédente.

**Théorème 2.6.** Soit  $x^*$  un minimum du problème (2.6). Alors, il existe nécessairement un vecteur  $\lambda \in \mathbb{R}^m$  vérifiant :

$$\nabla F(x^*) + A'\lambda = 0. \quad (2.7)$$

**Remarque.** La condition (2.7) peut être donnée autrement, en utilisant la fonction de Lagrange :

$$\nabla_x \mathcal{L}(x, \lambda) = \nabla F(x) + A'\lambda = 0 \quad (2.8)$$

Soit  $x^*$  un point de minimum relatif du problème (2.6). Il lui correspond donc un vecteur  $\lambda = (\lambda_1, \dots, \lambda_m)$  vérifiant la relation (2.8). Les nombres  $\lambda_i, 1 \leq i \leq m$ , sont uniques si et seulement si  $\text{rang}A = m$  comme le montre le théorème suivant :

**Théorème 2.7.** *Le vecteur des multiplicateurs de Lagrange  $\lambda$  est unique si et seulement si  $\text{rang}A = m$*

De plus, un minimum local est tout d'abord un point réalisable, qui vérifie :

$$Ax = b \Rightarrow \nabla_{\lambda} \mathcal{L}(x, \lambda) = Ax - b = 0 \quad (2.9)$$

En combinant les relations (2.8) et (2.9), on obtient alors la condition nécessaire d'optimalité de premier ordre pour le problème (2.6).

**Théorème 2.8 (théorème de Lagrange).** *Soit  $x^*$  un minimum local (ou global) pour le problème (2.6). Alors, il existe un vecteur multiplicateur de Lagrange  $\lambda^* \in \mathbb{R}^m$ , tel que :*

$$\nabla \mathcal{L}(x^*, \lambda^*) = 0 \Leftrightarrow \begin{cases} \nabla_{\lambda^*} \mathcal{L}(x^*, \lambda^*) = 0 \\ \nabla_{x^*} \mathcal{L}(x^*, \lambda^*) = 0 \end{cases} \quad (2.10)$$

Le couple  $(x^*, \lambda^*)$  est appelé point stationnaire de la fonction de Lagrange.

La condition nécessaire du second ordre pour le problème (2.6) est la suivante :

**Théorème 2.9.** *Soit  $x^*$  un minimum local pour le problème (2.6) et  $\lambda^*$  un vecteur multiplicateur de Lagrange vérifiant (2.10). Alors, la matrice  $\nabla^2 F(x^*)$  est semi-définie positive sur l'ensemble des points de la variété linéaire  $Ay = 0$ . Autrement dit :*

$$y' \nabla^2 F(x^*) y \geq 0, \forall y \in \mathcal{M}_0 = \{y \in \mathbb{R}^n : Ay = 0\}. \quad (2.11)$$

La condition suffisante de second ordre est la suivante :

**Théorème 2.10.** *Soit  $(x^*, \lambda^*)$  un couple de vecteurs vérifiant la condition nécessaire d'optimalité de premier ordre du problème (2.6) :*

$$\nabla \mathcal{L}(x^*, \lambda^*) = 0.$$

*Pour que  $x^*$  soit un minimum local du problème (2.6), il est suffisant que la matrice  $\nabla^2 F(x^*)$  soit définie positive sur le sous-espace vectoriel  $\mathcal{M}_0$ .*

## 2.2.4 Conditions d'optimalité pour le cas des contraintes linéaires de type inégalités

Considérons maintenant un problème non linéaire avec contraintes linéaires de type inégalités :

$$\begin{cases} \min F(x) \\ A'_i x - b_i \leq 0, \forall i \in I = \{1, 2, \dots, m\}. \end{cases} \quad (2.12)$$



**Définition 2.11.** Soit  $x$  une solution réalisable du problème (2.12). L'ensemble des contraintes actives (saturées) au point  $x$  est l'ensemble d'indices suivant :

$$I_0 = I_0(x) = \{i \in I : A'_i x = b_i\}.$$

**Lemme 2.12.** Pour les contraintes d'inégalités du problème (2.12), un vecteur  $d \in \mathbb{R}^n$  est une direction admissible au point  $x$  si et seulement si

$$A'd \leq 0, \forall i \in I_0(x)$$

**Lemme 2.13** (Farkas). Soit  $(m+1)$  vecteurs de  $\mathbb{R}^n$  :  $c, A_i, i = 1, 2, \dots, m$ , avec  $m < n$ . Si pour chaque vecteur  $x \in \mathbb{R}^n$  vérifiant

$$A'_i x \leq 0, i = 1, 2, \dots, m, \text{ on a } c'x \leq 0,$$

alors il existe des coefficients  $\lambda_i \geq 0$  et  $i = 1, \dots, m$ , tels que

$$c = \sum_{i=1}^m \lambda_i A_i.$$

**Théorème 2.14 (Théorème de Karush-Kuhn-Tucker 1951).** [3] Soit  $x^*$  un minimum local (ou global) du problème (2.12). Alors, il existe un  $m$ -vecteur  $\lambda^* > 0$  tel que :

(i) Pour la fonction de Lagrange  $\mathcal{L}(x^*, \lambda^*) = F(x^*) + \sum_{i=1}^m \lambda_i^* h_i(x^*)$ , la condition de stationnarité est vérifiée :

$$\nabla F(x^*) + \sum_{i=1}^m \lambda_i^* A_i = 0. \quad (2.13)$$

(ii) La condition de complémentarité (écarts complémentaires) est remplie :

$$\lambda_i^* h_i(x^*) = 0 \quad \forall i \in I.$$

## 2.3 Programmation convexe

Le concept de convexité est très important en programmation mathématique. Cette classe de problèmes, appelée optimisation convexe, concerne les problèmes d'optimisation non linéaire (2.1), où la fonction objectif  $F(x)$  est convexe. En particulier, les conditions nécessaires d'optimalité deviennent également suffisantes, et tout le résultat acquiert un caractère global.

**Définition 2.15.** On dit qu'un problème de programmation mathématique est convexe (respectivement strictement convexe), s'il consiste à minimiser une fonction convexe (respectivement strictement convexe) sur un domaine convexe.

Autrement dit :

Le problème (2.1) est dit convexe si la fonction objectif  $F$  est convexe,  $h_i(x), i \in \varepsilon$ , sont des fonctions linéaires et  $h_i(x), i \in \mathcal{I}$ , sont des fonctions convexes.

**Théorème 2.16.** *Soit  $F : S \in \mathbb{R}^n \rightarrow \mathbb{R}$  strictement convexe sur  $S$  convexe. Le minimum de  $F$  sur  $S$ , s'il existe, est unique.*

*Démonstration.*

Soit  $x^* \in X$  tel que  $F(x^*) \leq F(x), \forall x \in X$ . Supposons qu'il existe  $y^* \neq x^*$  tel que  $F(y^*) \leq F(x), \forall x \in S$ . Formons pour  $\lambda \in ]0, 1[$  le vecteur :

$$u = \lambda y^* + (1 - \lambda)x^*.$$

D'après la stricte convexité de  $F$ , et puisque nécessairement  $F(y^*) = F(x^*)$  on a :

$$F(u) < \lambda F(y^*) + (1 - \lambda)F(x^*).$$

Ce qui contredit le fait que  $x^*$  soit un minimum. On a donc  $x^* = y^*$ . □

Pour tout problème de programmation convexe, nous avons les propriétés suivantes :

**Propriétés 2.17.**

- Soit  $F$  une fonction convexe définie sur un ensemble convexe  $S$ . Alors, l'ensemble des points où  $F$  atteint son minimum est convexe.
- Tout minimum local est minimum global.
- Si la fonction  $F$  est strictement convexe, alors son minimum global lorsque qu'il existe est atteint en un seul point  $x^*$ .

Considérons maintenant le problème avec contraintes donnée sous la forme (2.1), où  $F$  est une fonction convexe de classe  $C^1$

**Théorème 2.18.** *Soit  $(x^*, \lambda^*)$  un couple de vecteurs vérifiant les conditions de Karush-Kuhn-Tucker :*

(i)  $\nabla \mathcal{L}_x(x^*, \lambda^*) = 0, \lambda_i^* < 0, i \in I;$

(ii)  $\lambda_i^* h_i(x^*) = 0, i \in I.$

Alors le vecteur  $x^*$  constitue un minimum global de  $F$  sur  $X$ .

Ainsi, les conditions de KKT sont donc à la fois nécessaires et suffisantes de minimalité (c'est le théorème de KKT-convexe).

### 2.3.1 Problème quadratique convexe (PQC)

Il s'agit d'un problème d'optimisation où la fonction objectif est quadratique, s'écrivant sous la forme  $F(x) = \frac{1}{2}x'Dx + c'x$ , avec  $D$  symétrique, que l'on minimise sur un polyèdre convexe fermé.

#### Propriétés 2.19.

- Si la matrice hessienne  $D$  est semi-définie positive, alors le problème (2.1) est un problème de programmation quadratique convexe et la solution optimale locale  $x^*$  est une solution globale.
- Si  $D$  est définie positive, alors (2.1) est un problème quadratique (PQ) convexe strict et  $x^*$  est une solution globale unique.
- Si  $D$  est indéfinie, alors (2.1) est un problème quadratique non convexe.

**Théorème 2.20.** [3] *Tout problème quadratique convexe dont la valeur est finie admet (au moins) une solution.*

## 2.4 Dualité en programmation convexe

Le concept de la dualité revient souvent dans la littérature sur la programmation mathématique. Le but est de trouver une formulation alternative équivalente au problème original, qui convient le plus à la pratique ou qui a une signification théorique importante. Le problème original est dit problème primal et le problème transformé est le dual. Souvent, les variables dans le dual peuvent être interprétées comme des multiplicateurs de Lagrange pour le cas linéaire et prennent la valeur de  $\lambda^*$  comme solution duale, quand  $\lambda^*$  est le multiplicateur associé à la solution optimale  $x^*$ . Cependant, dans le cas non linéaire, il existe toujours une fonction objectif (souvent reliée à la fonction de Lagrange) qui doit être optimisée. Ici, on va traiter la dualité associée à un problème de programmation convexe comme problème primal.

Il est important de remarquer que si le problème primal n'est pas convexe, alors le problème dual peut très bien ne pas avoir de solution à partir de laquelle la solution primale peut être déduite. Donc, ce n'est pas possible d'appliquer la dualité comme technique générale dans le but de rechercher une solution. Nous allons par conséquent nous contenter d'introduire un résultat de base de la théorie de la dualité en programmation convexe.

#### Définition 2.21. "Problème Primal"

On définit un problème primal comme un problème de minimisation qui consiste à trouver un vecteur  $x^*$ , s'il existe, tel que

$$\begin{cases} F(x^*) = \min F(x), \\ x^* \in X = \{x \in \mathbb{R}^n / h_i(x) \leq 0, i = 1, 2, \dots, m\} \end{cases} \quad (2.14)$$

où la fonction  $F$  est convexe sur l'ensemble  $S$  défini par les fonctions  $h_i(x)$ ,  $i = 1, \dots, m$  qui sont aussi convexes sur  $\mathbb{R}^n$ .

**Définition 2.22. "Problème Dual"**

On définit le problème dual du (2.14) comme un problème de maximisation, qui consiste à trouver deux vecteurs  $k^* \in \mathbb{R}^n$  et  $y^* \in \mathbb{R}^m$ , s'ils existent, tel que

$$\begin{cases} \mathcal{L}(k^*, y^*) = \max \mathcal{L}(k, y), \\ (k^*, y^*) \in V = \{(k, y) \in \mathbb{R}^n \times \mathbb{R}^m / \nabla_k \mathcal{L}(k, y) = 0, y \geq 0\} \end{cases} \quad (2.15)$$

où  $\mathcal{L}(k, y) = F(k) + y'h(k)$  est la fonction de Lagrange associée au problème (2.14).

Les relations existantes entre le programme primal et de son dual sont données par les deux théorèmes suivantes :

**Théorème 2.23 (Théorème faible de la dualité, Wolfe 1961).**

Soient  $F(x)$  et  $\mathcal{L}(k, y)$  les fonctions objectifs du problème primal et son dual respectivement. On a alors

$$\mathcal{L}(k, y) \leq F(x), \quad \forall (k, y) \in V, \forall x \in X.$$

**Théorème 2.24 (Théorème fort de la dualité, Wolfe 1961).**

Soit  $x^*$  une solution optimale du problème primal (2.14). Alors il existe  $y^* \in \mathbb{R}^m, y^* \geq 0$ , tel que le couple  $(x^*, y^*)$  est une solution optimale du dual (2.15) et on a :

$$F(x^*) = \mathcal{L}(x^*, y^*).$$

## Conclusion

Dans ce chapitre nous avons présenté les conditions d'optimalité d'un problème non linéaire, la méthode de Newton et enfin nous avons vu la notion de la dualité.

# Chapitre 3

## Méthodes de résolution en programmation quadratique convexe

### 3.1 Introduction

Dans ce chapitre, nous introduisons deux approches développées pour la résolution des problèmes de programmation quadratique convexe. Elles sont généralement des extensions de celles développées pour la programmation linéaire (PL) :

- La première approche dite de points intérieurs (MPI) est réputée pour une bonne complexité polynomiale et qui est efficace pour la résolution des PQs de grande taille.
- La deuxième approche dite d'activation des contraintes qui est une généralisation de la méthode du simplexe pour la (PL).

Dans ce qui suit, nous résumons l'état de l'art des méthodes concernées par notre étude.

### 3.2 La méthode de points intérieurs

En 1984, Karmarkar[11] a montré qu'une méthode de points intérieurs (MPI) peut résoudre des programmes linéaires (PLs) en un temps polynomial. Les deux décennies suivantes ont vu d'énormes efforts fournis par les spécialistes de l'optimisation pour étudier les propriétés théoriques et pratiques des différentes MPIs. Une des premières découvertes est que ces méthodes peuvent être vues comme une modification de la méthode de Newton [16], qui est capable de traiter des contraintes d'inégalités. Nesterov et Nemirovski [15] ont montré que cette approche est applicable à une large classe de problèmes que celles des programmes linéaires. Par exemple, les problèmes de programmation quadratique

peuvent être résolus polynomialement comme d'autres programmes convexes en utilisant des MPIs.

Les MPIs ont des caractéristiques communes qui se distinguent de celle de la méthode du simplexe [5]. Chaque itération est coûteuse à calculer, mais peut faire un progrès significatif vers la solution. Par contre, le simplexe demande généralement un plus grand nombre d'itérations non coûteuses. La méthode du simplexe chemine autour de la bordure du polytope réalisable, testant une séquence de sommets (points extrêmes) jusqu'à ce qu'il trouve un point optimal. Les MPIs approximent la solution soit de l'intérieur ou de l'extérieur du domaine réalisable et ne s'approchent de la frontière qu'en limite. Les MPIs se présentent sous forme de trois classes : primale, duale et primale-duale.

Dans ce mémoire, on va présenter la méthode primale-duale.

### 3.2.1 Position du problème

On considère, pour simplifier, le problème avec contraintes d'inégalités seulement, celles pour lesquelles les algorithmes de points intérieurs ont été conçus. On l'écrit sous la forme :

$$\begin{cases} \min F(x) = \frac{1}{2}x'Dx + c'x, \\ A_i'x \leq b_i, i \in I, \end{cases} \quad (3.1)$$

où  $x, c$  et  $A_i$  sont des  $n$ -vecteurs ;  $b$  est un  $m$ -vecteur ;  $D$  est une matrice symétrique d'ordre  $n$  et  $A = (A_i, i \in I)'$  est une  $m \times n$ -matrice.

### 3.2.2 Problème dual

Considérons le PQ primal (3.1), où la matrice  $D$  est supposée semi-définie positive. La fonction de Lagrange associée au problème est la suivante :

$$\mathcal{L}(x, \lambda) = F(x) + \lambda'(Ax - b)$$

où  $\lambda = (\lambda_i, i \in I)$  est le vecteur de multiplicateurs de Lagrange associé aux contraintes du problème.

On peut écrire le problème dual comme suit :

$$\begin{cases} \max \mathcal{L}(x, \lambda) = \frac{1}{2}x'Dx + c'x + \lambda'(Ax - b) \\ Dx + c + A'\lambda = 0, \\ \lambda_i \geq 0 \end{cases} \quad (*)$$

En prenant la transposition, puis en multipliant la relation (\*) par  $x$  à droite, on aura :

$$-x'Dx - c'x = \lambda'Ax$$

En utilisant cette expression dans la fonction objectif duale, le problème précédent devient alors :

$$\begin{cases} \max \mathcal{L}(x, \lambda) = -\frac{1}{2}x'Dx - b'\lambda \\ Dx + c + A'\lambda = 0, \\ \lambda_i \geq 0, \quad i \in I. \end{cases} \quad (3.2)$$

Si  $D$  est semi-définie positive, alors la fonction objectif  $F$  du problème primal est convexe sur  $\mathbb{R}^n$  et par conséquent, la fonction objectif  $\mathcal{L}$  du problème dual est concave sur  $\mathbb{R}^n \times \mathbb{R}^m$ . Dans ce cas les conditions d'optimalité de KKT suivantes sont à la fois nécessaires et suffisantes pour l'optimalité du couple  $(x, \lambda)$  :

$$\begin{cases} Dx + c + A'\lambda = 0, \\ Ax - b \leq 0, \\ \lambda_i(Ax - b)_i = 0, \quad \forall i \in I, \\ \lambda \geq 0. \end{cases} \quad (3.3)$$

Tout point  $x \in \mathbb{R}^n$  vérifiant la deuxième inégalité de problème (3.1) est dite solution réalisable primale. De même, tout couple  $(x, \lambda)$  vérifiant la première et la quatrième relation de (3.3) est dite solution réalisable duale.

### 3.2.3 Technique de la méthode

En introduisant le vecteur des variables d'écart  $s = b - Ax$ , on peut réécrire les conditions d'optimalité (3.3) du couple  $(x, \lambda) = (x^*, \lambda^*)$  comme suit :

$$\begin{cases} Dx + A'\lambda + c = 0, \\ Ax + s - b = 0, \\ s_i \lambda_i = 0, \quad \forall i \in I, \\ (\lambda, s)' \geq 0. \end{cases} \quad (3.4)$$

En écrivant sous forme matricielle les conditions de KKT, nous savons en effet que l'optimum global vérifie :

$$F(x, \lambda, s) = \begin{bmatrix} Dx + A'\lambda + c \\ Ax + s - b \\ SAe \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad (s, \lambda) \geq 0, \quad (3.5)$$

où  $S = \text{diag}(s_1, s_2, \dots, s_m)$  et  $A = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_m)$  sont des matrices carrées diagonales d'ordre  $m$ , ainsi que  $e = (1, 1, \dots, 1)'$  est un  $m$ -vecteur dont chaque composante est égale à 1.

Le principe de la méthode de points intérieurs primale-duale est d'essayer de résoudre directement ce système d'équations. Ce système est carré et possède  $2m + n$  variables et  $2m + n$  équations. Les deux premières séries d'équations sont linéaires, tandis que la

dernière est non linéaire.

On peut observer que si nous n'avions à faire qu'à des équations linéaires et que la contrainte  $(s, \lambda) \geq 0$  n'existait pas, la résolution se ferait alors très simplement par la méthode d'élimination de Gauss. La tâche est donc rendue compliquée par l'existence des contraintes de positivité et des équations non linéaires.

En regardant le système d'un point de vue fonctionnel, on s'aperçoit qu'il est équivalent à résoudre l'équation  $F(x, \lambda, s) = 0$ , où  $F$  est une fonction de  $\mathbb{R}^{2n+m}$  dans  $\mathbb{R}^{2n+m}$ . La résolution d'équation de ce type est un domaine bien connu des méthodes numériques. En effet, sans l'existence des contraintes de positivité, cette équation se résout très bien par la méthode de Newton.

Les méthodes de points intérieurs de type primal-dual génèrent des itérés  $(x^k, \lambda^k, s^k)$  qui vérifient les bornes strictes, c'est-à-dire que  $\lambda^k > 0$  et  $s^k > 0$ . Cette propriété est à l'origine du terme point intérieur.

Admettons qu'on a une estimation initiale  $(x^k, \lambda^k, s^k)$ , la méthode de Newton forme une approximation linéaire du système non linéaire et itère selon :

$$(x^{k+1}, \lambda^{k+1}, s^{k+1}) = (x^k, \lambda^k, s^k) + (\Delta x^k, \Delta \lambda^k, \Delta s^k)$$

où  $(\Delta x^k, \Delta \lambda^k, \Delta s^k)$  est la solution du système suivant :

$$J(x^k, \lambda^k, s^k) \begin{bmatrix} \Delta x^k \\ \Delta \lambda^k \\ \Delta s^k \end{bmatrix} = -F(x^k, \lambda^k, s^k). \quad (3.6)$$

$J(x^k, \lambda^k, s^k)$  est la matrice Jacobienne de la fonction  $F$  et  $(\Delta x^k, \Delta \lambda^k, \Delta s^k)$  la direction de recherche. On a

$$J(x^k, \lambda^k, s^k) = \begin{bmatrix} D & A' & 0 \\ A & 0 & I \\ 0 & A^k & S^k \end{bmatrix} \quad (3.7)$$

De plus, on a

$$F(x^k, \lambda^k, s^k) = \begin{bmatrix} 0 \\ 0 \\ S^k A^k e \end{bmatrix}$$

Donc, l'équation de Newton s'écrit :

$$\begin{bmatrix} D & A' & 0 \\ A & 0 & I \\ 0 & A^k & S^k \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta \lambda^k \\ \Delta s^k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -A^k S^k e \end{bmatrix}. \quad (3.8)$$



Un pas complet le long de cette direction n'est pas toujours permis, parce que la méthode de Newton n'est pas une méthode qui tient compte de contraintes; rien ne garantit que ce pas de Newton ne nous emmène pas en dehors de la zone admissible (à savoir la zone où  $s \geq 0$  et  $\lambda \geq 0$ ). C'est d'ailleurs à cause de cela qu'il a fallu faire une recherche linéaire le long de cette direction pour déterminer le paramètre  $\alpha^k$  indiquant la longueur du pas. Une fois la longueur du pas est déterminée, on calcule alors le nouveau itéré comme suit :

$$(x^{k+1}, \lambda^{k+1}, s^{k+1}) = (x^k, \lambda^k, s^k) + \alpha^k(\Delta x^k, \Delta \lambda^k, \Delta s^k)$$

Mais, malheureusement les modifications apportées à la méthode de Newton en introduisant le paramètre  $\alpha^k$  sont insuffisantes en pratique. Cela est dû au fait que les pas suivants ( $\alpha^{k+1}, \alpha^{k+2}, \dots$ ) sont très petits, car les itérés correspondants tendent vers le bord de la zone admissible, ce qui ralentit extrêmement la convergence de la méthode. La direction obtenue dans (3.6) est dite direction de Newton pure, et pour pallier à ce problème, on considère une modification de cette dernière dite direction de Newton centrée. Nous introduisons tout d'abord le concept du chemin central.

### 3.2.4 La technique de chemin central

Un chemin central  $\mathbb{C}$  est un arc de points réalisables stricts qui joue un rôle vital dans les algorithmes de type primal-dual, il est paramétré par un scalaire  $\tau > 0$ , et à chaque point  $(x_\tau, \lambda_\tau, s_\tau) \in \mathbb{C}$  on résout le système suivant :

$$F(x_\tau, \lambda_\tau, s_\tau) = \begin{bmatrix} 0 \\ 0 \\ \tau e \end{bmatrix}, \quad (\lambda_\tau, s_\tau) > 0. \quad (3.9)$$

Cette équation n'est autre que l'équation de Newton pour laquelle la condition de complémentarité est remplacée par  $\lambda_i s_i = \tau$  ( $i = 1, \dots, m$ ). Elle approxime le système (3.5) de plus en plus lorsque  $\tau$  tend vers zéro. Si  $\mathbb{C}$  converge lorsque  $\tau$  tend vers zéro, il doit converger vers la solution primale-duale du système (3.9). Donc le chemin central nous guide vers la solution optimale le long d'une route en gardant les produits  $\lambda_i s_i$  strictement positifs et il les tend vers zéro avec un même tau.

Choisissons un paramètre de centrage  $\sigma \in [0, 1]$  et introduisons la mesure de la dualité  $\mu$  définie par :

$$\mu = \frac{1}{n} \sum_{i=1}^n \lambda_i s_i = \frac{\lambda' s}{n}.$$

qui mesure la moyenne des produits  $\lambda_i s_i$ . En posant  $\tau = \sigma \mu$  et en appliquant la méthode de Newton au système (3.8), on obtient :

$$\begin{bmatrix} D & A' & 0 \\ A & 0 & I \\ 0 & A & S \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \lambda \\ \Delta s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ (\Lambda S + \sigma \mu)e \end{bmatrix} \quad (3.10)$$

Le déplacement  $(\Delta x, \Delta \lambda, \Delta s)$  est une direction de Newton vers le point  $(x_\tau, \lambda_\tau, s_\tau) \in \mathbb{C}$  pour lesquels les produits sont égale à  $\tau = \sigma \mu$ . Par contre, le déplacement (3.8) pointe directement du point pour lesquelles les conditions de KKT (3.4) sont satisfaites. Si  $\sigma = 1$ , les équations (3.8) définissent une direction centrée et si  $\sigma = 0$ , (3.8) définissent une direction standard de Newton.

Pour la majorité des problèmes, un point de départ strictement réalisable  $(x^0, \lambda^0, s^0)$  est difficile à calculer. Les méthodes de points intérieurs irréalisables remédient à ça en imposant uniquement que les composantes  $\lambda^0$  et  $s^0$  soient strictement positives. Par conséquent, on change légèrement le système (3.8) en définissant les résidus des deux équations de la manière suivante :

$$r_d = Dx + A'\lambda + c; \quad r_p = Ax - b + s.$$

Alors comme (3.8) pour le calcul de la direction d'amélioration, on résout le système modifié suivant :

$$\begin{bmatrix} D & A' & 0 \\ A & 0 & I \\ 0 & A & S \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \lambda \\ \Delta s \end{bmatrix} = \begin{bmatrix} r_d \\ r_p \\ (\Lambda S + \sigma \mu)e \end{bmatrix} \quad (3.11)$$

La prochaine itération est obtenue en posant :

$$(x^{k+1}, \lambda^{k+1}, s^{k+1}) = (x^k, \lambda^k, s^k) + \alpha^k (\Delta x^k, \Delta \lambda^k, \Delta s^k). \quad (3.12)$$

où  $\lambda$  est choisi de manière à retenir les composantes de  $\lambda_i$  et  $s_i$  strictement positives :

$$\lambda^{k+1} = \lambda^k + \alpha^k \Delta \lambda^k > 0, \quad i = 1, 2, \dots, n, \quad (3.13)$$

$$s^{k+1} = s^k + \alpha^k \Delta s^k > 0, \quad i = 1, 2, \dots, n. \quad (3.14)$$

Il n'est pas difficile de voir que la valeur maximale  $\alpha_{max}$  qui maintient ces conditions est donnée par la formule suivante :

$$\alpha_{max} = \min \left\{ \min_{i: \Delta \lambda_i < 0} -\frac{\lambda_i}{\Delta \lambda_i}, \min_{i: \Delta s_i < 0} -\frac{s_i}{\Delta s_i} \right\} \quad (3.15)$$

Nous pouvons nous éloigner de cette valeur maximale et empêcher ainsi chaque composante  $\lambda_i$  et si  $s_i$  d'être aussi proche de zéro en définissant le pas  $\alpha$ , comme suit :

$$\alpha = \min \{1, \eta \alpha_{max}\}, \quad (3.16)$$

où  $\eta$  est un coefficient proche de 1 et généralement on le prend égal à 0.999.

### 3.2.5 Schéma de l'algorithme

Dans cette partie, on présente l'algorithme de la méthode de point intérieur primal-dual pour la programmation quadratique convexe :

**Données :** Soit  $(x^0, \lambda^0, s^0)$  avec  $(s^0, \lambda^0) > 0$  une solution réalisable initiale et

$$\mu^0 = \frac{\lambda^{0'} s^0}{n};$$

Choisir  $\sigma \in [0, 1]$ ;

Pour  $k=1, 2, \dots$  (l'indice d'itération) ;

**Tan que**  $(\lambda, s)' \geq 0$  **faire**

✓ résoudre le système (3.8) avec  $(x^k, \lambda^k, s^k)$  et  $\tau = \sigma\mu$  pour obtenir  $(\Delta x^k, \Delta \lambda^k, \Delta s^k)$ ;

✓ calculer le pas  $\alpha^k$  tel que  $s^k + \alpha^k \Delta s^k > 0$  et  $\lambda^k + \alpha^k \Delta \lambda^k > 0$ ;

✓ mise à jour le point  $(x^{k+1}, \lambda^{k+1}, s^{k+1}) = (x^k, \lambda^k, s^k) + \alpha^k (\Delta x, \Delta \lambda, \Delta s)$ .

**k=k+1 ;**

## 3.3 La méthode d'activation des contraintes

La méthode d'activation des contraintes (active set method ASM) est une méthode classique, développée au début des années *soixante-dix* pour la résolution des problèmes de programmation linéaire et quadratique. Elle s'applique pour des problèmes d'optimisation avec des contraintes linéaires de types inégalités ou mixtes (égalités et inégalités). Le principe général de la méthode consiste à écarter temporairement un certain nombre de contraintes d'inégalités et de résoudre à chaque itération un problème avec uniquement des contraintes d'égalités, correspondantes aux contraintes actives. Par la suite l'ensemble des indices actifs est ajusté en ajoutant ou /et en supprimant une contrainte à la fois jusqu'à l'obtention de la solution optimale.

La méthode ASM pour le cas linéaire est facile à appliquer par rapport au cas quadratique puisque cela dépend du nombre de contraintes actives à l'optimum. D'après la théorie de la programmation linéaire, on sait à l'avance que la solution optimale correspond à un

sommet du polyèdre du domaine admissible, contrairement à un problème quadratique où la solution peut être un sommet, une face ou un point de l'intérieur du polyèdre.

Les méthodes d'activation des contraintes sont itératives et essaient d'identifier les contraintes actives à l'optimum. Elles se présentent sous forme de trois classes : primale, duale et primale-duale.

Dans ce mémoire, on va présenter la méthode primale. Nous considérons le programme quadratique avec des contraintes d'égalités et d'inégalités linéaires suivant :

$$\begin{cases} \min \frac{1}{2}x'Dx + c'x \\ h_i(x) = A'_i x - b_i = 0 & i \in \varepsilon \\ h_i(x) \leq 0 & i \in \mathcal{I} \end{cases} \quad (3.17)$$

### 3.3.1 Critère d'optimalité

En associant aux contraintes du problème (3.17) le  $m$ -vecteur multiplicateur  $\lambda$  tels que  $\lambda = (\lambda_i, i \in I)$ , la fonction de Lagrange s'écrit :

$$L(x, \lambda) = \frac{1}{2}x'Dx + c'x + \lambda(A'x - b). \quad (3.18)$$

En plus, on définit l'ensemble actif  $I_0(x^*)$  à la solution optimale  $x^*$  comme étant les indices des contraintes pour lesquelles l'égalité est vérifiée :

$$I_0(x^*) = \{i \in I; A'_i x^* = b_i\}.$$

La condition nécessaire d'optimalité du point  $x^*$  pour le problème (3.17) est qu'il existe un  $m$ -vecteur  $\lambda^*$ , tels que :

$$\begin{aligned} Dx^* + c + \lambda^*(A'_i x^* - b_i) &= 0, \\ A'_i x^* &= b_i, \quad \text{pour tout } i \in I_0(x^*), \\ A'_i x^* &\leq b_i, \quad \text{pour tout } i \in I \setminus I_0(x^*), \\ \lambda_i^* &\in \mathbb{R}, \quad i \in \varepsilon, \quad \lambda_i^* \geq 0, \quad \text{pour tout } i \in \mathcal{I}, \\ \lambda_i^*(A'_i x^* - b_i) &= 0, \quad \text{pour } i \in \mathcal{I}. \end{aligned}$$

Sachant que :

$$\begin{cases} \frac{\partial L}{\partial x^*} = 0, \\ \frac{\partial L}{\partial \lambda^*} = 0, \end{cases} \iff \begin{cases} Dx^* + c + A'\lambda^* = 0; \\ Ax^* = b. \end{cases}$$

En écrivant ce système de  $(n + m)$  équations à  $(n + m)$  variables sous forme compacte, on obtient :

$$\begin{pmatrix} D & A' \\ A & 0 \end{pmatrix} \begin{pmatrix} x^* \\ \lambda^* \end{pmatrix} = \begin{pmatrix} -c \\ b \end{pmatrix}. \quad (3.19)$$

L'expression (3.19) peut être réécrite sous une autre forme, utile pour le calcul de  $x^*$  en posant  $x^* = x + p$ , où  $x$  est une estimation de la solution et  $p$  est le déplacement désiré. Par introduction de cette notation et en réarrangeant les équations, on aura :

$$\begin{pmatrix} D & A' \\ A & 0 \end{pmatrix} \begin{pmatrix} p \\ \lambda^* \end{pmatrix} = - \begin{pmatrix} g \\ r \end{pmatrix}. \quad (3.20)$$

Où  $g = Dx + c$ ,  $r = Ax - b$ ,  $p = x^* - x$ .

La matrice de (3.20) est appelée matrice de Karush-Kuhn-Tucker (KKT), et les résultats suivants donnent sous quelles conditions elle est non singulière.

Notons par  $Z$  une matrice d'ordre  $n \times (n - m)$ , où ses colonnes forment une base du noyau de  $A$  et qui vérifient donc  $AZ = 0$  et  $Z$  est donnée de la manière suivante :

$$\begin{pmatrix} -A_B^{-1} A_N \\ I_N \end{pmatrix} \quad (3.21)$$

où  $A_B = A(I, J_B)$ ,  $A_N = A(I, J_N)$ , le sous-ensemble  $J_B \subset J = \{1, \dots, n\}$ , tel que  $|J_B| = m$ , est l'ensemble d'indices de la matrice de base  $A_B$ , vérifiant  $\det(A_B) \neq 0$  et  $J_N = J \setminus J_B$  et les vecteurs  $P$  et  $\lambda^*$  auront les expressions simples suivantes :

$$p = Zp_N, \quad \text{avec } p_N = p(J_N) = -(Z'DZ)^{-1}Z'g,$$

$$\lambda^* = - \begin{pmatrix} A_B^{-1} \\ 0 \end{pmatrix} \left[ \begin{pmatrix} g_B \\ g_N \end{pmatrix} + \begin{pmatrix} D(J_B, J) \\ D(J_N, J) \end{pmatrix} p \right] = - [A_B^{-1}]' (g_B + D(J_B, J)p).$$

**Lemme 3.1.** [16] Soit  $A$  une matrice de rang complet en lignes, et supposons que la matrice hessienne réduite  $Z'DZ$  est définie positive. Alors, la matrice de KKT :

$$K = \begin{pmatrix} D & A' \\ A & 0 \end{pmatrix} \quad (3.22)$$

est non singulière, et il existe une paire de vecteurs  $(x^*, \lambda^*)$  satisfaisant le système (3.19).

**Théorème 3.2.** [16] Supposons que les conditions du lemme précédent sont vérifiées. Alors, le vecteur  $x^*$  satisfaisant (3.19) est l'unique solution globale pour le problème (3.17).

### 3.3.2 Itération de la méthode

À une itération donnée de l'algorithme, soit  $x$  la solution courante de  $I_0$  l'ensemble d'indices actifs correspondant. une itération de l'algorithme consiste à exécuter les opérations suivantes :

1. On résout le sous-problème quadratique correspondant à l'ensemble de travail suivant :

$$\begin{cases} \min F(x) = \frac{1}{2}x'Dx + c'x, \\ A'_i x = b_i \quad i \in I_0 \end{cases} \quad (3.23)$$

La solution optimale pour (3.23) est de la forme  $x + p$ , où  $p$  est une certaine correction (direction de descente) de la solution courante  $x$ . Alors le  $n$ -vecteur  $p$  est calculé de telle sorte que la fonction objectif diminue. On obtient :

$$\begin{cases} \min F(x + p) = \frac{1}{2}(x + p)'D(x + p) + c'(x + p), \\ A'_i(x + p) = b_i \quad i \in I_0, \end{cases}$$

En simplifiant ce problème, alors le programme quadratique à résoudre est le suivant :

$$\begin{cases} \min F(x) = \frac{1}{2}p'Dp + g'p, \\ A'_i p = 0 \quad i \in I_0, \end{cases} \quad (3.24)$$

Où  $g = Dx + c$  est le gradient de  $F$  au point  $x$ .  
On envisage deux cas possibles :

- ✓ Si  $p = 0$ , aller à l'étape (2)
- ✓ sinon ( $p \neq 0$ ), posons :

$$\bar{x} = x + \theta p \quad \text{avec} \quad 0 < \theta \leq 1. \quad (3.25)$$

où  $\theta$  est la longueur du pas, doit être choisi de telle sorte que la nouvelle solution soit réalisable. Pour  $i \in I_0$  n'importe quelle valeur de  $\theta$  va maintenir la faisabilité de  $\bar{x}$ . Par contre, toute contrainte non saturée doit vérifier :

$$A'_i(x + \theta p) \leq b_i \quad , \quad \forall i \in I \setminus I_0.$$

Alors, la valeur optimale  $\theta^0$  est la suivante :

$$\theta^0 = \min\{1, \theta_{i0}\} \quad (3.26)$$

$$\text{Où} \quad \theta_{i0} = \min\{\theta_i, \forall i \in I \setminus I_0\} \quad , \quad \text{avec} \quad \theta_i = \begin{cases} \frac{b_i - A'_i x}{A'_i p}, & \text{si } A'_i > 0, \\ \infty, & \text{si } A'_i \leq 0 \quad , i \in I \setminus I_0. \end{cases}$$

Si  $\theta^0 = 1$ , alors dans ce cas l'ensemble de travail ne change pas, et on aura :

$$\bar{x} = x + p \quad , \quad \bar{I}_0 = I_0.$$

Sinon, la contrainte  $i_0$  est ajoutée à l'ensemble de travail. Par conséquent, le nouveau ensemble de travail est donc

$$\bar{I}_0 = I_0 \cup \{i_0\}.$$

2. Cette étape consiste à vérifier si la nouvelle solution obtenue  $\bar{x}$  est optimale ou non dans le problème (3.17). Ceci se fait en calculant les multiplicateurs de Lagrange associés au problème (3.23). Rappelons que pour  $i \notin I_0$ ,  $\lambda_i = 0$ . Les autres composantes du vecteur  $\lambda$  sont calculées lors de la résolution du problème (3.23).

- (a) Si  $\lambda_i \geq 0, \quad \forall i \in I_0$ , alors la condition d'optimalité de *KKT* est remplie pour le problème (3.17). Donc, la solution optimale est trouvée, et on arrête l'algorithme.
- (b) Sinon, il existe une composante de vecteur  $\lambda$  vérifiant  $\lambda_{i_1} < 0$ , avec  $i_1 \in I_0$ . Alors on peut améliorer la valeur de la fonction objectif en supprimant la contrainte  $i_1$  de l'ensemble de travail. Si plusieurs contraintes ne vérifient pas le critère d'optimalité, alors la contrainte à supprimer doit vérifier la relation suivante :

$$\lambda_{i_1} = \min\{\lambda_i, \quad \lambda_i < 0 \quad , \quad i \in \mathcal{I} \cap I_0\}; \quad (3.27)$$

Donc, le nouveau ensemble de travail est  $\bar{I}_0 = I_0 \setminus \{i_1\}$ . Ce processus de résolution est répété jusqu'à ce que l'optimum soit trouvé.

### 3.3.3 Schéma de l'algorithme

Soit  $x$  une solution réalisable du problème (3.17) et  $I_0$  l'ensemble du travail correspondant.

Le schéma de l'algorithme présente les étapes suivantes :

- (a) Calculer  $p$  en résolvant le programme quadratique (3.24) :
  - Si  $p = 0$ , alors aller à (c) :
  - Sinon, on pose

$$\bar{x} = x + \theta^0 p,$$

où  $\theta^0$  est défini par la relation (3.26). Aller à (b)

(b) Changement de l'ensemble actif :

– Si  $\theta^0 = \theta_{i_0} < 1$ , alors

$$\bar{I}_0 = I_0 \cup \{i_0\}$$

– Sinon ( $\theta^0 = 1$ ), on pose

$$\bar{I}_0 = I_0$$

Aller à (c).

(c) Calculer les multiplicateurs de Lagrange de (3.23).

– Si  $\lambda_i \geq 0, \forall i \in I_0$ , alors l'algorithme est arrêté, avec  $\bar{x}$  solution optimale.

– Sinon, déterminer

$$\lambda_{i_1} = \min\{\lambda_i : \lambda_i < 0, i \in \mathcal{I} \cap I_0\}$$

posons  $\bar{I}_0 = I_0 \setminus \{i_1\}$ . Aller à (a).

### 3.3.4 Exemple numérique

Considérons le programme quadratique suivant :

$$\begin{cases} \min F(x) = 2x_1^2 + x_1x_2 + x_2^2 - 12x_1 - 10x_2, \\ x_1 + x_2 \leq 4, \\ -x_1 \leq 0, \\ -x_2 \leq 0. \end{cases} \quad (3.28)$$

Les données de ce problème sont les suivantes :

$$D = \begin{pmatrix} 4 & 1 \\ 1 & 2 \end{pmatrix}, \quad c = \begin{pmatrix} -12 \\ -10 \end{pmatrix}, \quad A = \begin{pmatrix} 1 & 1 \\ -1 & 0 \\ 0 & -1 \end{pmatrix}, \quad b = \begin{pmatrix} 4 \\ 0 \\ 0 \end{pmatrix}.$$

La méthode qu'on utilise dans cet exemple pour la résolution du problème (3.24) est la méthode d'élimination simple des variables.

**Itération 1 :** Soit  $x^1 = (0, 0)'$  la solution réalisable de départ pour ce problème. L'ensemble d'indices actifs en ce point est  $I_0^1 = \{2, 3\}$ . Alors pour le problème (3.23), on a :

$$A = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Il est clair que  $p^1 = (0, 0)'$  est optimal pour (3.24), puisque en fait aucun autre point n'est réalisable pour cet ensemble actif. Le vecteur des multiplicateur de Lagrange associé est :

$$\lambda^1 = -A^{-1}g(x^1) = -\begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} -12 \\ -10 \end{pmatrix} = \begin{pmatrix} -12 \\ -10 \end{pmatrix}.$$



Aucun indice actif n'est optimal et par conséquent la deuxième contrainte sera supprimée de l'ensemble de travail. Donc, on aura :

$$x^2 = x^1, \quad I_0^2 = I_0^1 \setminus \{2\} = \{3\}.$$

**Itération 2 :** Résolution du problème (3.23) avec les paramètres suivants :

$$A = (0 \quad -1), \quad b = 0, \quad Z = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad Z'DZ = 4.$$

Le déplacement optimal est le suivant :

$$p^2 = -Z(Z'DZ)^{-1}Z'g = -\frac{1}{4} \begin{pmatrix} 1 \\ 0 \end{pmatrix} (1 \quad 0) \begin{pmatrix} -12 \\ -10 \end{pmatrix} = \begin{pmatrix} 3 \\ 0 \end{pmatrix}.$$

Le nouveau point obtenu est :

$$x^3 = x^2 + p^2 = \begin{pmatrix} 3 \\ 0 \end{pmatrix},$$

qui est réalisable. Le multiplicateur de Lagrange associé à la contrainte saturée est alors :

$$\lambda_3 = -[A_B^{-1}]' [g_B + D(J_B, J)p] = -(-1) \left[ -10 + (1 \quad 2) \begin{pmatrix} 3 \\ 0 \end{pmatrix} \right] = -7 < 0.$$

Par conséquent, la troisième contrainte est supprimée de l'ensemble des indices actifs ; donc, on aura :

$$I_0^3 = I_0^2 \setminus \{3\} = \emptyset$$

**Itération 3 :** A cette itération aucune contrainte n'est active au point  $x^3$ . Par conséquent, la solution du problème (3.28) sans contrainte nous donne :

$$x^* = -D^{-1}c = -\begin{pmatrix} \frac{2}{7} & -\frac{1}{7} \\ -\frac{1}{7} & \frac{4}{7} \end{pmatrix} \begin{pmatrix} -12 \\ -10 \end{pmatrix} = \begin{pmatrix} 2 \\ 4 \end{pmatrix},$$

qui n'est pas réalisable. Le déplacement du point  $x^3$  à  $x^*$  est le vecteur :

$$p^3 = x^* - x^3 = \begin{pmatrix} -1 \\ 4 \end{pmatrix}.$$

Calcule du  $\theta^0$  :

$$A_1'p^4 = (1 \quad 1) \begin{pmatrix} -1 \\ 4 \end{pmatrix} = 4 > 0 \implies \theta_1 = \frac{b_1 - A_1'x^3}{A_1'p^4} = \frac{4 - (1 \quad 1) \begin{pmatrix} 3 \\ 0 \end{pmatrix}}{3} = \frac{1}{3};$$

$$A_2'p^4 = (-1 \quad 0) \begin{pmatrix} -1 \\ 4 \end{pmatrix} = 1 > 0 \implies \theta_2 = \frac{b_2 - A_2'x^3}{A_2'p^4} = \frac{0 - (-1 \quad 0) \begin{pmatrix} 3 \\ 0 \end{pmatrix}}{1} = 3;$$

$$A'_3 p^4 = (0 \quad -1) \begin{pmatrix} -1 \\ 4 \end{pmatrix} = -4 \leq 0 \implies \theta_3 = \infty.$$

$$\theta_{i_0} = \min\{\theta_1, \theta_2, \theta_3\} = \theta_1 = \frac{1}{3} \implies i_0 = 1 \quad \text{alors} \quad \theta^0 = \min\{1, \theta_{i_0}\} = \theta_{i_0} = \frac{1}{3}.$$

La nouvelle solution obtenue, ainsi que l'ensemble de travail correspondant sont les suivants :

$$x^4 = \begin{pmatrix} 3 \\ 0 \end{pmatrix} + \frac{1}{3} \begin{pmatrix} -1 \\ 4 \end{pmatrix} = \begin{pmatrix} \frac{8}{3} \\ \frac{4}{3} \end{pmatrix}; \quad I_0^4 = I_0^3 \cup \{1\} = \{1\}.$$

**Itération 4** : Données correspondant à  $x^4$  et  $I_0^4$  :

$$A = (1 \quad 1), \quad b = 4, \quad Z = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \quad Z'DZ = 4, \quad g = g(x^4) = \begin{pmatrix} 0 \\ -\frac{14}{3} \end{pmatrix}.$$

Le déplacement optimal est le suivant :

$$p^4 = -Z(Z'DZ)^{-1}Z'g = -\frac{1}{4} \begin{pmatrix} 1 \\ -1 \end{pmatrix} (1 \quad -1) \begin{pmatrix} 0 \\ -\frac{14}{3} \end{pmatrix} = \begin{pmatrix} -\frac{7}{6} \\ \frac{7}{6} \end{pmatrix}$$

Calcul  $\theta^0$

$$A'_2 p^4 = (-1 \quad 0) \begin{pmatrix} -\frac{7}{6} \\ \frac{7}{6} \end{pmatrix} = \frac{7}{6} > 0 \implies \theta_2 = \frac{b_2 - A'_2 x^4}{A'_2 p^4} = \frac{0 - (-1 \quad 0) \begin{pmatrix} \frac{8}{3} \\ \frac{4}{3} \end{pmatrix}}{\frac{7}{6}} = \frac{16}{7}$$

$$A'_3 p^4 = (0 \quad -1) \begin{pmatrix} -\frac{7}{6} \\ \frac{7}{6} \end{pmatrix} = -\frac{7}{6} \leq 0 \implies \theta_3 = \infty$$

$\theta^0 = \min\{1, \theta_2, \theta_3\} = 1$ , alors on aura

$$x^5 = x^4 + p^4 = \begin{pmatrix} \frac{8}{3} \\ \frac{4}{3} \end{pmatrix} + \begin{pmatrix} -\frac{7}{6} \\ \frac{7}{6} \end{pmatrix} = \begin{pmatrix} \frac{3}{2} \\ \frac{15}{6} \end{pmatrix}.$$

La valeur du multiplicateur de KKT associé à la première contrainte est :

$$\lambda_3 = -[A_B^{-1}]'(g_B + D(J_B, J)P) = -\left[-\frac{14}{3} + (2 \quad 1) \begin{pmatrix} -\frac{7}{6} \\ \frac{7}{6} \end{pmatrix}\right] = \frac{7}{2} \geq 0.$$

Donc  $x^5$  est la solution optimale pour le problème (3.28).

## Conclusion

La méthode d'activation des contraintes a l'avantage de traiter les problèmes d'optimisation linéaire et quadratique tels qu'ils se présentent, sans chercher à modifier leurs contraintes. Cela permet d'avoir un gain en espace mémoire et en temps d'exécution sur la machine. L'inconvénient de cette méthode est que l'ensemble actif change lentement contrairement à d'autres méthodes qui permettent un changement rapide de l'ensemble actif.

# Chapitre 4

## Mise en œuvre

### 4.1 Introduction

L'évolution des outils informatiques a profondément influencé les méthodes de travail des ingénieurs et chercheurs sans oublier l'enseignement. Le traitement numérique des données, leur visualisation, ainsi que les techniques de modélisation et de simulation se sont notamment généralisés. Dans ce domaine, un logiciel commercial est devenu, ces dernières années, presque incontournable : il s'agit de Matlab de la société The Mathworks [2] et [4]

Ce dernier met à la disposition de l'utilisateur un environnement performant pour mener à bien les calculs numériques.

### 4.2 Choix du langage

Le choix s'est porté sur l'emploi du langage du logiciel Matlab 7.0, car il répond aux critères suivants :

- La maniabilité du langage : constitué d'un ensemble de possibilités faisant en sorte que le programmeur travaille avec aisance, assuré d'une part par la syntaxe du langage et d'autre part par un aspect visuel clair représentatif à la fois du détail et du global.
- Le bagage du langage : il contient une interface graphique puissante ainsi qu'une grande variété de méthodes scientifiques implémentées (prédéfinies).
- Facilité de manipulation des matrices (elles sont considérées comme une seule variable).

### 4.2.1 Généralités sur le langage

Matlab est un logiciel parfaitement dédié à la résolution de problèmes d'analyse numérique ou de traitement du signal. Permet d'effectuer des calculs matriciels ou de visualiser les résultats sous forme graphique. La formulation des problèmes s'apparente à la formulation mathématique des problèmes à résoudre. L'utilisation de ce logiciel consiste à lancer des lignes de commandes, qui peuvent le plus souvent ressembler à la programmation en C.

Le nom Matlab vient de MATrix LABoratory, les éléments de données de base manipulés par Matlab étant des matrices (mais pouvant évidemment se réduire à des vecteurs et des scalaires) qui ne nécessitent ni dimensionnement ni déclaration de type. Contrairement aux langages de programmation classiques, les fonctions du Matlab permettent de manipuler directement et interactivement ces données matricielles, le rendant ainsi particulièrement efficace en calcul numérique, analyse et visualisation de données en particulier. Il existe deux modes de fonctionnement sur Matlab :

- 1) **Le mode interactif** : les instructions sont exécutées au fur et mesure qu'elles sont données par l'utilisateur.
- 2) **Le mode exécutif** : dans ce cas, l'utilisateur utilise un fichier "M-file" contenant toutes les instructions à exécuter.

### 4.2.2 Programmation avec Matlab

Il existe deux façons pour l'écriture des fonctions Matlab : soit directement dans la fenêtre de commandes, soit en utilisant l'éditeur de développement de Matlab, en sauvegardant les programmes dans des fichiers texte avec l'extension ".m".

#### Fichiers \*.m

Les programmes sauvegardés dans les fichiers Matlab (\*.m) sont alors directement utilisables comme des fonctions Matlab à partir de la fenêtre de commande. Pour cela, le fichier doit se trouver dans le répertoire Matlab, qui est en pratique le dossier Work.

#### Création d'une fonction

La création d'une fonction dans Matlab se fait par la syntaxe suivante :

$$\textit{function} [s_1, s_2, \dots] = \textit{nom\_fonction}(e_1, e_2, \dots).$$

Les variables  $s_i$ , sont les paramètres de sortie de la fonction, et les variables  $e_j$  sont ses paramètres d'entrée.

**Remarque.**

*Le fichier \*.m(M-file) doit avoir le même nom que la fonction qu'il contient.*

### 4.2.3 Générateur d'exemples

Afin de bien tester et comparer l'efficacité des méthodes étudiées dans ce mémoire, nous avons développé un générateur de problèmes quadratiques convexes sous la forme standard. Ensuite, résoudre ces problèmes avec ces méthodes de résolution. Puis, comparer les résultats obtenus.

#### Générateur d'exemples tests

$[D, c, A, b] = \text{GénérateurExempleTest}(m, n);$

**Paramètres d'entrées :**

$n$  : nombre de variables.

$m$  : nombre de contraintes.

**Paramètres de sorties :**

$A$  : matrice d'ordre  $m \times n$ , notée la matrice des contraintes .

$D$  : matrice carrée symétrique d'ordre  $n$ , elle est semi-définie positif.

$c$  :  $n$ -vecteur.

$b$  :  $m$ -vecteur.

**Convention :**

`generer` : est une fonction qui génère des nombres aléatoires dans  $[0,1]$ , suivant une loi uniforme. {En Langage Matlab, `rand`}.

## 4.3 La méthode d'activation des contraintes et La méthode de points intérieurs en Matlab

Dans nos expérimentations, on a utilisé le solver "quadprog" (disponible sous Matlab pour les versions ultérieures à 6.5), en l'appelant ainsi :

$[x, fval, exitflag, output] = \text{quadprog}(D, c, A, b, options);$

Pour choisir l'algorithme à utiliser, on précise :

```
options = optimset('Algorithm','interior-point-convex');  
ou options = optimset('Algorithm','active-set');
```

Pour avoir en sortie le nombre d'itération on aura à composé le code suivant :

```
output.iterations
```

Pour récupérer l'algorithme utilisé on compose ce code :

```
output.algorithm
```

Pour plus d'informations sur ce solver, vous composez ce code :

```
help quadprog
```

### 4.3.1 Exemple d'exécution

Nous avons déjà trouvé la solution du problème (3.28), et maintenant nous allons résoudre ce problème sous matlab :

$D =$

```
4 1  
1 2
```

$A =$

```
1 1  
-1 0  
0 -1
```

$c =$

```
-12  
-10
```

$b =$

```

                4
                0
                0
F =

Inline function :

F(D,c,x) = 0.5 * x' * D * x + c * x

[x,fval,exitflag,output] = quadprog(D, c, A, b);

x1 =

        1.5000
        2.5000
fval1 =

       -28.5000
exitflag =

         1

        output =

iterations: 2

algorithm: 'medium-scale: active-set'

firstorderopt: []

cgiterations : []

message: 'Optimization terminated.'
```

On voit bien que la solution obtenue " $x_1$ " est la solution optimale du problème (3.28) :

**Remarque.**

- *fval* : La valeur de la fonction objectif à l'optimum.
- *exitflag* : Le problème converge si *exitflag*=1, (pour les autres valeurs consultez laide de Matlab)
- *output* : C'est un enregistrement ayant le nombre d'itération et l'algorithme utilisé.



## 4.4 Comparaison entre les deux méthodes

**Notation :**

- CPUtime : Le temps d'exécution (CPU) des problèmes tests en secondes. On a utilisé la fonction Matlab *tic-toc*.
- nbr Iter : Le nombre d'itérations requis pour la résolution des problèmes tests.
- $\Delta F$  : C'est la différence entre la valeur de la fonction objectif obtenue par les deux méthodes : la méthode de points intérieurs (MPI), la méthode d'activation des contraintes (ASM).

$n$	$m$	MPI		ASM		$\Delta F$
		CPUtime	nbr Iter	CPUtime	nbr Iter	
5	5	0.54	5	0.96	5	$6.402 \times 10^{-11}$
10	5	0.56	5	1.2	4	$2.401 \times 10^{-12}$
10	10	0.88	5	1.06	6	$5.696 \times 10^{-11}$
15	10	0.66	7	0.97	8	$2.401 \times 10^{-12}$
30	20	0.56	6	1.12	13	$5.684 \times 10^{-14}$
30	30	0.55	10	1.36	23	$3.676 \times 10^{-9}$
50	40	0.58	7	1.30	32	$2.842 \times 10^{-14}$
50	50	0.58	8	1.62	40	$5.684 \times 10^{-14}$
100	50	0.78	9	1.55	57	$4.843 \times 10^{-11}$
200	100	1.11	12	2.90	62	$9.952 \times 10^{-9}$
200	150	1.21	11	2.89	140	$1.377 \times 10^{-7}$

### 4.4.1 Discussion des résultats

Les résultats obtenus montrent clairement que les deux méthodes donnent presque la même valeur de la fonction objectif à l'optimum puisque  $\Delta F$  tend vers zéro.

Les résultats montrent également que la méthode de points intérieurs converge après quelques itérations sur tous les problèmes générés, alors que la méthode d'activation des contraintes prend plus d'itérations. De plus, on observe que MPI est plus performant que ASM en terme du temps d'exécution.

Ces résultats sont dus à l'inconvénient major de la méthode ASM : l'ensemble actif change lentement, c'est-à-dire, une seule contrainte change d'une itération à une autre. Alors que la méthode de points intérieurs fait un traitement significatif dans chaque itération ce qui réduit le nombre des itérations et accélère la convergence.

# Conclusion Générale

Dans ce travail, nous avons d'abord rappelé dans le premier et le deuxième chapitre les notions fondamentales d'algèbre linéaire et d'optimisation quadratique convexe avec contraintes.

Ensuite, dans le troisième chapitre, nous avons étudié deux méthodes de résolution des problèmes de minimisation quadratiques convexes, à savoir :

- La méthode d'activation des contraintes.
- La méthode de points intérieurs.

Dans le quatrième chapitre, nous avons vu les deux méthodes implémentées sous l'environnement MATLAB à travers le solver *quadprog*. Nous avons implémenté un générateur des problèmes de minimisation quadratiques convexes. Puis, appliquer les deux algorithmes étudiés et faire des comparaisons numériques entre les résultats en termes de temps d'exécution et de nombre d'itérations.

## Perspectives

Résoudre un problème de minimisation quadratique convexe par d'autres méthodes telles que : la méthode du gradient projeté, la méthode du simplexe, la méthode de support, et la méthode adaptée.

# Bibliographie

- [1] J. Abadie. On the Kuhn-Tucker theorem. In J. Abadie, editor, *Nonlinear Programming*. North Holland, Amsterdam, 1967.
- [2] Alfred A. Manuel. *Eléments de MATLAB*, Université de Genève, 15 octobre 2004
- [3] M. Bezoui. Méthode adaptée de programmation quadratique convexe. Mémoire de fin d'études, Université de Béjaia, 2008.
- [4] H. Carfantan. *Initiation à Matlab : Langage Matriciel*, 2000.
- [5] G. B. Dantzig. *Linear programming and extensions*. Princeton University Press, Princeton, N.J, 1963.
- [6] S. DJEMAI. Résolution des problèmes de classification SVM par la méthode adaptée. Thèse de doctorat, Université de Béjaia, 2016.
- [7] R. Fletcher and C.M.Reeves. *Functional minimization by conjugate gradients-comput*, 1964.p 149-154
- [8] R. Fletcher. *Practical Methods of Optimization*. J. Wiley and Sons, Chichester, England,second edition, 1987.
- [9] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization*. Academic Press Inc, London,1981.
- [10] T.Haddad. *Cours d'inégalités variationnelle*. Département de Mathématiques. Université de Jijel, 2017.
- [11] N. K. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4 :375-395, 1984
- [12] A. LAOUAR. Méthode adaptée pour la résolution d'un problème de programmation quadratique convexe à variables hybrides. Mémoire de Magister. Université de Béjaia, 2010.
- [13] Carl D. Meyer. *Matrix Analysis and Applied Linear Algebra*. -S.I.A.M, 2000.
- [14] M. Minoux. *Programmation Mathématique*. Bordas et C.N.E.T.-E.N.S.T., Paris, 1983.
- [15] Y. E. Nesterov and A. S. Nemirovsky. *Interior-Point Polynomial Methods in Convex Programming*. SIAM Publications, 1994.
- [16] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer-Verlag, New York, 1999.

- 
- [17] M.S.Radjeff. Cours de programmation Mathématique, 4<sup>eme</sup> année Recherche Opérationnelle. Université de Béjaia, 2007.
  - [18] S. Radjeff. Application de la méthode adaptée aux problèmes multicritères. Thèse de doctorat, Université de Béjaia, 2011.
  - [19] W. Sun and Ya-Xiang Yuan. Optimization theory and methods, Nonlinear Programming. Springer, 2006.
  - [20] S.J. Wright. Primal-Dual Interior-Point Methods. SIAM Publications, Philadelphia, 1997.