

République Algérienne Démocratique et Populaire

Ministère de l'enseignement Supérieur et de la

Recherche Scientifique



Université de Mohamed El Bachir El Ibrahimi de Borj

Bou Arréridj

Faculté des Mathématiques et d'Informatique

Département de Recherche Opérationnelle



Mémoire de Master

Mémoire présenté en vue de l'obtention du diplôme de Master en :

Domaine : Mathématique et Informatique.

Filière : Mathématique Appliquées.

Spécialité : Méthodes et outils pour la recherche opérationnelle.

THEME

Algorithme de points intérieurs pour la programmation
quadratique convexe.

Présenté par :

- MENDOUD Amel

- BOUAOUINA Fatiha

Sous la direction de :

Mme. GUERRA Loubna

Soutenu publiquement 14/Juin/2025 devant le jury composé de :

M.ZOUACHE Djaffar	Professeur	Université de BBA	Président
M.BELKACEM Nazihddine	Maître Assistant A	Université de BBA	Examineur
Mme. GUERRA Loubna	Maître de Conférence A	Université de BBA	Encadreur

2024/2025

Dédicace

*Avec toute ma gratitude et mon amour, je dédie ce modeste travail en
signe de respect et de reconnaissance à :*

*À moi-même, en reconnaissance de ma ténacité et de mes efforts
constants tout au long de mon parcours académique, et en célébration de la
réalisation de ce grand accomplissement.*

*Mes très chers parents, Hassina et Essaid, qui m'ont toujours encouragée
durant mes études. Je n'oublierai jamais leurs sacrifices pour moi. Un grand
merci à vous.*

*Mes sœurs, Ahlem et Chaherazzade, qui sont la lumière de mes yeux et le
battement de mon cœur.*

*Mon frère et mon soutien, Azzeddine, pour m'avoir toujours encouragée.
Mes amies, Manel, Samira, Laldja, Aya, Ines, Mounira et Ferial, qui
m'encouragent et me soutiennent.*

★ Fatiha★

Dédicace

Louange à Dieu, louange à Celui qui mérite toute gratitude pour Ses bienfaits, louange à Dieu qui m'a guidé et aidé jusqu'à ce que j'atteigne ce que je suis aujourd'hui.

Je dédie ce travail à mon cher père Saïd, et à ma chère mère Shahrzade, qui ont été la lumière illuminant mon chemin et le soutien qui ne m'a jamais abandonné. Mon père, tu as toujours été un exemple d'homme respectueux, honnête et compréhensif. Grâce à toi, j'ai appris la valeur du travail et de la responsabilité, et ton soutien a éclairé ma voie. Ma mère, tu as cru en moi quand j'avais des doutes, et tu as toujours été une source de force et de tendresse à chaque étape. Aucun mot ne peut exprimer pleinement ma reconnaissance envers vous deux. Ce succès est le fruit de votre amour, de vos sacrifices et de vos prières qui m'ont précédé à chaque réussite.

À ma sœur unique, Yasmine, et à mes frères Zineddine, Youssef, et Eljoudi, vous avez toujours été la lumière de ma vie et mon soutien à chaque instant. Vous êtes les piliers silencieux qui me donnent force, et la lumière qui dissipe l'obscurité des jours. Chaque réussite accomplie et chaque mot dans ce travail portent l'empreinte de votre soutien, de votre affection, et de votre présence

*précieuse dans ma vie. Vous êtes ma source d'inspiration pour continuer
toujours.*

*À mes chères amies, fidèles compagnes de route, dont l'amitié est un trésor
précieux dans ma vie, qui ont rendu ce chemin plus beau et plus léger par leurs
rires sincères et leur soutien constant. Vous avez partagé avec moi les moments
de joie et été un appui dans les moments difficiles. Merci pour votre présence,
votre écoute, et vos cœurs qui ont toujours été un refuge par leur gentillesse,
leur fidélité, et leur affection. Chaque succès accompli est aussi le fruit de votre
amour et de votre compagnie. J'espère que ce travail exprime sincèrement ma
reconnaissance.*

*Enfin, je remercie sincèrement tous ceux qui ont cru en moi, ceux qui m'ont
redonné espoir quand je l'avais presque perdu, et ceux qui sont restés à mes
côtés dans les moments de doute et de défi.*

★ Amel ★

Remerciement

Avant tout, nous remercions « **Allah** » Tout-Puissant pour la force et la patience qu'Il nous a accordées pour mener à bien ce projet. Sans Sa guidance, nous n'aurions pas pu atteindre ce résultat.

Nous exprimons également notre gratitude à notre encadrante, Dr. GUERRA Loubna maître de conférence classe A, à l'université Mohamed El-Bachir-El-Ibrahimi, pour ses conseils et son soutien tout au long de la réalisation de ce travail.

Nous adressons nos sincères remerciements aux membres du jury pour le temps qu'ils ont consacré à examiner et évaluer notre projet. Leurs remarques et recommandations seront sans aucun doute des éléments précieux pour le développement et l'amélioration de ce travail.

Nous n'oublions pas de remercier chaleureusement nos familles, qui ont été une source constante de soutien et d'encouragement durant toute la période de réalisation de ce projet. Elles ont été à nos côtés à chaque étape, nous offrant force et motivation pour continuer à travailler avec assiduité.

Enfin, nous remercions toutes les personnes qui ont contribué, de près ou de loin, à l'accomplissement de ce projet, que ce soit par leurs conseils, leur soutien moral ou toute autre forme d'aide. Nous leur sommes profondément reconnaissants pour leur rôle dans la réalisation de ce travail.

Table des matières

Notation et terminologie	viii
Notations principales	ix
Introduction Générale	1
1 Notions élémentaires d'un problème quadratique convexe	3
1.1 Matrices et vecteurs	3
1.1.1 Vecteurs linéairement indépendants	4
1.1.2 Normes et produit scalaire	4
1.1.3 Matrices	5
1.1.4 Le rang d'une matrice	6
1.1.5 Gradient et Hessienne	7
1.2 Convexité	8
1.2.1 Ensembles et fonctions convexes	8
1.2.2 La relation entre la convexité et l'affinité	10
1.2.3 Convexité et différentiabilité	11
1.3 Programmation quadratique convexe (PQC)	11
1.3.1 Cas particuliers	12
1.4 Domaines d'application d'un PQC	13
1.5 Modélisation de quelques problèmes réels sous forme d'un PQC	14
2 Méthode de points intérieurs pour la programmation quadratique convexe	18
2.1 Position du problème	18
2.1.1 Problème primal	19

2.1.2	La fonction de Lagrange	19
2.1.3	Problème dual	20
2.1.4	Dualité en programmation quadratique	21
2.2	Conditions d'optimalité	23
2.3	Méthode de trajectoire centrale avec point	23
2.3.1	La mesure de proximité	27
2.3.2	Algorithme de TC avec poids	28
2.3.3	convergence de l'algorithme	28
2.3.4	Analyse de la complexité	35
3	Réalisation numérique	37
3.1	Résultats numérique	38
3.1.1	Programmation linéaire	38
3.1.2	Programmation quadratique	44
	Conclusion Générale	51
	Références	51

Notation et terminologie

PQC : Programmation quadratique convexe,

PC : Programmation quadratique ,

FS : Forme standard,

FC : Forme canonique,

PL : Programmation linéaire,

MW : mégawatt,

DQC : Le dual d'un programme quadratique convexe,

P : Le problème convexe primal,

D : Le problème dual de (P),

TC : Trajectoire centrale,

K.K.T : Karush-Kuhn-Tucker,

c-à-d : C'est à dire,

CPI : Condition de point intérieur.

Notations principales

Operations

$x \geq 0$: les composantes de x , $x_i \geq 0$ pour tout i ;

$x > 0$: les composantes de x , $x_i > 0$ pour tout i ;

xz = $(x_1z_1, \dots, x_nz_n)^T$ (produit d'Hadamard);

$\frac{x}{z}$ = $\left(\frac{x_1}{z_1}, \dots, \frac{x_n}{z_n}\right)^T$, $z \neq 0$;

\sqrt{x} = $(\sqrt{x_1}, \dots, \sqrt{x_n})^T$, $x > 0$;

x^{-1} = $\left(\frac{1}{x_1}, \dots, \frac{1}{x_n}\right)^T$, $x \neq 0$;

$\langle x, z \rangle$ = $x^T z = e^T(xz) = \sum_{k=1}^n x_k z_k$;

$\|x\|_\infty$: $\max_{1 \leq i \leq n} |x_i|$ (norme infinie);

$\|x\|_2$: $\sqrt{\sum_{i=1}^n x_i^2}$ (norme euclidienne).

Ensembles

\mathbb{R}^n : L'espace euclidien des n -composantes réelles;

\mathbb{R}_+^n : L'orthant positif de l'espace \mathbb{R}^n ;

$\mathbb{R}^{m \times n}$: L'espace des matrices réelles à m lignes et n colonnes;

$\mathbb{R}^{n \times n}$: L'espace des matrices carrées d'ordre n ;

$\mathcal{F}_{(P)}$ = $\{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$,

l'ensemble des solutions réalisables de (P);

$\mathcal{F}_{(D)}$ = $\{(y, z) \in \mathbb{R}^{m \times n} : A^T y - \nabla f(x) + z = 0, x \geq 0, z \geq 0\}$,

l'ensemble des solutions réalisables de (D),

\mathbb{R}_{++}^n = $\{x \in \mathbb{R}^n \mid x > 0\}$;

\mathbb{R}_+^n = $\{x \in \mathbb{R}^n \mid x \geq 0\}$.

Vecteurs

- x^T : le vecteur transposé de x ;
 x_i : la i -ème composante de x ;
 x_k : le k -ème vecteur d'une suite de vecteurs ;
 e : le vecteur de \mathbb{R}^n dont toutes les composantes sont égales à 1.

Matrices

- A^T : la matrice transposée de A ;
 X : $\text{diag}(x)$, la matrice diagonale X avec $X_{ii} = x_i$;
 I : $\text{diag}(e)$, la matrice identité d'ordre n ;
 X^{-1} : $\text{diag}(x^{-1})$, l'inverse de X avec $X_{ii}^{-1} = \frac{1}{x_i}$, $x > 0$;
 A est de plein rang : A est de rang m ($m \leq n$) si ses lignes sont libres ;
 $A \in \mathbb{R}^{n \times n}$ est symétrique : si $A = A^T$, c.-à-d. $a_{ij} = a_{ji}$ pour tout i, j ;
 a_{ij} : Élément de la matrice $A \in \mathbb{R}^{m \times n}$;
 $0_{m \times n}$: La matrice zéro de taille ($m \times n$) ;
 I_m : La matrice identité de taille ($m \times m$).

Fonctions

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$, une fonction différentiable à plusieurs variables (x_1, \dots, x_n) . Alors :

$$\nabla f(x) = \left(\frac{\partial f}{\partial x_1}(x), \dots, \frac{\partial f}{\partial x_n}(x) \right)^T, \text{ gradient de } f \text{ au point } x;$$

$$\nabla^2 f(x) = \left(\frac{\partial^2 f}{\partial x_i \partial x_j}(x) \right)_{1 \leq i, j \leq n}, \text{ la matrice Hessienne};$$

$C^1(\mu)$: Espace vectoriel des fonctions continues et à dérivée continue sur μ ;

$C^q(\mu)$: Espace vectoriel des fonctions q fois continûment différentiables sur μ .

Introduction Générale

La programmation mathématique constitue une branche fondamentale des mathématiques appliquées. Elle vise à étudier les aspects théoriques des problèmes d'optimisation, en plus de concevoir et d'implémenter les algorithmes permettant de résoudre ces problèmes. L'objectif principal de l'optimisation mathématique est de trouver la meilleure solution possible à un problème donné parmi un ensemble d'alternatives admissibles, en maximisant ou minimisant une fonction objectif soumise à des contraintes spécifiques. Ce domaine revêt une grande importance, car il constitue un outil efficace permettant de modéliser des systèmes complexes et de guider la prise de décision de manière rigoureuse et méthodique, ce qui en fait un pilier central dans de nombreux domaines de recherche et d'application.

Les applications de l'optimisation mathématique sont extrêmement variées. On la retrouve en physique à travers les problèmes de minimisation d'énergie, dans l'industrie pour améliorer la qualité de la production et l'efficacité des processus, ainsi qu'en économie pour traiter des problématiques de localisation, de gestion de la production et de la distribution, de logistique et de transport, de gestion des stocks, de tarification et d'optimisation des flux dans les réseaux. Elle est également présente en finance pour améliorer les portefeuilles d'investissement, en traitement d'images, en biologie, en ingénierie, et même dans des problèmes pointus de géométrie mathématique comme celui de la plus petite sphère englobante.

Parmi les modèles les plus importants dans ce contexte figure la programmation quadratique convexe (PQC), qui repose sur des fonctions objectifs quadratiques et des contraintes linéaires. conférant à ces modèles une grande puissance théorique et une efficacité pratique dans de nombreuses applications.

Dans les années 1970, la théorie de la complexité est devenue un élément central de la programmation linéaire, en particulier après que Klee et Minty [1] ont démontré, à travers

un exemple célèbre, que la méthode du simplexe pouvait nécessiter un nombre exponentiel d'itérations. Cela a soulevé une question cruciale : existe-t-il un algorithme de complexité polynomiale pour résoudre les problèmes de programmation linéaire ?

La réponse est apparue en 1979 avec les travaux de Khachiyan [2], qui a démontré que la méthode des ellipsoïdes possède une complexité d'ordre $O(n^4L)$ en nombre d'itérations. Toutefois, bien que cette méthode offre des garanties théoriques intéressantes, elle s'est révélée peu performante en pratique, notamment par rapport à la méthode du simplexe qui, malgré sa complexité exponentielle, s'avère très efficace dans les applications réelles.

En 1984, Karmarkar a proposé un algorithme projectif de points intérieurs avec une complexité polynomiale d'ordre $O(n^{3,5}L)$, permet de concurrencer la méthode du simplexe dans la résolution de grands problèmes d'optimisation [3]. Durant les années 1990, les algorithmes de trajectoire central ont démontré une grande efficacité pratique. Kojima, Mizuno et Yoshizawa [4] ont proposé un algorithme de complexité $O(nL)$, tandis que, Mehrotra [5] a présenté un autre algorithme très performant, de type "prédicteur-correcteur", avec une complexité de $O(nL)$, qui s'est avéré très efficace en pratique.

Dans ce travail, on va étudier un algorithme de points intérieurs de type primal-dual (voir [6]) pour résoudre approximativement la programmation quadratique convexe.

Ce mémoire est composé de trois chapitres :

Premier chapitre : Il introduit certaines notions fondamentales et résultats clés qui seront utilisés par la suite, notamment les matrices et vecteurs, la convexité, la programmation quadratique convexe (PQC), ses domaines d'application ainsi que la modélisation de quelques problèmes réels sous forme de PQC.

Deuxième chapitre : Il traite des conditions d'optimalité et présente un algorithme Primal-Dual de trajectoire centrale avec poids, appliqué à la programmation quadratique convexe.

Troisième chapitre : Il expose les tests numériques effectués sur divers problèmes, notamment les problèmes linéaires et quadratiques . Une comparaison basée les différentes valeurs de θ est réalisée pour voir le comportement numérique de l'algorithme .

Enfin, le mémoire se conclut par une synthèse générale.

Chapitre 1

Notions élémentaires d'un problème quadratique convexe

Ce chapitre présente les fondements théoriques et les concepts clés de la programmation quadratique convexe (PQC). On commencera par un rappel sur les matrices, puis on abordera les notions de convexité et dualité en programmation quadratique. On examinera ensuite les principes de la PQC et les différentes classifications des programmes mathématiques. Enfin, On présentera les domaines d'application de la PQC, qui sont nombreux et variés.

1.1 Matrices et vecteurs

Définition 1.1 : Une matrice est un tableau rectangulaire constitué d'éléments, généralement des nombres ou des fonctions. Ces éléments sont le plus souvent des réels ou des complexes. Toutefois, dans ce qui suit, nous limiterons aux grandeurs réelles.

Une matrice A de dimension $m \times n$ est notée $A \in \mathbb{R}^{m \times n}$. Elle est composée de m lignes et n colonnes, et s'écrit sous la forme :

$$A = [a_{ij}] = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & a_{ij} & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}.$$

Une matrice V ayant une seule colonne et notée $V \in \mathbb{R}^{m \times 1}$, est appelée vecteur colonne et

s'écrit sous la forme :

$$V = \begin{bmatrix} v_1 \\ \vdots \\ v_m \end{bmatrix}.$$

De même, une matrice V constituée d'une seule ligne notée par $V \in \mathbb{R}^{1 \times n}$, est appelée vecteur ligne et s'écrit :

$$V = [v_1 \cdots v_n].$$

Par convention, tout vecteur est considéré comme une matrice colonne.

Enfin, si $m = n$, la matrice est dite carrée.

1.1.1 Vecteurs linéairement indépendants

Soient $\vec{v}_1, \vec{v}_2, \dots, \vec{v}_n$ des vecteurs dans un espace vectoriel. On dit qu'ils sont linéairement indépendants si la seule solution à l'équation suivante est la solution triviale [7] c-à-d :

$$\lambda_1 \vec{v}_1 + \lambda_2 \vec{v}_2 + \cdots + \lambda_n \vec{v}_n = \vec{0},$$

implique que

$$\lambda_1 = \lambda_2 = \cdots = \lambda_n = 0.$$

Autrement dit, aucun vecteur de l'ensemble ne peut s'écrire comme une combinaison linéaire des autres.

Si une autre solution (non triviale) existe, alors les vecteurs sont linéairement dépendants.

1.1.2 Normes et produit scalaire

Définition 1.2 : La norme vectorielle est une application de \mathbb{R}^n dans \mathbb{R}_+ , notée $\|\cdot\|$, et vérifie les conditions suivantes [8] :

1. $\forall x \in \mathbb{R}^n : \|x\| = 0 \Leftrightarrow x = 0$,
2. $\forall x, y \in \mathbb{R}^n : \|x + y\| \leq \|x\| + \|y\|$,
3. $\forall x \in \mathbb{R}^n, \forall \lambda \in \mathbb{R} : \|\lambda x\| = |\lambda| \|x\|$.

Définition 1.3 : Le produit scalaire usuel de deux vecteurs x et y de \mathbb{R}^n est défini par [8] :

$$\langle x, y \rangle = \sum_{i=1}^n x_i y_i = x^\top y.$$

Remarque 1.1 : Si le produit scalaire $\langle x, y \rangle = 0$, on dit que x, y sont orthogonaux.

1.1.3 Matrices

Définition 1.4 : (Matrices Symétriques)

Une matrice carrée A de taille n est dite **symétrique** si elle est égale à sa transposée [9], c-à-d :

$$A^T = A.$$

Ce qui signifie que ses éléments vérifient la propriété suivante :

$$a_{ji} = a_{ij}, \quad \forall i, j = 1, 2, \dots, n.$$

Théorème 1.1 : (Diagonalisation des Matrices Symétriques)

Une propriété fondamentale des matrices symétriques est qu'elles sont **toujours diagonalisables**. Plus précisément, toute matrice symétrique $A \in \mathbb{R}^{n \times n}$ peut être factorisée sous la forme [10] :

$$A = U \Lambda U^T,$$

où :

- U est une matrice **orthogonale** contenant les vecteurs propres de A , c-à-d : $UU^T = I_n$.
- Λ est une **matrice diagonale** contenant les valeurs propres de A .

Ce résultat est connu sous le nom de **théorème spectral**, qui garantit que toute matrice symétrique est diagonalisable dans une base orthonormée.

Définition 1.5 : (Matrices (semi-) définies positives)

Soit $A \in \mathbb{R}^{n \times n}$.

On dit que la matrice A est **semi-définie positive** si [11]

$$x^T A x \geq 0, \quad \forall x \in \mathbb{R}^n.$$

On dit que A est **définie positive** si,

$$x^T A x > 0, \quad \forall x \in \mathbb{R}^n, x \neq 0.$$

Proposition :

On a les propriétés suivantes :

1. $(A^t)^t = A$.
2. $(A^{-1})^t = (A^t)^{-1}$.
3. Si λ est un scalaire, alors $(\lambda A)^t = \lambda A^t$.
4. Si $A \in \mathbb{R}^{m \times p}$ et $B \in \mathbb{R}^{p \times n}$, alors $(AB)^t = B^t A^t$.
5. $(A + B)^t = A^t + B^t$ si A et B sont de la même taille.

Lemme 1.1.1 :

Soit $A \in \mathbb{R}^{m \times n}$ (avec $m \leq n$), alors la matrice $A^t A$ est symétrique et semi définie positive, donc les coefficients situés sur la diagonale principale sont positifs ou nuls.

1.1.4 Le rang d'une matrice

Le rang d'une matrice correspond au nombre maximum de colonnes ou de lignes linéairement indépendantes [7]. On note par $\text{rang}(A) = K$.

Une matrice $A, A \in \mathbb{R}^{m \times n}$, est dite de plein rang si :

$$\text{rang}(A) = \min(m, n).$$

Définition 1.6 : (Matrice de plein rang)

On dit qu'une matrice $A \in \mathbb{R}^{m \times n}$ est **de plein rang** (ou de rang maximal) si son rang est égal à $\min(m, n)$. Autrement dit, la dimension de l'espace engendré par ses lignes (ou par ses colonnes) est la plus grande possible.

- **Cas d'une matrice carrée :** Si $m = n$, être de plein rang signifie que $\text{rang}(A) = n$. Dans ce cas, la matrice est inversible et son déterminant est non nul.
- **Cas d'une matrice rectangulaire :**
 - Si $m > n$, on parle de **plein rang colonne** lorsque $\text{rang}(A) = n$.
 - Si $m < n$, on parle de **plein rang ligne** lorsque $\text{rang}(A) = m$.

Proposition :

Soit $A \in \mathbb{R}^{m \times n}$. Alors, on a :

$$\text{rang}(A) = \text{rang}(A^T) = \text{rang}(AA^T) = \text{rang}(A^T A).$$

1.1.5 Gradient et Hessienne

Définition 1.7 : Soit $F : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction de classe C^1 . Le gradient de la fonction F est défini par :

$$\nabla F(x) = \begin{pmatrix} \frac{\partial F}{\partial x_1}(x) \\ \frac{\partial F}{\partial x_2}(x) \\ \vdots \\ \frac{\partial F}{\partial x_n}(x) \end{pmatrix}.$$

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

Définition 1.8 : Soit $F : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction de classe C^2 . La hessienne de la fonction F est définie par :

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

Définition 1.9 : (La matrice Jacobienne)

Soit $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$, la fonction $J(x) : \mathbb{R}^n \rightarrow \mathbb{R}^{m \times n}$ est appelée matrice Jacobienne et définie par :

$$J(x) = \nabla F(x)^\top = \begin{pmatrix} \nabla f_1(x)^\top \\ \vdots \\ \nabla f_m(x)^\top \end{pmatrix} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{pmatrix}.$$

Remarque 1.2 : La matrice Hessienne est toujours symétrique.

1.2 Convexité

La convexité est un concept fondamental en optimisation mathématique, jouant un rôle central tant sur le plan théorique que numérique. En tant qu'outil indispensable, elle permet d'établir les conditions garantissant une solution optimale. Un problème convexe se distingue par des propriétés particulières qui en simplifient la résolution, ce qui confère à la convexité une importance essentielle. Ce concept se manifeste sous deux formes principales : les ensembles convexes et les fonctions convexes [12].

1.2.1 Ensembles et fonctions convexes

Définition 1.10 : Un ensemble D de \mathbb{R}^n est dit [13] :

- un **ensemble convexe** si

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

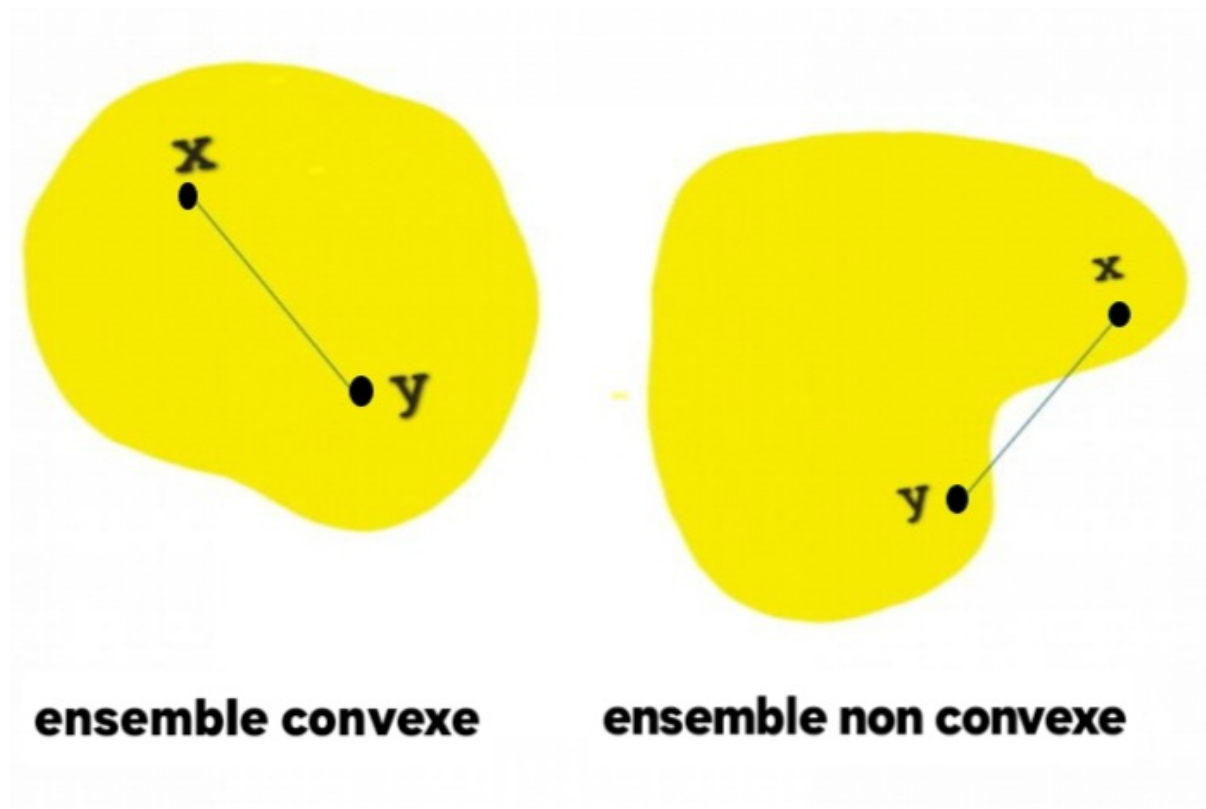


FIGURE 1.1 – Illustration des ensembles convexe et non convexe.

- un **polyèdre convexe** s'il est de la forme suivante :

$$D = \{x \in \mathbb{R}^n \mid Ax \leq b\},$$

où $A \in \mathbb{R}^{m \times n}$ et $b \in \mathbb{R}^m$.

- un **ensemble affine** si

$$\lambda x + (1 - \lambda)y \in D, \quad \forall x, y \in D, \quad \forall \lambda \in \mathbb{R}.$$

Définition 1.11 : Soit $f : D \rightarrow \mathbb{R}$ une fonction et D un ensemble convexe de \mathbb{R}^n , alors :

- f est dite **affine** sur D si

$$f(\lambda x + (1 - \lambda)y) = \lambda f(x) + (1 - \lambda)f(y), \quad \forall x, y \in D, \forall \lambda \in \mathbb{R}.$$

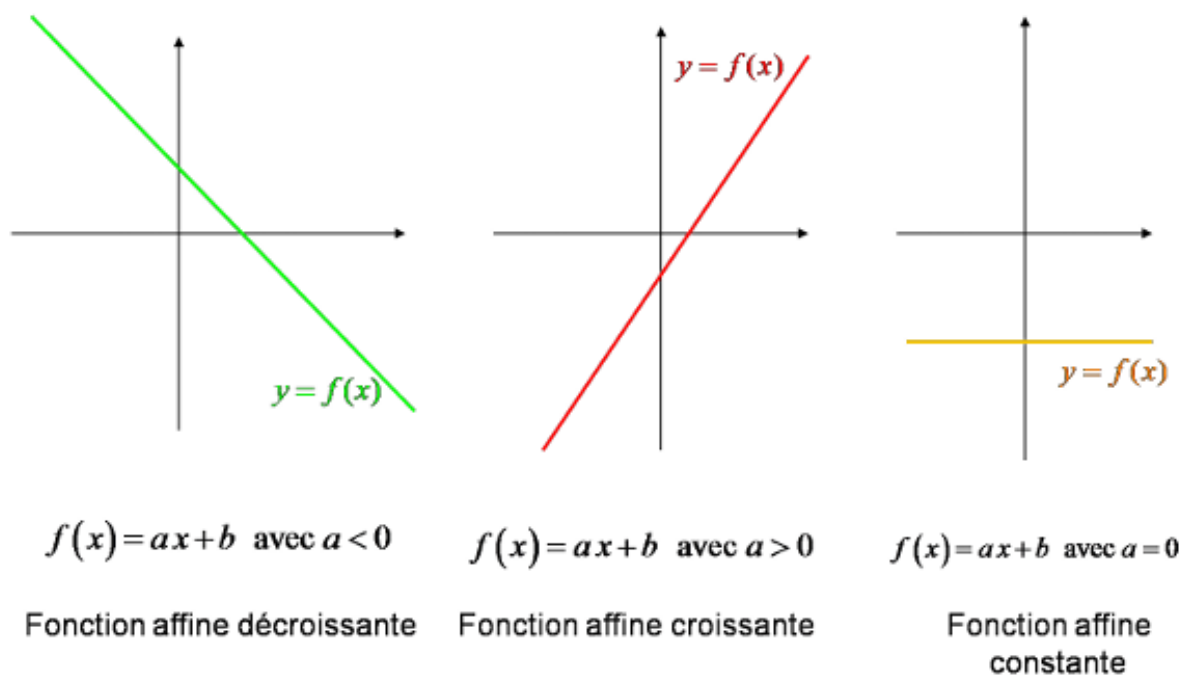


FIGURE 1.2 – Fonction affine.

- f est dite **convexe** sur D si

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y), \quad \forall x, y \in D, \forall \lambda \in [0, 1].$$

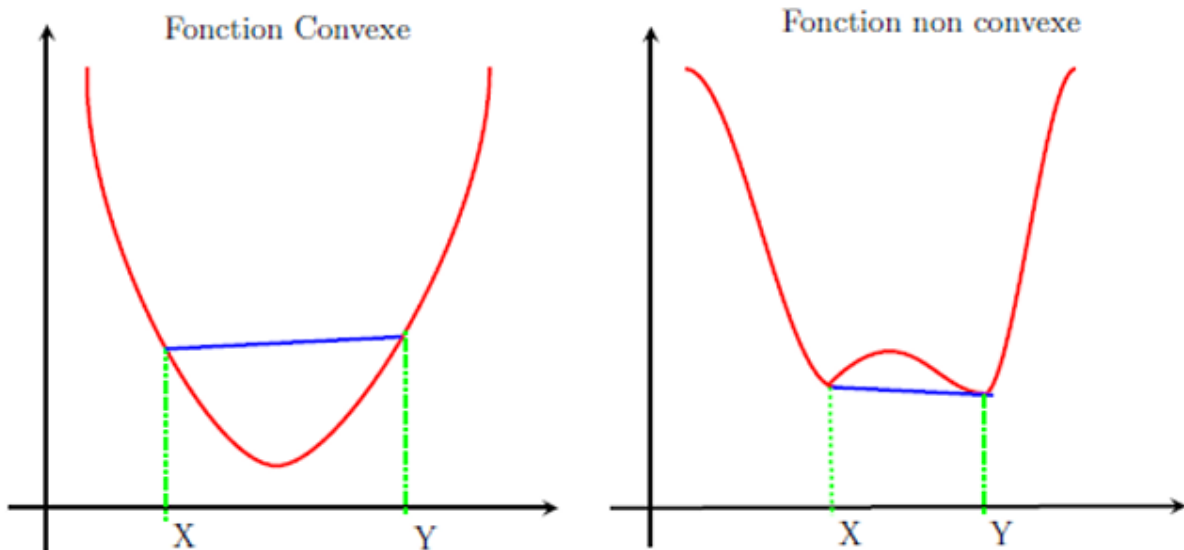


FIGURE 1.3 – Fonction convexe et non convexe.

- f est dite **strictement convexe** sur D si

$$f(\lambda x + (1 - \lambda)y) < \lambda f(x) + (1 - \lambda)f(y), \quad \forall x, y \in D \text{ et } x \neq y, \forall \lambda \in]0, 1[.$$

- f est dite **coercive** sur D si

$$\lim_{\|x\| \rightarrow +\infty} f(x) = +\infty.$$

Remarque 1.3 :

Tout fonction affine f définie sur $D \subset \mathbb{R}^n$ s'écrit sous la forme :

$$f(x) = c^T x + d;$$

avec $c \in \mathbb{R}^n$ et $d \in \mathbb{R}$

de plus, si $f(0) = 0$ alors f est linéaire et s'écrit sous la forme :

$$f(x) = c^t x$$

1.2.2 La relation entre la convexité et l'affinité

Tout ensemble affine est convexe, car toute combinaison convexe de deux points qu'il contient satisfait également la définition d'un ensemble affine. Cependant, la réciproque n'est pas vraie : tout ensemble convexe n'est pas forcément affine, car un ensemble convexe peut ne pas contenir toutes les extensions linéaires des points qu'il comprend [14, 13].

1.2.3 Convexité et différentiabilité

Lemme 1.1.2. Soit $f \in C^1(\Omega)$ où Ω est un ouvert et $D \subset \Omega$ un ensemble convexe, alors les propriétés suivantes sont équivalentes [15] :

- (i) f est une fonction convexe.
- (ii) $f(y) - f(x) \geq \langle \nabla f(x), y - x \rangle, \quad \forall x, y \in D.$
- (iii) $\langle \nabla f(y) - \nabla f(x), x - y \rangle \geq 0, \quad \forall x, y \in D.$

De plus, f est dite strictement convexe si les inégalités (ii) et (iii) sont rendues strictes pour $x \neq y$.

Définition 1.12 : Soit $f \in C^2(\Omega)$ où Ω est un ouvert et $D \subset \Omega$ un ensemble convexe [13], on dit que :

La fonction f est strictement convexe si et seulement si sa matrice hessienne $\nabla^2 f(x) = H$ est définie positive, c'est-à-dire que, pour tout $y \in D$ et tout $y \neq 0$, on a :

$$y^T H y > 0$$

Autrement dit, toutes les valeurs propres de $\nabla^2 f(x)$ sont strictement positives.

La fonction f est convexe si et seulement si sa matrice hessienne $\nabla^2 f(x) = H$ est semi-définie positive, c'est-à-dire que, pour tout $y \in D$, on a :

$$y^T H y \geq 0$$

Autrement dit, toutes les valeurs propres de $\nabla^2 f(x)$ sont positives ou nulles.

1.3 Programmation quadratique convexe (PQC)

La programmation quadratique constitue un domaine d'optimisation non linéaire, où la fonction objectif à minimiser est de nature quadratique, tandis que les contraintes définissant l'ensemble des solutions admissibles peuvent être linéaires ou quadratiques [13].

Elle se distingue par ses propriétés théoriques et son large éventail d'applications dans plusieurs disciplines scientifiques, notamment la finance, la médecine, l'économie et les télécommunications.

En effet, de nombreux problèmes issus du monde réel et du milieu académique peuvent

être formulés sous forme d'un programme quadratique. Parmi eux, on retrouve l'optimisation de portefeuilles financiers, la transmission de signaux multi-niveaux, les problématiques liées aux systèmes de capteurs à faisceaux multiples ainsi que certaines questions géométriques.

Un programme d'optimisation quadratique sous contraintes linéaires peut être formulé de la manière suivante :

$$\begin{cases} \min f(x) = \frac{1}{2}x^t Qx + c^t x, \\ Ax = b, \\ Dx \leq e, \\ x \in \mathbb{R}^n. \end{cases}$$

où $Q \in \mathbb{R}^{n \times n}$ est une matrice symétrique, $A \in \mathbb{R}^{m \times n}$, $D \in \mathbb{R}^{p \times n}$, $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$,

et $e = [1, 1, \dots, 1]^t \in \mathbb{R}^p$.

Tout programme quadratique peut se ramener à l'une des formes suivantes :

Forme standard (FS) :

$$\begin{cases} \min f(x) = \frac{1}{2}x^T Qx + c^T x, \\ Ax = b, \\ x \geq 0. \end{cases}$$

Forme canonique (FC) :

$$\begin{cases} \min f(x) = \frac{1}{2}x^T Qx + c^T x, \\ Ax \leq b, \\ x \geq 0. \end{cases}$$

1.3.1 Cas particuliers

- **Programation linéaire**

Étant donné un problème linéaire noté par (PL) comme suit [16] :

$$(PL) \quad \begin{cases} \min c^T x \\ Ax \leq b, \\ x \geq 0. \end{cases}$$

Où $A \in \mathbb{R}^{m \times n}$; $b \in \mathbb{R}^m$; $c \in \mathbb{R}^n$ sont les données et $x \in \mathbb{R}_+^n$ est l'inconnu de PL.

- **Problème des moindres carrés**

On se place dans le cas où la fonction F est linéaire [17], i.e. :

$$F(x) = Ax - b, \quad \text{avec } A \in M_{p,n}(\mathbb{R}) \text{ et } b \in \mathbb{R}^p,$$

et on résout donc le problème des moindres carrés linéaire suivant :

$$\min_{x \in \mathbb{R}^n} r(x) := \frac{1}{2} \|Ax - b\|_2^2$$

qui est équivalent à

$$\min_x \frac{1}{2} x^t (2A^t A) x + (-2A^t b)^t x + b^t b.$$

On pose :

$$Q = 2A^t A \in \mathbb{R}^{n \times n}, \quad C = -2A^t b \in \mathbb{R}^n$$

et

$$\alpha = b^t b \in \mathbb{R}.$$

Alors, le problème des moindres carrés s'écrit sous la forme :

$$\begin{cases} \min_{x \in \mathbb{R}^n} \frac{1}{2} x^t Q x + C^t x + \alpha \end{cases}$$

On note que la matrice Q est symétrique et semi-définie positive.

1.4 Domaines d'application d'un PQC

La programmation quadratique convexe représente le problème d'optimisation non linéaire avec contraintes le plus élémentaire. Elle est couramment employée dans la modélisation de

diverses applications, notamment :

- Le contrôle optimal [18].
- L'analyse structurelle [18].
- L'économie mathématique [19, 20].
- Les machines à vecteurs de supports (*SVM*) [21, 22, 23].
- La gestion des portefeuilles [24].

1.5 Modélisation de quelques problèmes réels sous forme d'un PQC

Problème 1 : Allocation optimale de la production d'énergie

Contexte :

Un opérateur énergétique doit décider combien produire à partir de deux centrales électriques pour minimiser le coût de production tout en respectant la demande en énergie.

- La centrale 1 a un coût de production quadratique :

$$C_1(x_1) = 2x_1 + 0.5x_1^2.$$

- La centrale 2 a un coût de production quadratique :

$$C_2(x_2) = 3x_2 + 0.2x_2^2.$$

- La demande totale en énergie doit être égale à 50 MW.
- Chaque centrale ne peut produire qu'une quantité positive d'énergie.

Nous devons résoudre :

$$\min_{x_1, x_2} 2x_1 + 0.5x_1^2 + 3x_2 + 0.2x_2^2.$$

Sous les contraintes :

$$x_1 + x_2 = 50,$$

$$x_1 \geq 0, \quad x_2 \geq 0.$$

Pour cela, on suit les étapes suivantes :

- **Étape 1** : Reformulation sous forme matricielle

Nous écrivons le problème sous la forme standard d'optimisation quadratique :

$$\min_x \quad c^T x + \frac{1}{2} x^T Q x,$$

avec :

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad c = \begin{bmatrix} 2 \\ 3 \end{bmatrix}, \quad Q = \begin{bmatrix} 1 & 0 \\ 0 & 0.4 \end{bmatrix}.$$

La contrainte d'égalité $x_1 + x_2 = 50$ est écrite sous forme matricielle :

$$A = \begin{bmatrix} 1 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 50 \end{bmatrix}.$$

Et les contraintes de positivité $x_1, x_2 \geq 0$ sont conservées.

- **Étape 2** : Résolution par la méthode des multiplicateurs de Lagrange

La fonction de Lagrange associée est :

$$\mathcal{L}(x_1, x_2, \lambda) = 2x_1 + 0.5x_1^2 + 3x_2 + 0.2x_2^2 + \lambda(x_1 + x_2 - 50),$$

où $\lambda \in \mathbb{R}$.

On prend les dérivées partielles et on résout :

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial x_1} = x_1 + 2 + \lambda = 0, \\ \frac{\partial \mathcal{L}}{\partial x_2} = 0.4x_2 + 3 + \lambda = 0, \\ \frac{\partial \mathcal{L}}{\partial \lambda} = x_1 + x_2 - 50 = 0. \end{cases}$$

- **Étape 3** : Résolution du système d'équations

On résout le système linéaire précédent, on obtient :

$$\begin{cases} \lambda = -x_1 - 2, \\ \lambda = -0.4x_2 - 3, \\ x_1 + x_2 = 50. \end{cases}$$

En égalisant les expressions de λ , on trouve :

$$-x_1 - 2 = -0.4x_2 - 3 \Rightarrow -x_1 + 0.4x_2 = -1.$$

Avec $x_1 + x_2 = 50$, on obtient :

$$x_1 = \frac{50 \cdot 0.4 - 1}{1.4} = 32,$$

et

$$x_2 = 50 - 32 = 18.$$

- **Étape 4** : Solution optimale

La production optimale est :

- Centrale 1 : $x_1^* = 32$ MW.
- Centrale 2 : $x_2^* = 18$ MW.

Le coût total minimal est :

$$\begin{aligned} C(32, 18) &= 2(32) + 0.5(32)^2 + 3(18) + 0.2(18)^2, \\ &= 64 + 512 + 54 + 64.8 = 694.8. \end{aligned}$$

L'opérateur doit produire 32 MW avec la centrale 1 et 18 MW avec la centrale 2 pour minimiser les coûts tout en respectant la demande.

Problème 2 : Problème d'optimisation de portefeuille

Contexte :

Un investisseur souhaite allouer son capital entre 4 actifs financiers afin de minimiser le risque (variance), tout en s'assurant que tout le capital est investi et sans ventes à découvert [25, 26, 27].

Afin de résoudre ce problème, on suit les étapes suivantes :

- **Étape 1** : Modélisation mathématique

Nous cherchons le portefeuille optimal x qui résout le problème suivant :

$$\min_x \left(\frac{1}{2} x^T Q x \right),$$

Sous les contraintes :

- Tout le capital est investi

$$\sum_{i=1}^n x_i = 1.$$

- Pas de ventes à découvert (les proportions doivent être positives)

$$x_i \geq 0, \quad \forall i = 1, \dots, n$$

où :

- $x = (x_1, x_2, x_3, \dots, x_n)^t$ représente les proportions investies dans chaque actif.
- Q est la matrice de covariance des rendements, qui reflète le risque ($Q \in \mathbb{R}^{n \times n}$ et inversible et définie positive).
- **Étape 2** : Solution analytique sans contrainte $x \geq 0$

D'abord, nous supposons que les ventes à découvert sont autorisées, puis nous utilisons la méthode des multiplicateurs de Lagrange pour trouver la solution analytique :

$$x^* = \frac{1}{e^T Q^{-1} e} \times Q^{-1} e,$$

où :

- Q^{-1} est l'inverse de la matrice de covariance.
- e est un vecteur rempli de 1, garantissant que la somme des proportions est égale à 1.

Chapitre 2

Méthode de points intérieurs pour la programmation quadratique convexe

Dans ce chapitre, nous nous intéressons à la méthode de points intérieurs de trajectoire centrale (TC) de type primal-dual, basée sur une nouvelle direction de recherche pour résoudre un programme non linéaire. Il est à noter que l'approche associée à cette étude théorique s'inspire des travaux de Wright et Nocedal [28] pour l'optimisation non linéaire.

2.1 Position du problème

En raison des nombreuses applications de la programmation quadratique, ce type de problème a suscité un grand intérêt auprès des chercheurs. Plusieurs de ses variantes ont été étudiées, notamment les cas convexes et non convexes, ainsi que les modèles intégrant des variables entières ou binaires. Parmi les travaux ayant abordé ces thématiques, on peut citer les études présentées dans [29, 30, 31]. La notion de convexité joue un rôle essentiel en programmation mathématique, car un problème convexe possède des propriétés remarquables qui facilitent grandement sa résolution. Rockafellar, dans [32], a souligné l'importance de distinguer clairement les problèmes convexes des problèmes non convexes. Dans le cadre de ce mémoire, on concentra sur l'étude des problèmes quadratiques convexes sous leur forme standard.

2.1.1 Problème primal

Un PQC sous forme standard primal s'écrit comme suit :

$$(P) \quad \begin{cases} \min & f(x) = \frac{1}{2}x^t Qx + c^t x, \\ Ax = b, \\ x \geq 0, \end{cases}$$

où $Q \in \mathbb{R}^{n \times n}$ est une matrice symétrique semi-définie positive, $A \in \mathbb{R}^{m \times n}$, $c, x \in \mathbb{R}^n$ et $b \in \mathbb{R}^m$.

On note par

$$F_{(P)} = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\},$$

$$F_{(P)}^+ = \{x \in \mathbb{R}^n : Ax = b, x > 0\},$$

les ensembles des solutions réalisables et strictement réalisables de (PQC) , respectivement.

La valeur optimale primale du (P) est définie par :

$$p^* = \inf_x \left\{ C^t x + \frac{1}{2} x^t Q x : Ax = b, x \geq 0 \right\},$$

x^* est dite solution optimale de (P) si :

$$x^* \in F_p \text{ et } p^* = C^t x^* + \frac{1}{2} (x^*)^t Q x^*.$$

2.1.2 La fonction de Lagrange

La fonction Lagrangien L associée an problème (P) est définie par

$$L : \mathbb{R}_+^n \times \mathbb{R}^m \mapsto \mathbb{R}^+$$

$$(x, y) \mapsto L(x, y) = c^t x + \frac{1}{2} x^t Q x + (b - Ax)^t y.$$

Le dual de (P) est donné par :

$$\max_{y \in \mathbb{R}^m} \left(\min_{x \in \mathbb{R}_+^n} L(x, y) \right) = \max_{y \in \mathbb{R}^m} H(y),$$

avec

$$\begin{aligned}
H(y) &= \min_{x \in \mathbb{R}_+^n} \left(c^t x + \frac{1}{2} x^t Q x + (b - Ax)^t y \right), y \in \mathbb{R}^m, \\
&= \min_{x \in \mathbb{R}_+^n} \left(c^t x + x^t Q x - \frac{1}{2} x^t Q x - x^t A^t y + b^t y \right), y \in \mathbb{R}^m, \\
&= \min_{x \in \mathbb{R}_+^n} \left(x^t (c + Qx - A^t y) \right) - \frac{1}{2} x^t Q x + b^t y, y \in \mathbb{R}^m. \\
H(y) &= \begin{cases} b^t y - \frac{1}{2} x^t Q x & \text{si } c + Qx - A^t y \in \mathbb{R}_+^n \\ -\infty & \text{sinon} \end{cases}
\end{aligned}$$

Donc le dual du problème (P) est donné :

$$\begin{cases} \max b^T y - \frac{1}{2} x^T Q x \\ c + Qx - A^T y \geq 0, \\ y \in \mathbb{R}^m, x \in \mathbb{R}_+^n \end{cases}$$

qui équivalent à :

$$(D) \begin{cases} \max_{x,y,z} b^T y - \frac{1}{2} x^T Q x \\ A^T y - Qx + z = c, \\ x \in \mathbb{R}_+^n, y \in \mathbb{R}^m, z \in \mathbb{R}_+^n. \end{cases}$$

2.1.3 Problème dual

Le problème dual (D) associé au problème primal (P) peut s'écrire sous la forme suivante :

$$(D) \begin{cases} \max -\frac{1}{2} x^t Q x + b^t y, \\ A^t y + z - Qx = c, \\ z \geq 0, x \geq 0, y \in \mathbb{R}^m. \end{cases}$$

La dualité joue un rôle fondamental dans l'étude et la résolution de (P). Entre autre, elle fournit des informations très utiles. On note par

$$F_{(D)} = \{(x, y, z) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n : A^t y + z - Qx = c, \quad x, z \geq 0\},$$

$$F_{(D)}^+ = \{(x, y, z) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n : A^t y + z - Qx = c, \quad x, z > 0\},$$

les ensembles des solutions réalisables et strictement réalisables de (D), respectivement.

La valeur optimale duale de (D) est donnée par :

$$d^* = \sup_{(x,y,z)} \left\{ b^t y - \frac{1}{2} x^t Q x : A^t y + z - Qx = c; (x, z) \geq 0, y \in \mathbb{R}^m \right\}$$

(x^*, y^*, z^*) est dite solution optimale de (D) si :

$$(x^*, y^*, z^*) \in F_D \text{ et } d^* = b^t y^* - \frac{1}{2} (x^*)^t Q x^*.$$

2.1.4 Dualité en programmation quadratique

- **Dualité faible :**

Définition 2.1 : (Saut de dualité)

Soient $x \in F$ et $(x, y, z) \in F_D$, alors la différence :

$$P^* - d^* = x^t z,$$

est appelée le saut de dualité des deux problèmes (D) et (P).

Théorème 2.1 : (Dualité faible) [33]

Soient $(x, y, z) \in F_P \times F_D$, alors

$$c^t x + \frac{1}{2} x^t Q x \geq b^t y - \frac{1}{2} x^t Q x,$$

c.à.d :

$$P^* \geq d^*.$$

Preuve : Soient $(x, y, z) \in F_P \times F_D$, alors on a :

$$\begin{aligned}
 P^* - D^* &= \left(\frac{1}{2}x^t Qx + c^t y\right) - \left(-\frac{1}{2}x^t Qx + b^t y\right) \\
 &= c^t x + \frac{1}{2}x^t Qx - b^t y + \frac{1}{2}x^t Qx \\
 &= (A^t y + z - Qx)^t x - (Ax)^t y + x^t Qx \\
 &= (y^t A + z^t - x^t Q^t)x - (x^t A^t)y + x^t Qx \\
 &= y^t Ax + z^t x - x^t Qx - x^t A^t y + x^t Qx \\
 &= y^t Ax - x^t A^t y + z^t x \\
 &= (Ax)^t y - x^t A^t y + z^t x \\
 &= x^t A^t y - x^t A^t y + z^t x \\
 &= z^t x \\
 &= x^t z \\
 &= \sum_{i=1}^n x_i z_i \geq 0 \quad \text{pour } x_i \geq 0, z_i \geq 0
 \end{aligned}$$

- **Dualité forte**

Théorème 2.2 : (Dualité forte) [33]

Soient $(x^*, y^*, z^*) \in F_P \times F_D$, tel que :

$$\frac{1}{2}(x^*)^t Q(x^*) + c^t(x^*) = -\frac{1}{2}(x^*)^t Qx^* + b^t y^*$$

c.à.d :

$$P^* = d^*.$$

où (x^*, y^*, z^*) sont des solutions optimales pour (D) et (P) respectivement.

Afin de résoudre les deux problèmes, on suppose qu'il vérifient les hypothèses suivantes :

Hypothèse 1 :

A est une matrice de plein rang , c-à-d :

$$\text{rang}(A) = m < n.$$

Hypothèse 2 :

Condition de point intérieur (CPI), $\exists(x^0, y^0, z^0) \in F_P^+ \times F_D^+$

c.-à-d :

$$\begin{cases} Ax^0 = b \\ A^T y^0 + z^0 - Qx^0 = c \\ x^0 > 0, z^0 > 0, y^0 \in \mathbb{R}^m \end{cases}$$

Hypothèse 3 :

La matrice Q est symétrique et semi-définie positive.

2.2 Conditions d'optimalité

La programmation quadratique convexe consiste à minimiser une fonction continue et convexe (car la matrice Q est symétrique et semi-définie positive).

Sous l'ensemble des contraintes qui est un polyèdre fermé, borné, convexe et d'intérieur non vide (d'après la deuxième hypothèse), donc la condition de Slater est satisfaite, c.-à-d. les contraintes sont qualifiées et les conditions de Karush–Kuhn–Tucker (K.K.T) sont des conditions nécessaires et suffisantes et s'écrivent comme suit :

$$\text{(KKT)} \quad \begin{cases} Ax = b \\ A^t y + z - Qx = c \\ xz = 0 \\ x > 0, z > 0, \quad y \in \mathbb{R}^m \end{cases} \quad (2.1)$$

2.3 Méthode de trajectoire centrale avec point

L'idée fondamentale de la méthode de trajectoire centrale consiste à modifier la dernière équation du système (2.1) en y introduisant un terme de perturbation.

$$\Leftrightarrow \begin{cases} Ax = b \\ A^t y + z - Qx = c \\ xz = \mu e, \quad x > 0, z > 0, y \in \mathbb{R}^m \end{cases} \quad (2.2)$$

Tel que $\mu > 0$, $e = (1, 1, \dots, 1)^t \in \mathbb{R}^n$

L'ensemble :

$$\{x(\mu), y(\mu), z(\mu); \mu > 0\},$$

S'appelle la trajectoire centrale. De plus, si $\mu \rightarrow 0$, alors $x^t z$ tend vers le zéro c-à-d trouver une solution optimale pour (P) et (D) revient à résoudre le système.

Dans ce travail, on s'intéresse à la résolution de (P) et (D) pour la méthode de TC avec un point Pour ce la on remplace la dernière équation de système (2.2) par :

$$xz = \tau, \quad \text{où } \tau > 0, \tau \in \mathbb{R}^n.$$

et on obtient le système suivant :

$$\begin{cases} Ax = b \\ A^t y + z - Qx = c \\ xz = \tau, x > 0, z > 0, y \in \mathbb{R}^m. \end{cases} \quad (2.3)$$

Remarque 2.1 : Si $\tau = \mu e$ avec $\mu > 0$, alors la trajectoire pondérée correspond parfaitement à la trajectoire centrale classique.

Le système (2.3) devient :

$$\Leftrightarrow \begin{cases} Ax - b = 0, \\ A^t y + z - Qx - c = 0, \\ xz - \tau = 0. \end{cases} \quad (2.4)$$

Ce système d'équations non linéaires est résolu par l'application la méthode de Newton à la fonction :

$$F(x, y, z) = 0.$$

où $F : \mathbb{R}^{2n+m} \rightarrow \mathbb{R}^{2n+m}$ définie par :

$$F(x, y, z) = \begin{pmatrix} Ax - b \\ A^t y + z - Qx - c \\ xz - \tau \end{pmatrix}.$$

Soient $(x, y, z) \in F_P^+ \times F_D^+$, tel que $xz \neq \tau$. La direction de Newton $(\Delta x, \Delta y, \Delta z)$ est la solution de système linéaire :

$$\nabla F(x, y, z) \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = -F(x, y, z).$$

On va calculer la matrice jacobienne de F :

$$\nabla F(x, y, z) = \begin{pmatrix} A & 0 & 0 \\ -Q & A^t & I_n \\ Z & 0 & X \end{pmatrix}.$$

En ce point est l'unique solution du système d'équations linéaire suivante :

$$\begin{pmatrix} A & 0 & 0 \\ -Q & A^t & I \\ Z & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \tau - Xz \end{pmatrix} \quad (2.5)$$

où $X := \text{diag}(x), Z := \text{diag}(z)$.

Sous nos hypothèses et le fait que $\text{rang}(A) = m$, le système (2.5) admet une solution unique $(\Delta x, \Delta y, \Delta z)$. Ainsi, une nouvelle itération de Newton pondérée complète est construite comme

suit :

$$x_+ := x + \Delta x; \quad y_+ := y + \Delta y; \quad z_+ := z + \Delta z \quad (2.6)$$

Pour simplifier les choses, nous définissons les vecteurs :

$$v := \sqrt{xz} \quad \text{et} \quad d := \sqrt{xz^{-1}}.$$

Le vecteur d est utilisé pour mettre à l'échelle les vecteurs x et z vers le même vecteur v comme suit :

$$d^{-1}x = dz = v \quad (2.7)$$

et aussi pour les directions originales vers les directions mises à l'échelle :

$$d_x = d^{-1}\Delta x \quad \text{et} \quad d_z = d\Delta z.$$

Il en découle que :

$$x\Delta z + z\Delta x = v(d_z + d_x) \quad (2.8)$$

et

$$d_x^T d_z = \Delta_x^T \Delta z = \Delta_x^T Q \Delta x \geq 0 \quad (2.9)$$

puisque Q est une matrice semi-définie positive.

Ainsi, en utilisant (2.7), (2.8) et (2.9), le système (2.5) devient :

$$\begin{pmatrix} \bar{A} & 0 & 0 \\ -\bar{Q} & \bar{A}^T & I \\ I & 0 & I \end{pmatrix} \begin{pmatrix} d_x \\ d_y \\ d_z \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ p_v \end{pmatrix} \quad (2.10)$$

où

$$p_v = v^{-1}(\tau - v^2) \quad (2.11)$$

et $\bar{A} = DAD$ et $\bar{Q} = DQD$ avec $D := \text{diag}(d)$.

Dans la sous-section suivante, nous décrivons l'algorithme générique faisable de suivi de chemin primal-dual pondéré pour résoudre le CQO.

2.3.1 La mesure de proximité

Dans cette partie, on définit pour tout vecteur positif v , une mesure de proximité basée sur la norme 2 comme suit :

$$\delta(x, z; \tau) = \frac{\|p_v\|}{2\sqrt{\min(\tau)}} = \frac{\|v^{-1}(\tau - v^2)\|}{2\sqrt{\min(\tau)}} \quad (2.12)$$

On peut facilement vérifier que :

$$\delta(x, z; \tau) = 0 \Leftrightarrow v^2 = \tau \Leftrightarrow xz = \tau.$$

Ainsi, la valeur $\delta(x, z; \tau)$ mesure la distance d'un point (x, y, z) au trajectoire pondéré $(x(\tau), y(\tau), z(\tau))$.

On définit une autre mesure $\sigma_C(\tau)$ comme suit :

$$\sigma_C(\tau) = \frac{\max(\tau)}{\min(\tau)} \quad (2.13)$$

Le rôle de $\sigma_C(\tau)$ est de mesurer l'appartenance de τ au trajectoire central.

Ici,

$$\min(\tau) = \min_i(\tau_i).$$

Et de même

$$\max(\tau) = \max_i(\tau_i).$$

Notons que dans (2.13), $\sigma_C(\tau) \geq 1$, avec égalité si $\tau = \mu e$.

Maintenant, on va présenter un algorithme de points intérieurs de type primal-dual pour résoudre approximativement un problème quadratique convexe.

2.3.2 Algorithme de TC avec poids

Algorithme de TC avec poids de type Primal-Dual pour PQC

Entrée

Un paramètre seuil $0 < \delta < 1$ (par défaut $\delta = \frac{1}{\sqrt{2}}$);

un paramètre de précision $\varepsilon > 0$;

un paramètre fixe de mise à jour de barrière $0 < \theta < 1$ (par défaut $\theta = \frac{1}{2\sqrt{n}\sigma_c(\tau^0)}$);

un point initial (x^0, y^0, z^0) qui vérifie la CPI et τ^0 tel que $\delta(x^0, z^0; \tau^0) \leq \frac{1}{\sqrt{2}}$.

Début

$x := x^0; y := y^0; z := z^0; \tau := \tau^0;$

tant que $x^T z \geq \varepsilon$ **faire**

Début

$\tau := (1 - \theta)\tau;$

Résoudre le système (2.5) pour obtenir la direction $(\Delta x, \Delta y, \Delta z);$

Mettre à jour $x := x + \Delta x, y := y + \Delta y, z := z + \Delta z;$

Fin

Fin

Algorithme 2.3

On note que si $\theta = n$ alors l'algorithme est dit à petit pas, si $\theta = \text{constant}$ alors l'algorithme est dit à grand pas.

2.3.3 convergence de l'algorithme

Dans le lemme suivant, nous énonçons quelques résultats techniques utiles qui seront utilisés plus tard dans l'analyse de l'algorithme.

Lemme 2.1 [[6],Lemme 3.1 en 2014]

Soit (d_x, d_z) une solution de (2.10) $\tau > 0$. Si $\delta := \delta(x, z; \tau)$, alors on a :

$$0 \leq d_x^T d_z \leq 2\delta^2 \min(\tau) \quad (2.14)$$

et

$$\|d_x d_z\|_\infty \leq \delta^2 \min(\tau) \quad \text{et} \quad \|d_x d_z\| \leq \sqrt{2} \delta^2 \min(\tau) \quad (2.15)$$

Preuve : Puisque $0 \leq d_x^T d_z$, l'expression (2.14) découle immédiatement de l'égalité suivante :

$$\|d_x\|^2 + \|d_z\|^2 + 2d_x^T d_z = \|d_x + d_z\|^2 = \|p_v\|^2 = 4\delta^2 \min(\tau).$$

Pour (2.15), (voir Lemme C.4 dans [34]), parce que :

$$\|d_x d_z\|_\infty \leq \frac{1}{4} \|p_v\|^2 \quad \text{et} \quad \|d_x d_z\| \leq \frac{1}{2\sqrt{2}} \|p_v\|^2.$$

Cela conclut la preuve.

Analyse de la stricte de faisabilité

Le lemme suivant montre que la faisabilité du pas de Newton complet pondéré est assurée sous la condition $\delta := \delta(x; z; \tau) < 1$.

Lemme 2.2 [[6], Lemme 3.2 en 2014] (condition suffisante)

Soit (x, z) un point primal-dual strictement faisable. Alors $x_+ = x + \Delta x > 0$ et $z_+ = y + \Delta z > 0$ si et seulement si $\tau + d_x d_z > 0$.

Preuve : Pour la première expression, nous avons :

$$\begin{aligned} x_+ z_+ &= (x + \Delta x)(z + \Delta z), \\ &= xz + x\Delta z + z\Delta x + \Delta x \Delta z, \\ &= xz + (\tau - xz) + \Delta x \Delta z, \\ &= \tau + \Delta x \Delta z. \end{aligned}$$

Puis, à partir de l'équation dans (2.9), nous avons :

$$\begin{aligned} x_+ z_+ &= \tau + \Delta x \Delta z, \\ &= \tau + d_x d_z. \end{aligned}$$

Si le pas de Newton complet est strictement faisable $x_+ > 0$ et $z_+ > 0$ alors $x_+z_+ > 0$ et donc $\tau + d_x d_z > 0$.

Pour montrer que x_+ et z_+ sont positifs, on introduit une longueur de pas $\alpha \in [0, 1]$ et on définit :

$$x^\alpha = x + \alpha \Delta x, \quad z^\alpha = z + \alpha \Delta z.$$

Donc $x^0 = x$, $x^1 = x_+$ et des notations similaires pour z , donc $x^0 z^0 = xz > 0$. Nous avons :

$$x^\alpha z^\alpha = (x + \alpha \Delta x)(z + \alpha \Delta z) = xz + \alpha(x\Delta z + z\Delta x) + \alpha^2 \Delta x \Delta z.$$

En utilisant maintenant (2.8), on obtient :

$$x^\alpha z^\alpha = xz + \alpha(\tau - xz) + \alpha^2 \Delta x \Delta z.$$

On suppose que $\tau + d_x d_z > 0$, on déduit que $\tau + \Delta x \Delta z > 0$ ce qui équivaut à $\Delta x \Delta z > -\tau$. Substituant, on obtient :

$$\begin{aligned} x^\alpha z^\alpha &> xz + \alpha(\tau - xz) - \alpha^2 \tau \\ &= (1 - \alpha)xz + \alpha(1 - \alpha)\tau. \end{aligned}$$

Puisque xz et τ sont positifs, il s'ensuit que $x^\alpha z^\alpha > 0$ pour tout $\alpha \in [0, 1]$. Ainsi, aucune des composantes de x^α et z^α ne s'annule pour $\alpha \in [0, 1]$. Comme x^0 et z^0 sont positifs, cela implique que $x^\alpha > 0$ et $z^\alpha > 0$ pour tout $\alpha \in [0, 1]$. Par un argument de continuité, les vecteurs x^α et z^α doivent être positifs, ce qui prouve que $x_+ > 0$ et $z_+ > 0$. Ceci termine la démonstration.

Lemme 2.3 [[6],[Lemme 3.3 en 2014]]

Si $\delta := \delta(x, z; \tau) < 1$, alors le pas de Newton complet est strictement faisable, c'est-à-dire, $x_+ > 0$ et $z_+ > 0$.

Preuve : Dans le Lemme 2.2, nous avons vu que :

$$x_+ z_+ > 0 \quad \text{si} \quad \tau + d_x d_z > 0.$$

Donc $\tau + d_x d_z > 0$ est vrai si :

$$\tau_i + (d_x)_i (d_z)_i > 0, \text{ pour tout } i.$$

Nous avons :

$$\tau_i + (d_x)_i (d_z)_i \geq \tau_i - |(d_x)_i (d_z)_i| \geq \min(\tau) - \|d_x d_z\|_\infty \quad \text{pour tout } i.$$

D'après (2.15), Lemme 2.1, il en découle que :

$$\min(\tau) - \|d_x d_z\|_\infty > \min(\tau)(1 - \delta^2).$$

Ainsi, $\tau + d_x d_z > 0$ est vrai si $\delta < 1$. Cela termine la preuve.

Pour simplifier, on peut écrire

$$v_+ = \sqrt{x_+ z_+}$$

Lemme 2.4 [[6],[Lemme 3.4 en 2014]

Si $\delta < 1$. Alors

$$\|v_+^{-1}\| \leq \frac{1}{\sqrt{\min(\tau)(1 - \delta^2)}}.$$

Preuve : Cela découle directement du Lemme 2.3 et puisque

$$v_+^{-2} = \frac{e}{\tau + d_x d_z}.$$

Convergence quadratique de la mesure de proximité :

Dans le lemme suivant, nous montrons l'influence du pas de Newton complèt pondérée sur la mesure de proximité.

Lemme 2.5 [[6],[Lemme 3.5 en 2014]

Si $\delta < 1$. Alors

$$\delta_+ := \delta(x_+, z_+; \tau) \leq \frac{\delta^2}{\sqrt{2(1 - \delta^2)}}.$$

Preuve : Par définition, nous avons

$$\begin{aligned}\delta_+ &= \frac{1}{2\sqrt{\min(\tau)}} \|v_+^{-1}(\tau - v_+^2)\| \\ &\leq \frac{1}{2\sqrt{\min(\tau)}} \|v_+^{-1}\| \|\tau - v_+^2\|.\end{aligned}$$

Mais $\tau - v_+^2 = -d_x d_z$ et $v_+^{-1} = \frac{e}{\sqrt{\tau + d_x d_z}}$, alors par les Lemme 2.1 et 2.4. On a

$$\begin{aligned}\delta_+ &= \frac{1}{2\sqrt{\min(\tau)}} \cdot \left\| \frac{d_x d_z}{\sqrt{\tau + d_x d_z}} \right\|, \\ &= \frac{1}{2\sqrt{\min(\tau)}} \cdot \frac{\|d_x d_z\|}{\|\sqrt{\tau + d_x d_z}\|}, \\ &\leq \frac{1}{2\sqrt{\min(\tau)}} \cdot \frac{\sqrt{2} \min(\tau) \delta^2}{\sqrt{\min(\tau) - \|d_x d_z\|_\infty}}, \\ &\leq \frac{1}{2\sqrt{\min(\tau)}} \cdot \frac{\sqrt{2} \min(\tau) \delta^2}{\sqrt{\min(\tau)(1 - \delta^2)}}, \\ &\leq \frac{\delta^2}{\sqrt{2(1 - \delta^2)}}.\end{aligned}$$

Cela termine la preuve.

Corollaire

Si $\delta < 1$, alors $\delta_+ \leq \delta^2$ ce qui indique une convergence quadratique de la mesure de proximité lorsque les itérations sont proche de la trajectoire. De plus, si $\delta \leq \frac{1}{\sqrt{2}}$, alors $\delta_+ \leq \frac{1}{2}$.

La mise à jour du τ

Le lemme suivant présente l'influence de la mesure de proximité avec la mise à jour du paramètre de barrière $\tau_+ = (1 - \theta)\tau$ sur le processus de Newton durant la trajectoire pondéré.

Lemme 2.6 [[6],[Lemme 3.6 en 2014]]

Si $\delta(x, z; \tau) < 1$ et $\tau_+ = (1 - \theta)\tau$ où $0 < \theta < 1$, alors

$$\delta(x_+, z_+; \tau_+) \leq \frac{\theta}{2\sqrt{1 - \theta}\sqrt{1 - \delta^2}} \sqrt{n}\sigma_c(\tau) + \frac{1}{\sqrt{2(1 - \theta)}} \delta_+.$$

De plus, si $\delta \leq \frac{1}{\sqrt{2}}$, $\theta = \frac{1}{2\sqrt{n}\sigma_C(\tau)}$ et $n \geq 3$, alors nous avons :

$$\delta(x_+, z_+; \tau_+) \leq \frac{1}{\sqrt{2}}.$$

Preuve : Soit $\delta(x_+, z_+; \tau_+)$ et $\tau_+ = (1 - \theta)\tau$ avec $0 < \theta < 1$. alors, par définition, nous avons :

$$\begin{aligned} \delta(x_+, z_+; \tau_+) &= \frac{1}{2\sqrt{\min(\tau_+)}} \|v_+^{-1}(\tau_+ - v_+^2)\|, \\ &= \frac{1}{2\sqrt{1-\theta}\sqrt{\min(\tau)}} \|v_+^{-1}(\tau_+ - v_+^2)\|, \\ &= \frac{1}{2\sqrt{1-\theta}\sqrt{\min(\tau)}} \|v_+^{-1}(\tau_+ - \tau + \tau - v_+^2)\|, \\ &\leq \frac{1}{2\sqrt{1-\theta}\sqrt{\min(\tau)}} (\|v_+^{-1}\| (\|\tau_+ - \tau\| + \|\tau - v_+^2\|)). \end{aligned}$$

Maintenant, puisque $\tau - v_+^2 = -d_x d_z$ et $\tau_+ - \tau = -\theta\tau$ et d'après les Lemmes 2.1 et 2.4 ainsi que le fait que $\|\tau\| \leq \sqrt{n}\|\tau\|_\infty$, on obtient :

$$\begin{aligned} \delta(x_+, z_+; \tau_+) &\leq \frac{1}{2\sqrt{1-\theta}\sqrt{\min(\tau)}\sqrt{1-\delta^2}} [\|\theta\tau\| + \|d_x d_z\|], \\ &\leq \frac{1}{2\sqrt{1-\theta}\sqrt{\min(\tau)}\sqrt{1-\delta^2}} [\|\theta\tau\| + \min(\tau)\delta^2], \\ &\leq \frac{\theta\|\tau\|}{2\sqrt{1-\theta}\sqrt{\min(\tau)}\sqrt{1-\delta^2}} + \frac{\delta^2}{2\sqrt{1-\theta}\sqrt{1-\delta^2}}, \\ &\leq \frac{\theta\sqrt{n}\|\tau\|_\infty}{2\sqrt{1-\theta}\sqrt{\min(\tau)}\sqrt{1-\delta^2}} + \frac{\delta^2}{2\sqrt{1-\theta}\sqrt{1-\delta^2}}, \\ &= \frac{\theta\sqrt{n}\max(\tau)}{2\sqrt{1-\theta}\sqrt{\min(\tau)}\sqrt{1-\delta^2}} + \frac{\delta^2}{2\sqrt{1-\theta}\sqrt{1-\delta^2}}. \end{aligned}$$

En utilisant le Lemme 2.5 et l'équation (2.13), on obtient :

$$\delta(x_+, z_+; \tau_+) \leq \frac{\theta\sqrt{n}\sigma_C(\tau)}{2\sqrt{1-\theta}\sqrt{1-\delta^2}} + \frac{\delta_+}{\sqrt{2(1-\theta)}}.$$

Si $\theta = \frac{1}{2\sqrt{n}\sigma_C(\tau)}$ et sachant que $\sigma_C(\tau) \geq 1$, et pour $n \geq 4$, on a $\theta \leq \frac{1}{4}$. De plus, si $\delta \leq \frac{1}{\sqrt{2}}$, alors d'après le Corollaire, on a $\delta_+ \leq \frac{1}{2}$. Finalement, les inégalités ci-dessus donnent $\delta(x_+, z_+; \tau_+) \leq \frac{1}{\sqrt{2}}$. Cela complète la preuve. \square

Proposition 2.1 Pour chaque itération k , on a :

$$\sigma_c(\tau^k) = \sigma_c(\tau^0).$$

Preuve :

À l'itération k , on a :

$$\begin{aligned} \sigma_c(\tau^k) &= \frac{\max((\tau^k))}{\min((\tau^k))}, \\ &= \frac{\max((1-\theta)^k(\tau^0))}{\min((1-\theta)^k(\tau^0))}, \\ &= \frac{(1-\theta)^k \max((\tau^0))}{(1-\theta)^k \min((\tau^0))}, \\ &= \frac{\max((\tau^0))}{\min((\tau^0))} = \sigma_c(\tau^0). \end{aligned}$$

Ce qui complète la preuve.

On note que, dans toutes les itérations produites par l'algorithme 2.3, on a $\sigma_C(\tau) = \sigma_C(\tau^0)$. Ainsi on déduit du Lemme 2.6 que, pour la valeur $\theta = \frac{1}{2\sqrt{n}\sigma_C(\tau^0)}$, les conditions $x, z > 0$ et $\delta(x_+, z_+; \tau_+) \leq \frac{1}{\sqrt{2}}$ sont maintenues pendant l'exécution de cet algorithme.

Cela confirme que l'algorithme 2.3 est bien défini.

L'influence du nouveau itéré en saut dualité

La borne supérieure de l'écart de dualité après un pas complet de Newton est donnée par le lemme suivant .

Lemme 2.7 [[6],[Lemme 3.7 en 2014]

Soit $\delta := \delta(x, z; \tau) \leq \frac{1}{\sqrt{2}}$ et $x_+ = x + \Delta x$; $z_+ = z + \Delta z$. Alors, l'écart de dualité vérifie :

$$x_+^T z_+ \leq (n+1) \max(\tau).$$

Preuve : D'après le Lemme 2.2, on a :

$$x_+ z_+ = \tau + d_x d_z.$$

Donc,

$$\begin{aligned} e^T(x_+z_+) &= e^T \tau + e^T d_x d_z, \\ &= e^T \tau + d_x^T d_z. \end{aligned}$$

D'après l'équation (2.15), le Lemme 2.1 et le fait que $\delta \leq \frac{1}{\sqrt{2}}$, on déduit :

$$\begin{aligned} x_+^T z_+ &\leq e^T \tau + 2\delta^2 \min(\tau), \\ &\leq e^T \tau + \min(\tau). \end{aligned}$$

Puisque $e^T \tau \leq n \max(\tau)$, on obtient :

$$x_+^T z_+ \leq (n+1) \max(\tau).$$

Cela conclut la preuve.

2.3.4 Analyse de la complexité

Le lemme suivant donne une borne supérieure pour le nombre total d'itérations produites par l'algorithme.

Lemme 2.8 [[6],[Lemme 3.8 en 2014]]

Soient x^{k+1} et z^{k+1} le $(k+1)$ -ème itéré produit par l'algorithme 2.3, avec $\tau := \tau^k$. Alors

$$(x^{k+1})^T z^{k+1} \leq \varepsilon$$

si

$$k \geq \frac{1}{\theta} \log \left(\frac{2n \max(\tau^0)}{\varepsilon} \right).$$

Preuve : D'après le Lemme 2.7, on a :

$$(x^{k+1})^T z^{k+1} \leq (n+1) \max((\tau)^k)$$

avec

$$\tau^k = (1 - \theta) \tau^{k-1} = (1 - \theta)^k \tau^0.$$

Donc, nous avons :

$$(x^{k+1})^T z^{k+1} \leq (1 - \theta)^k (n + 1) \max(\tau^0) \leq (1 - \theta)^k 2n \max(\tau^0),$$

puisque $n + 1 \leq 2n$ pour tout $n \geq 1$.

Ainsi l'inégalité $(x^{k+1})^T z^{k+1} \leq \varepsilon$ est satisfaite si

$$(1 - \theta)^k 2n \max(\tau^0) \leq \varepsilon.$$

En prenant les logarithmes, on peut écrire

$$k \log(1 - \theta) \leq \log \varepsilon - \log(2n \max(\tau^0))$$

et puisque $-\log(1 - \theta) \geq \theta$ pour $0 < \theta < 1$, alors l'inégalité est satisfaite si

$$k\theta \geq \log \left(\frac{2n \max(\tau^0)}{\varepsilon} \right).$$

Cela complète la preuve.

Théorème 2.4

Supposons que x^0 et z^0 sont des points initial strictement faisables pour le PQC, $\tau^0 = \frac{x^0 z^0}{2 \max(x^0 z^0)}$, et que $\delta(x^0, z^0; \tau^0) \leq \frac{1}{\sqrt{2}}$ pour $n \geq 3$. Si $\theta = \frac{1}{2\sqrt{n}\sigma_C(\tau^0)}$ alors, l'algorithme nécessite au plus $O(\sqrt{n}\sigma_C(\tau^0) \log \frac{n}{\varepsilon})$ itérations pour obtenir une solution ε -approchée du PQC.

En particulier, si $\tau^0 = \text{Constant}$, alors l'algorithme 2.3 nécessite au plus PQC itérations, ce qui est actuellement la meilleure borne connue pour le nombre d'itérations pour les méthodes à mise à jour courte.

Preuve : En prenant la valeur de θ et τ^0 dans le Lemme 2.8, le résultat découle directement. Cela complète la preuve.

Chapitre 3

Réalisation numérique

Ce chapitre présente une série de tests numériques visant à évaluer la performance de l'algorithme introduit précédemment. Il est important de rappeler que la programmation linéaire et la programmation quadratique s'inscrivent dans le cadre de la programmation convexe, leurs formulations mathématiques respectant les propriétés de convexité, comme expliqué dans Boyd et Vandenberghe [26]. Cette caractéristique permet une application efficace des outils de l'optimisation convexe, rendant ces problèmes particulièrement adaptés aux expérimentations numériques abordées ici.

Dans ce cadre, nous intéressons particulièrement aux expériences numériques mettant en œuvre l'Algorithme : trajectoire centrale avec point de départ donné, l'objectif étant d'évaluer la performance de l'Algorithme 2.3 proposé en fonction des différentes valeurs du paramètre θ . Les expériences ont été réalisées en utilisant le langage MATLAB exécuté sur la version (R2016a) sous Windows 11, avec un processeur Intel core i5, qui offre un environnement puissant pour la modélisation, l'implémentation et l'analyse des algorithmes d'optimisation. On note par la suite :

- $\delta(x, z; \tau)$: La mesure de proximité qui est associée à l'Algorithme 2.3.
- $(x^0, y^0; z^0)$: Un point initial strictement réalisable et vérifié :

$$\delta(x, z; \tau) = \frac{\|v^{-1}(\tau - v^2)\|}{2\sqrt{\min(\tau)}} \leq \frac{1}{\sqrt{2}}.$$

- $\sigma_c(\tau^0) = \frac{\max(\tau^0)}{\min(\tau^0)}$ tel que :

$$\tau^0 = x^0 z^0.$$

- $(x^*; y^*; z^*)$: Une solution optimale du problème primal (P) et dual (D) respectivement.
- **iter** : Le nombre d'itérations générées par l'Algorithme .
- **CPU** : Le temps d'exécution de l'algorithme en secondes.

3.1 Résultats numérique

3.1.1 Programmation linéaire

La programmation linéaire (PL) est un type de problème d'optimisation visant à minimiser une fonction linéaire tout en respectant des contraintes linéaires. Elle est considérée comme l'une des formes les plus courantes et les plus simples d'optimisation sur le plan théorique. Le problème suivant représente un programme linéaire sous forme standard.

$$(P) \quad \min_x c^T x \quad \text{s. à : } Ax = b, x \geq 0,$$

Le dual de P , est donné par :

$$(D) \quad \max_y b^T y \quad \text{s. à : } A^T y + z = c, x \geq 0, z \geq 0,$$

où A est une matrice de type (m, n) , $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $x \in \mathbb{R}^n$, $z \in \mathbb{R}^m$ et $y \in \mathbb{R}^m$.

Problème 1.

Soit le programme linéaire de taille $n = 4$, $m = 2$ suivant :

$$A = \begin{pmatrix} 1 & 2 & 1 & 0 \\ 3 & 2 & 0 & 1 \end{pmatrix}, b = \begin{pmatrix} 2.5 \\ 3 \end{pmatrix},$$

$$c = (-10 \quad -9 \quad -1.5 \quad -2.5)^t.$$

On prend comme un point réalisable initial :

$$x^0 = (0.5 \quad 0.5 \quad 1 \quad 0.5)^t,$$

$$z^0 = (1 \quad 1 \quad 0.5 \quad 0.5)^t,$$

$$y^0 = (-2 \quad -3)^t.$$

Une solution optimale est :

$$x^* = (0.25 \quad 1.125 \quad 0 \quad 0)^t,$$

$$z^* = (0 \quad 0 \quad 0.25 \quad 0.25)^t,$$

$$y^* = (-1.75 \quad -2.75)^t.$$

La valeur optimale pour les deux problème est -12.6250.

Le tableau suivant présente les résultats numériques obtenus par l'Algorithme 2.3 pour des différentes valeurs de θ avec $\varepsilon = 10^{-5}$:

θ	$\frac{1}{2\sqrt{n}\sigma_c(\tau^0)}$	$\frac{1}{\sqrt{n}}$	$\frac{2}{2\sqrt{n}}$	$\frac{1}{3\sqrt{n}}$
iter	47	10	22	35
CPU	0.0186	0.0168	0.0185	0.0175

TABLE 3.1 – Résultats comparatifs basé sur le choix de θ .

Problème 2.

Soit le programme linéaire détaille $n = 7$, $m = 4$ suivant :

$$A = \begin{pmatrix} 7 & 2 & 3 & 1 & -1 & -2 & 4 \\ -4 & -5 & -2 & 3 & -5 & 9 & 6 \\ 2 & 7 & -6 & 7 & -3 & 4 & 2 \\ 6 & -6 & -1 & 7 & 5 & -5 & 3 \end{pmatrix}, \quad b = \begin{pmatrix} 13.5 \\ -3.3 \\ 11.8 \\ 5.6 \end{pmatrix},$$

$$c = (0.5 \quad 0.6 \quad 2.7 \quad -0.9 \quad 0.4 \quad 1.7 \quad 1)^t.$$

On prend comme un point réalisable initial :

$$x^0 = (1 \quad 1.2 \quad 0.5 \quad 0.4 \quad 0.8 \quad 0.5 \quad 1)^t,$$

$$z^0 = (0.5 \quad 1.5 \quad 1 \quad 0.7 \quad 1 \quad 1.5 \quad 0.3)^t,$$

$$y^0 = (0.2 \quad 0.1 \quad -0.2 \quad -0.1)^t.$$

Une solution optimale est :

$$x^* = (1.1513 \quad 0.8972 \quad 0 \quad 0.2145 \quad 0 \quad 0 \quad 0.8580)^t,$$

$$z^* = (0 \quad 0 \quad 1.6079 \quad 0 \quad 1.6538 \quad 0.9482 \quad 0)^t,$$

$$y^* = (0.2614 \quad 0.0843 \quad -0.0549 \quad -0.1471)^t.$$

La valeur optimale pour les deux problème est : 1.7789.

Le tableau suivant présente les résultats numériques obtenus par l'Algorithme 2.3 pour des

différentes valeurs de θ avec $\varepsilon = 10^{-5}$:

θ	$\frac{1}{2\sqrt{n}\sigma_c(\tau^0)}$	$\frac{1}{\sqrt{n}}$	$\frac{1}{2\sqrt{n}}$	$\frac{1}{3\sqrt{n}}$	$\frac{1}{7\sqrt{n}}$	$\frac{1}{9\sqrt{n}}$	$\frac{1}{12\sqrt{n}}$
iter	221	15	33	50	120	154	206
CPU	0.0266	0.0164	0.0188	0.0197	0.0222	0.0246	0.0276

TABLE 3.2 – Résultats comparatifs basé sur le choix de θ .

Problème 3. Soit le programme linéaire de taille $n = 9, m = 5$:

$$A = \begin{pmatrix} 0 & 1 & 2 & -1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 2 & 3 & 4 & -1 & 0 & 1 & 0 & 0 \\ -1 & 0 & -2 & 1 & 2 & 0 & 0 & 1 & 0 \\ 1 & 2 & 0 & -1 & -2 & 0 & 0 & 0 & 1 \\ 1 & 3 & 4 & 2 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad b = \begin{pmatrix} -1.3 \\ 19 \\ 10.1 \\ -3.9 \\ 12.5 \end{pmatrix},$$

$$c = (1.01 \quad 0.59 \quad 1.27 \quad 0.25 \quad 1.04 \quad 0.82 \quad 0.51 \quad 0.22 \quad 1.01)^t.$$

On prend comme un point réalisable initial :

$$x^0 = (0.9 \quad 0.2 \quad 0.5 \quad 4 \quad 1 \quad 0.5 \quad 1.2 \quad 6 \quad 0.8)^t,$$

$$z^0 = (1 \quad 0.5 \quad 1.2 \quad 0.2 \quad 1 \quad 0.8 \quad 0.5 \quad 0.2 \quad 1)^t,$$

$$y^0 = (0.02 \quad 0.01 \quad 0.02 \quad 0.01 \quad 0.01)^t.$$

Une solution optimale est :

$$x^* = (0 \quad 0 \quad 1.1889 \quad 3.863 \quad 0.0185 \quad 0.1667 \quad 0 \quad 8.5778 \quad 0)^t,$$

$$z^* = (1.2666 \quad 0.0333 \quad 0 \quad 0 \quad 0.0001 \quad 0 \quad 0.2333 \quad 0 \quad 1.1333)^t,$$

$$y^* = (0.82 \quad 0.2767 \quad 0.22 \quad -0.1233 \quad -0.19)^t.$$

La valeur optimale pour les deux problèmes est 4.5187.

Le tableau suivant présente les résultats numériques obtenus par l'Algorithme 2.3 pour des différentes valeurs de θ avec $\varepsilon = 10^{-5}$:

Problème 4.

θ	$\frac{1}{2\sqrt{n}\sigma_c(\tau^0)}$	$\frac{1}{\sqrt{n}}$	$\frac{1}{2\sqrt{n}}$	$\frac{1}{3\sqrt{n}}$	$\frac{1}{7\sqrt{n}}$	$\frac{1}{9\sqrt{n}}$	$\frac{1}{12\sqrt{n}}$
iter	479	18	38	58	139	179	239
CPU	0.1305	0.0223	0.0221	0.0239	0.0276	0.0293	0.0322

TABLE 3.3 – Résultats comparatifs basé sur le choix de θ .

Soit le programme linéaire de taille $n = 10, m = 8$ suivant :

$$A = \begin{pmatrix} 3 & -5 & 2 & -8 & -5 & 6 & 10 & 5 & 2 & 1 \\ -2 & 6 & -7 & 5 & 5 & 6 & -9 & -8 & 7 & 9 \\ -9 & -3 & 2 & 4 & -8 & 10 & -5 & 6 & -7 & 3 \\ 1 & -3 & -1 & 5 & -8 & -4 & 4 & 1 & 4 & -2 \\ 4 & -1 & -2 & 5 & -8 & 4 & -1 & -6 & 0 & 7 \\ -3 & 2 & 0 & -5 & 8 & -2 & 1 & 5 & -6 & 5 \\ 8 & -9 & -5 & 5 & -4 & 8 & 5 & 6 & 5 & 5 \\ 1 & 3 & -2 & -1 & 6 & -2 & -4 & 1 & 0 & 5 \end{pmatrix}, b = \begin{pmatrix} 47 \\ 82 \\ -152 \\ -108 \\ -29 \\ 104 \\ 145 \\ 106 \end{pmatrix},$$

$$c = (41 \quad -67 \quad -106 \quad 73 \quad -64 \quad 221 \quad 18 \quad 85 \quad 103 \quad 216)^t.$$

On prend comme un point réalisable initial :

$$x^0 = (12 \quad 1 \quad 2 \quad 1 \quad 14 \quad 5 \quad 3 \quad 4 \quad 2 \quad 6)^t,$$

$$z^0 = (1 \quad 15 \quad 5 \quad 13 \quad 1 \quad 3 \quad 4 \quad 3 \quad 5 \quad 2)^t,$$

$$y^0 = (9 \quad 10 \quad 6 \quad 5 \quad 1 \quad 3 \quad 10 \quad 7)^t.$$

Une solution optimale est :

$$x^* = (11.7039 \quad 0 \quad 0.4038 \quad 0 \quad 13.6563 \quad 4.6543 \quad 2.4190 \quad 3.6440 \quad 1.6626 \quad 5.7012)^t,$$

$$z^* = (0 \quad 11.0520 \quad 0 \quad 32.2823 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0)^t,$$

$$y^* = (11.0592 \quad 10.0844 \quad 6.1357 \quad 4.9128 \quad 0.6864 \quad 1.9020 \quad 9.0003 \quad 9.2566)^t.$$

La valeur optimale pour les deux problèmes est 2347.7.

Le tableau suivant présente les résultats numériques obtenus par l'Algorithme 2.3 pour des différentes valeurs de θ avec $\varepsilon = 10^{-4}$:

Problème 5.

θ	$\frac{1}{2\sqrt{n}\sigma_c(\tau^0)}$	$\frac{1}{\sqrt{n}}$	$\frac{1}{2\sqrt{n}}$
iter	65	20	42
CPU	0.0240	0.0195	0.0248

TABLE 3.4 – Résultats comparatifs basé sur le choix de θ .

Soit le programme linéaire de taille $n = 15, m = 13$ suivant :

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

$$b = \left(139 \ 17 \ 117 \ 42 \ 123 \ 51 \ 168 \ 146 \ -30 \ 170 \ 95 \ 31 \ 112 \right)^t,$$

$$c = (35 \ 86 \ 83 \ 84 \ 120 \ -86 \ 158 \ 91 \ 87 \ 111 \ 104 \ 80 \ 8 \ 125 \ 26)^t.$$

On prend comme un point réalisable initial :

$$x^0 = (5 \ 8 \ 3 \ 6 \ 7 \ 4 \ 2 \ 5 \ 9 \ 3 \ 6 \ 2 \ 4 \ 1 \ 7)^t,$$

$$z^0 = (3 \ 7 \ 5 \ 4 \ 6 \ 2 \ 8 \ 3 \ 1 \ 5 \ 6 \ 3 \ 7 \ 2 \ 4)^t,$$

$$y^0 = (4 \ 6 \ 2 \ 7 \ 3 \ 1 \ 5 \ 8 \ 6 \ 2 \ 4 \ 3 \ 5)^t.$$

Une solution optimale est :

$$x^* = \left(9.9288 \ 8.0340 \ 3.3353 \ 0 \ 4.0528 \ 3.2157 \ 2.6153 \ 7.9968 \ 11.0913 \ 3.8053 \ 2.8887 \ 4.8731 \ 0 \ 1.1920 \ 7.1473 \right)^t,$$

$$z^* = \left(0 \ 0 \ 0 \ 4.4288 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 4.2557 \ 0 \ 0 \ 0 \right)^t,$$

$$y^* = \left(4.3325 \ 8.2385 \ -3.3383 \ 3.8714 \ 2.3740 \ 6.4904 \ 10.4465 \ 9.5253 \ 5.6539 \ -1.0859 \ 4.8209 \ 5.5545 \ 4.6378 \right)^t.$$

La valeur optimale pour les deux problèmes est 5078.4.

Le tableau suivant présente les résultats numériques obtenus par l'Algorithme 2.3 pour des différentes valeurs de θ avec $\varepsilon = 10^{-4}$:

θ	$\frac{1}{2\sqrt{n}\sigma_c(\tau^0)}$	$\frac{1}{\sqrt{n}}$	$\frac{1}{2\sqrt{n}}$	$\frac{1}{3\sqrt{n}}$	$\frac{1}{7\sqrt{n}}$	$\frac{1}{9\sqrt{n}}$	$\frac{1}{12\sqrt{n}}$
iter	1620	27	56	85	201	259	345
CPU	0.1948	0.0276	0.0241	0.0286	0.0397	0.0418	0.0504

TABLE 3.5 – Résultats comparatifs basé sur le choix de θ .

3.1.2 Programmation quadratique

Un problème de programmation quadratique consiste à trouver le minimum d'une fonction quadratique sous des contraintes linéaires.

Le problème suivant est un programme quadratique primal sous la forme standard suivante :

$$(P) \quad \min_x \left\{ c^T x + \frac{1}{2} x^T Q x : Ax = b, x \geq 0 \right\}$$

et son dual :

$$(D) \quad \max_{x,y,z} \left\{ b^T y - \frac{1}{2} x^T Q x : A^T y + z - Qx = c, z \geq 0 \right\}$$

où Q est une matrice carrée symétrique semi-définie positive d'ordre n et A est une matrice de type (m, n) ,

$c \in \mathbb{R}^n, b \in \mathbb{R}^m, x \in \mathbb{R}^n, z \in \mathbb{R}^n$ et $y \in \mathbb{R}^m$.

Problème 1.

Soit le problème quadratique de taille $n = 3, m = 2$ suivant :

$$Q = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad A = \begin{pmatrix} -1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 0.9999 \\ 2 \end{pmatrix},$$

$$c = (-1.9998 \quad -3.9996 \quad 0.0001)^t.$$

On prend comme un point réalisable initial :

$$x^0 = (0.3262 \quad 1.3261 \quad 0.3477)^t,$$

$$z^0 = (0.7247 \quad 0.7247 \quad 2.0722)^t,$$

$$y^0 = (0 \quad -2.0721)^t.$$

Une solution optimale est :

$$x^* = (0.5000 \quad 1.4999 \quad 0)^t,$$

$$z^* = (0 \quad 0 \quad 0.9998)^t,$$

$$y^* = (-0 \quad -0.9997)^t.$$

La valeur optimale pour les deux problèmes est -4.4993.

Le tableau suivant présente les résultats numériques obtenus par l'Algorithme 2.3 pour des différentes valeurs de θ avec $\varepsilon = 10^{-5}$:

θ	$\frac{1}{2\sqrt{n}\sigma_c(\tau^0)}$	$\frac{1}{\sqrt{n}}$	$\frac{1}{2\sqrt{n}}$	$\frac{1}{3\sqrt{n}}$	$\frac{1}{5\sqrt{n}}$	$\frac{1}{7\sqrt{n}}$
Iter	84	9	19	30	51	72
CPU	0.0301	0.0198	0.0220	0.0228	0.0248	0.0236

TABLE 3.6 – Résultats comparatifs basé sur le choix de θ .

Problème 2.

Soit le problème quadratique de taille $n = 4, m = 2$ suivant :

$$Q = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}, \quad A = \begin{pmatrix} -1 & 1 & 1 & 0 \\ 2 & 3 & 0 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 0.3333 \\ 1.9998 \end{pmatrix},$$

$$c = (-0.6666 \quad -6.6666 \quad -0.6666 \quad -0.6666)^t.$$

On prend comme un point réalisable initial :

$$x^0 = (0.3333 \quad 0.3333 \quad 0.3333 \quad 0.3333)^t,$$

$$z^0 = (2 \quad 2 \quad 2 \quad 2)^t,$$

$$y^0 = (-2 \quad -2)^t.$$

Une solution optimale est :

$$\begin{aligned}
 x^* &= (0.2 \quad 0.5333 \quad 0 \quad 0)^t, \\
 z^* &= (0 \quad 0 \quad 1.4134 \quad 0.5067)^t, \\
 y^* &= (-2.08 \quad -1.1733)^t.
 \end{aligned}$$

La valeur optimale pour les deux problèmes est -3.3641.

Le tableau suivant présente les résultats numériques obtenus par l'Algorithme 2.3 pour des différentes valeurs de θ avec $\varepsilon = 10^{-5}$:

θ	$\frac{1}{2\sqrt{n}\sigma_c(\tau^0)}$	$\frac{1}{\sqrt{n}}$
iter	23	11
CPU	0.0277	0.0283

TABLE 3.7 – Résultats comparatifs basé sur le choix de θ .

Problème 3.

Soit le problème quadratique de taille $n = 5$, $m = 3$ suivant :

$$Q = \begin{pmatrix} 20 & 1.2 & 0.5 & 0.5 & -1 \\ 1.2 & 32 & 1 & 1 & 1 \\ 0.5 & 1 & 14 & 1 & 1 \\ 0.5 & 1 & 1 & 15 & 1 \\ -1 & 1 & 1 & 1 & 16 \end{pmatrix}, \quad A = \begin{pmatrix} 1 & 1.2 & 1 & 1.8 & 0 \\ 3 & -1 & 1.5 & -2 & 1 \\ -1 & 2 & -3 & 4 & 2 \end{pmatrix},$$

$$b = (9.3100 \quad 5.4500 \quad 7.0600)^t,$$

$$c = (1 \quad -1.5 \quad 2 \quad 1.5 \quad 3)^t.$$

On prend comme un point réalisable initial :

$$x^0 = (2.4200 \quad 1 \quad 1.55 \quad 2.3 \quad 1.465)^t,$$

$$z^0 = (3.06 \quad 15.719 \quad 8.175 \quad 7.225 \quad 7.87)^t,$$

$$y^0 = (20 \quad 11 \quad 5)^t.$$

Une solution optimale est :

$$\begin{aligned}
x^* &= (2.6604 \quad 0.7035 \quad 1.3244 \quad 2.4894 \quad 1.1645)^t, \\
z^* &= 10^{-6} (0.0402 \quad 0.3224 \quad 0.1380 \quad 0.0963 \quad 0.1428)^t, \\
y^* &= (25.0011 \quad 12.1538 \quad 5.6674)^t.
\end{aligned}$$

La valeur optimale pour les deux problèmes est 175.2459.

Le tableau suivant présente les résultats numériques obtenus par l'Algorithme 2.3 pour des différentes valeurs de θ avec $\varepsilon = 10^{-6}$:

θ	$\frac{1}{2\sqrt{n}\sigma_c(\tau^0)}$	$\frac{1}{\sqrt{n}}$	$\frac{1}{2\sqrt{n}}$	$\frac{1}{3\sqrt{n}}$	$\frac{1}{4\sqrt{n}}$
iter	87	17	37	57	77
CPU	0.0462	0.0297	0.0302	0.0339	0.0341

TABLE 3.8 – Résultats comparatifs basé sur le choix de θ .

Problème 4.

Soit le problème quadratique de taille $n = 10$, $m = 3$ suivant :

$$A = \begin{pmatrix} 1 & -1 & 1.9 & 1.25 & 1.2 & 0.4 & -0.7 & 1.06 & 1.5 & 1.05 \\ 1.3 & 1.2 & 0.15 & 2.15 & 1.25 & 1.5 & 0.4 & 1.52 & 1.3 & 1 \\ 1.5 & -1.1 & 3.5 & 1.25 & 1.8 & 2 & 1.95 & 1.2 & 1 & -1 \end{pmatrix}, \quad b = \begin{pmatrix} 11.6477 \\ 16.6659 \\ 21.2892 \end{pmatrix},$$

$$Q = \begin{pmatrix} 30 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 21 & 0 & 1 & -1 & 1 & 0 & 1 & 0.5 & 1 & 1 \\ 1 & 0 & 15 & -0.5 & -2 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & -0.5 & 30 & 3 & -1 & 1 & -1 & 0.5 & 1 & 1 \\ 1 & -1 & -2 & 3 & 27 & 1 & 0.5 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & -1 & 1 & 16 & -0.5 & 0.5 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0.5 & -0.5 & 8 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & -1 & 1 & 0.5 & 1 & 24 & 1 & 1 & 1 \\ 1 & 0.5 & 1 & 0.5 & 1 & 0 & 1 & 1 & 39 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 11 \end{pmatrix},$$

$$c = \begin{pmatrix} -0.4854 & -0.9915 & 0.0019 & 0.0282 & -0.4765 & 0.0070 & 0.0097 & -0.9971 & -0.4823 & -0.9874 & 0 & 0 \end{pmatrix}^t.$$

On prend comme un point réalisable initial :

$$x^0 = (1.475 \quad 0.868 \quad 1.752 \quad 1.637 \quad 1.560 \quad 1.926 \quad 1.538 \quad 1.416 \quad 1.223 \quad 0.728)^t,$$

$$z^0 = (23.525 \quad 13.117 \quad 3.001 \quad 7.235 \quad 15.434 \quad 0.836 \quad 8.723 \quad 5.656 \quad 23.057 \quad 1.441)^t,$$

$$y^0 = (5.6566 \quad 16.7085 \quad 3.6733)^t.$$

Une solution optimale est :

$$x^* = (0.9635 \quad 0.5093 \quad 1.7398 \quad 1.9042 \quad 1.2428 \quad 2.6264 \quad 1.3219 \quad 1.6170 \quad 0.8236 \quad 0.8968)^t,$$

$$z^* = 10^{-4} \begin{pmatrix} 0.2566 & 0.1593 & 0.0215 & 0.0443 & 0.1380 & 0.0044 & 0.0723 & 0.0353 & 0.2439 & 0.0083 \end{pmatrix}^t,$$

$$y^* = (4.2434 \quad 22.3621 \quad 5.1918)^t.$$

La valeur optimale pour les deux problèmes est 264.1343.

Le tableau suivant présente les résultats numériques obtenus par l'Algorithme 2.3 pour des différentes valeurs de θ avec $\varepsilon = 10^{-4}$:

θ	$\frac{1}{2\sqrt{n}\sigma_c(\tau^0)}$	$\frac{1}{\sqrt{n}}$	$\frac{1}{2\sqrt{n}}$	$\frac{1}{3\sqrt{n}}$	$\frac{1}{6\sqrt{n}}$	$\frac{1}{9\sqrt{n}}$
iter	1478	20	43	65	132	177
CPU	0.1142	0.0192	0.0214	0.0247	0.0281	0.0307

TABLE 3.9 – Résultats comparatifs basé sur le choix de θ .

Problème 5.

Soit le problème quadratique de taille $n = 10, m = 3$ suivant :

$$Q = \begin{pmatrix} 30 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 21 & 0 & 1 & -1 & 1 & 0 & 1 & 0.5 & 1 \\ 1 & 0 & 15 & -0.5 & -2 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & -0.5 & 30 & 3 & -1 & 1 & -1 & 0.5 & 1 \\ 1 & -1 & -2 & 3 & 27 & 1 & 0.5 & 1 & 1 & 1 \\ 1 & 1 & 1 & -1 & 1 & 16 & -0.5 & 0.5 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0.5 & -0.5 & 8 & 1 & 1 & 1 \\ 1 & 1 & 1 & -1 & 1 & 0.5 & 1 & 24 & 1 & 1 \\ 1 & 0.5 & 1 & 0.5 & 1 & 0 & 1 & 1 & 39 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 11 \end{pmatrix},$$

$$A = \begin{pmatrix} 1 & -1 & 1.9 & 1.25 & 1.2 & 0.4 & -0.7 & 1.06 & 1.5 & 1.05 \\ 1.3 & 1.2 & 0.15 & 2.15 & 1.25 & 1.5 & 0.4 & 1.52 & 1.3 & 1 \\ 1.5 & -1.1 & 3.5 & 1.25 & 1.8 & 2 & 1.95 & 1.2 & 1 & -1 \end{pmatrix}, \quad b = \begin{pmatrix} 0.7660 \\ 1.1770 \\ 1.2100 \end{pmatrix},$$

$$c = (-1.7 \quad 4.35 \quad -1.3 \quad -2.15 \quad -1.5 \quad 0.1 \quad 3.05 \quad -0.83 \quad -2.4 \quad 2.95)^t.$$

On prend comme un point réalisable initial :

$$x^0 = (0.1 \quad 0.1 \quad 0.1 \quad 0.1 \quad 0.1 \quad 0.1 \quad 0.1 \quad 0.1 \quad 0.1 \quad 0.1)^t,$$

$$z^0 = (6 \quad 6 \quad 6 \quad 6 \quad 6 \quad 6 \quad 6 \quad 6 \quad 6 \quad 6)^t,$$

$$y^0 = (-1 \quad -1 \quad -1)^t.$$

La solution optimale est :

$$x^* = (0.0973 \quad 0.0000 \quad 0.0236 \quad 0.1725 \quad 0.0810 \quad 0.2061 \quad 0.0000 \quad 0.1124 \quad 0.0730 \quad 0.0000)^t,$$

$$z^* = (0.0000 \quad 1.5550 \quad 0.0000 \quad 0.0000 \quad 0.0000 \quad 0.0000 \quad 0.5016 \quad 0.0000 \quad 0.0000 \quad 3.8706)^t,$$

$$y^* = (-1.5354 \quad 2.0035 \quad 0.5461)^t.$$

La valeur optimale pour les deux problèmes est 0.4532.

Le tableau suivant présente les résultats numériques obtenus par l'algorithme 2.3 pour des différentes valeurs de θ avec $\varepsilon = 10^{-5}$:

θ	$\frac{1}{2\sqrt{n}\sigma_c(\tau^0)}$	$\frac{1}{\sqrt{n}}$
iter	40	19
CPU	0.0254	0.0214

TABLE 3.10 – Résultats comparatifs basé sur le choix de θ .

Conclusion Générale

Dans ce travail, on s'intéresse à l'étude d'un algorithme de trajectoire centrale avec poids de type primal-dual qui est présenté par M.Achache en 2014 (voir [6]). Dont Le but est de résoudre approximativement un problème quadratique convexe. Au début de l'algorithme, on commence par un point primal-dual (x^0, y^0, z^0) strictement réalisable et vérifie :

$$\delta(x^0, z^0, \tau^0) = \frac{\|v^{-1}(\tau^0 - v^2)\|}{2\sqrt{\min \tau^0}} \leq \frac{1}{\sqrt{2}} \quad ; \quad \text{avec } \tau^0 = x^0 z^0$$

qui justifié la difficulté de cette méthode; à chaque itération; l'algorithme utilise le pas de Newton complet pour trouver une Solution approximative optimale primale-duale. il est montré que la complexité de cet algorithme est d'ordre :

$$O\left(\sqrt{n} \sigma_c(\tau^0) \log \frac{n}{\varepsilon}\right)$$

avec

$$\sigma_c(\tau^0) = \frac{\max(\tau^0)}{\min(\tau^0)}.$$

Notons que les tests numériques comparatifs basés sur les différents choix de θ , pour voir l'influence de ce dernier sur le comportement numérique de cet algorithme est la contribution réalisé par cette étude .

Références

- [1] V. Klee and G. J. Minty, “How good is the simplex algorithm ?” in *Inequalities*, O. Shisha, Ed. New York : Academic Press, 1972, pp. 159–175.
- [2] L. G. Khachiyan, “A polynomial algorithm in linear programming,” *Soviet Mathematics Doklady*, vol. 20, pp. 191–194, 1979.
- [3] N. Karmarkar, “A new polynomial-time algorithm for linear programming,” in *Proceedings of the 16th Annual ACM Symposium on Theory of Computing*, 1984, pp. 303–311.
- [4] M. Kojima, S. Mizuno, and A. Yoshise, “A primal-dual interior point algorithm for linear programming,” in *Progress in Mathematical Programming : Interior Point and Related Methods*, N. Megiddo, Ed. New York : Springer-Verlag, 1989, pp. 29–47.
- [5] S. Mehrotra, “On an implementation of primal-dual predictor-corrector algorithms,” in *Second Asilomar Workshop on Progress in Mathematical Programming*, Asilomar, CA, 1990, presented at the workshop.
- [6] M. Achache, “A weighted full-newton step primal-dual interior point algorithm for convex quadratic optimization,” *Statistics, Optimization and Information Computing*, vol. 2, pp. 21–32, Mar. 2014.
- [7] G. Strang, *Introduction to Linear Algebra*, 5th ed. Wellesley-Cambridge Press, 2016.
- [8] L. Guerra, “Méthodes de points intérieurs et fonctions noyaux pour l’optimisation quadratique semi-définie convexe,” Thèse de doctorat, Université Ferhat Abbas Sétif 1, Faculté des Sciences, Département de Mathématiques, 2019.
- [9] “Dernière compilation,” 2022, version Web : <http://botafogo.saitis.net>, compilée le 2022-12-09 à 06 :44 :16+01 :00.

- [10] S. L. Digabel, “12 matrices symétriques et matrices définies positives,” 2025, hiver 2025, Version 1, Jan. 6.
- [11] Équipe de Mathématiques Appliquées, “Mt23-algèbre linéaire,” Jan. 2012.
- [12] R. T. Rockafellar, *Convex Analysis*. Princeton, NJ, USA : Princeton University Press, 1970.
- [13] N. Boudjellal, “Méthodes de points intérieurs pour la programmation quadratique convexe : théorie, algorithmes et applications,” Thèse de doctorat, Université Ferhat Abbas Sétif-1, Département de Mathématiques, Sétif, Algérie, 2021.
- [14] J.-B. Hiriart-Urruty and C. Lemaréchal, *Cours d’optimisation*. Springer-Verlag, 1993.
- [15] I. S. Ciupercai, “Cours optimisation en master m1 sitn,” cours non publié.
- [16] Z. Ramdani, “Cours de programmation linéaire,” 2021.
- [17] A. Rondepierre, “Méthodes numériques pour l’optimisation non linéaire déterministe,” 2018, année universitaire 2017–2018.
- [18] H. J. Ferreau, “An online active set strategy for fast solution of parametric quadratic programs with applications to predictive engine control,” Ph.D. dissertation, University of Heidelberg, 2006.
- [19] K. J. Arrow and G. Debreu, “Existence of an equilibrium for a competitive economy,” *Econometrica*, vol. 22, pp. 265–290, 1954.
- [20] G. Debreu, “Integration of correspondences,” in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 2 : Contributions to Probability Theory, Part 1*. The Regents of the University of California, 1967.
- [21] V. N. Vapnik, *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.
- [22] B. Schölkopf and A. J. Smola, *Learning With Kernels : Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, 2002.
- [23] N. Cristianini and J. Shawe-Taylor, *Introduction to Support Vector Machines and other Kernel-based Learning Methods*. United Kingdom : Cambridge University Press, 2000.

- [24] H. M. Markowitz, "Portfolio selection," *Journal of Finance*, vol. 7, no. 1, pp. 77–91, 1952.
- [25] H. Markowitz, "Portfolio selection," *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, 1952.
- [26] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [27] F. J. Fabozzi, P. N. Kolm, D. A. Pachamanova, and S. M. Focardi, *Robust Portfolio Optimization and Management*. Wiley, 2007.
- [28] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York : Springer, 2006.
- [29] M. Frank and P. Wolfe, "An algorithm for quadratic programming," *Naval Research Logistics Quarterly*, vol. 8, pp. 95–110, 1956.
- [30] P. Hammer, P. Hansen, and B. Simeone, "Roof duality complementation and persistency in quadratic 0-1 optimization," *Mathematical Programming*, vol. 28, pp. 121–155, 1984.
- [31] H. A. L. Thi, "An efficient algorithm for globally minimizing a quadratic function under convex quadratic constraints," *Mathematical Programming*, vol. 87, no. 3, pp. 401–426, 2000.
- [32] R. T. Rockafellar, *Convex Analysis*. Princeton, NJ : Princeton University Press, 1970.
- [33] S. J. Wright, *Primal-dual interior point methods*. SIAM, 1997.
- [34] C. Roos, T. Terlaky, and J.-P. Vial, *Theory and Algorithms for Linear Optimization : An Interior Point Approach*. Chichester, UK : John Wiley & Sons, 1997.

Résumé :

Dans ce travail, on a étudié un algorithme primal-dual de points intérieurs avec poids pour résoudre un problème quadratique convexe (PQC) (voir [6]). La stratégie de cet algorithme est de remplacer l'équation de centralité par le poids. À chaque itération, on utilise le pas de Newton complet pour trouver une solution approximative. Cette étude est suivie par des tests numériques comparatifs basés sur le choix de θ pour voir l'influence de ce dernier sur le comportement numérique de cet algorithme.

Mots clés :

Méthode de points intérieurs ; programmation quadratique convexe ; Algorithme primal-dual ; tests numériques comparatifs.

Abstract:

In this work, we have studied a primal-dual interior-point algorithm with weights for solving a convex quadratic problem (CQP) (see [6]). The strategy of this algorithm is to replace the centrality equation by the weight, with at each iteration, the full Newton step is used to find an approximate solution. This is followed by comparative numerical tests based on the choice of θ to see its influence on the numerical behavior of this algorithm.

Keywords:

Interior point method; convex quadratic programming; primal-dual algorithm; comparative numerical tests.

ملخص:

في هذا العمل، درسنا خوارزمية نقطة داخلية أولية ثنائية ذات أوزان لحل مشكلة تربيعية محدبة (PQC) (انظر [6]). وتمثل استراتيجية هذه الخوارزمية في استبدال معادلة المركزية بالوزن. مع كل تكرار، يتم استخدام خطوة نيوتن الكاملة لإيجاد حل تقريبي. يتبع هذه الدراسة اختبارات عددية مقارنة تعتمد على اختيار θ لمعرفة تأثير هذا الأخير على السلوك العددي لهذه الخوارزمية.

الكلمات المفتاحية:

طريقة النقطة الداخلية؛ البرمجة التربيعية المحدبة؛ خوارزمية أولية ثنائية؛ اختبارات عددية مقارنة.