

**RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA
RECHERCHE SCIENTIFIQUE**



**UNIVERSITE FERHAT ABBAS SETIF 1
FACULTE DES SCIENCES
DEPARTEMENT DE MATHÉMATIQUES**



THESE

Présentée par :

GOUTALI Moufida

Pour obtenir le titre de Doctorat en Sciences

OPTION

MATHÉMATIQUES APPLIQUÉES

THEME

Complexité et implémentation numérique d'une méthode de points intérieurs pour la programmation convexe

Soutenue le : 15 / 11 / 2018

Devant le jury composé de :

Président :	Mr. N. BENSALÉM	Prof. U.F.A. Sétif-1
Rapporteur :	Mr. M. ACHACHE	Prof. U.F.A. Sétif-1
Examineurs :	Mr. N. BENHAMIDOUCHE	Prof. U.M.B. M'sila
	Mr. M. ZERGUINE	M.C.A U.H.L.Batna
Invité :	Mr. B.MERIKHI	M.C.A U.F.A. Sétif-1

Année 2017/ 2018

Remerciements

Je voudrais tout d'abord remercier vivement mon directeur de thèse Monsieur **M. ACHACHE**, Professeur à l'Université Sétif-1, pour sa disponibilité, ses conseils et ses encouragements qui m'ont été très estimables, son aide et discussion et son support pendant l'élaboration de cette thèse.

Mes sincères remerciements s'adressent également à Monsieur **N. BENSALÉM**, Professeur à l'Université de Sétif-1. Je tiens à lui exprimer ma gratitude pour l'honneur qu'il me fait en présidant ce jury.

Je suis très honoré à remercier de la présence à mon jury de thèse et je tiens à remercier : Monsieur **N. BENHAMIDOUCHE** Professeur à l'Université de Mohamed Boudiaf M'sila pour l'honneur qu'il m'a fait en acceptant d'être membre de mon jury de thèse. Je tiens à l'assurer de ma profonde reconnaissance pour l'intérêt qu'il porte à ce travail.

Monsieur **M. ZERGUINE** Docteur en mathématique et Maître de conférences classe -A - Université Hadj Lakhdar Batna. pour l'honneur qu'il m'a fait pour sa participation à mon jury de thèse en qualité de rapporteur de mon travail

Je tiens également à exprimer ma reconnaissance à Monsieur **B. MERIKHI** Professeur à l'Université Sétif-1, pour ses encouragements.

Les remerciements sont aussi dus à tous les membres de l'équipe d'optimisation, du Laboratoire LMFN de Sétif, leurs remarques et leurs conseils m'ont permis d'améliorer la qualité de ce travail, à tous les membres du conseil scientifique, à tous les enseignants de l'Université de Sétif-1.

Merci également à tous mes amis et mes collègues.

Je remercie chaleureusement mes parents pour leurs soutiens qui m'a été bien utile durant ma thèse.

A titre plus personnel, Je remercie mon mari, **M.KHALFOUNE** pour sa grande patience, l'encouragement ininterrompu et ses nombreux conseils tout au long de ma thèse.

M. GOUTALI

Contents

1	Analyse convexe et programmation mathématique	14
1.1	Éléments d'analyse convexe	14
1.1.1	Définitions	14
1.1.2	Ensemble affine	15
1.1.3	Ensemble convexe	15
1.1.4	Fonction convexe	16
1.1.5	Cônes Convexes	17
1.2	Programmation Mathématique	18
1.2.1	Solutions optimales locales et globales	18
1.2.2	Classification d'un programme mathématique	19
1.2.3	Qualification des contraintes	19
1.2.4	Résolution d'un programme mathématique	20
1.3	Programmation convexe	21
1.3.1	Existence (unicité) d'une solution optimale	22
1.3.2	Conditions d'optimalité	22
1.3.3	Le dual d'un programme convexe	23
1.4	Algorithme d'optimisation	24
1.4.1	Description	24
1.4.2	Convergence	24
1.5	Méthode de Newton-Raphson pour un système non linéaire	27

1.6	Méthodes de résolution d'un programme mathématique	28
1.6.1	Méthodes de type gradient	28
1.6.2	Méthodes simpliciales	29
1.6.3	Méthodes de points intérieurs	29
2	Méthodes de points intérieurs de type de TC pour (PCCL)	32
2.1	Existence et unicité de la trajectoire centrale (suivi de chemin) pour PCCL	32
2.1.1	Le problème perturbé	34
2.1.2	Les conditions d'optimalité de $(K.K.T)_\mu$	34
2.1.3	Le chemin central est bien défini	36
2.1.4	Le Théorème principal de la trajectoire centrale	37
2.2	Principe des méthodes de trajectoire centrale	40
2.2.1	Concept de proximité ou de centralisation	41
2.2.2	Mise à jour du paramètre barrière μ	42
2.2.3	Itération de Newton	42
2.2.4	Pas de déplacement	43
2.3	Algorithme générale primal-dual de suivi de trajectoire	44
3	Méthodes primal-dual pour (PCCL)	45
3.1	Directions de Newton	45
3.1.1	Description algorithmique	48
3.1.2	Algorithme	48
3.2	Convergence de l'algorithme et analyse de la complexité	49
3.2.1	Résultats préliminaires	49
3.2.2	Analyse de la faisabilité	50
3.2.3	Analyse de la complexité	56
4	Simulation numérique	58
4.1	Mise en œuvre de cet algorithme	58
4.1.1	Paramètre barrière	58

4.1.2	Directions de déplacement	59
4.1.3	Point d'initialisation	59
4.2	Résultats numériques	60
4.2.1	Programmation linéaire	60
4.2.2	Programmation quadratique	66
4.2.3	Cas d'une fonction convexe	72
4.2.4	Commentaires	78
4.3	Perspectives	79
4.4	Conclusion générale	85

Résumé

Dans cette thèse, on s'intéresse à l'étude théorique (notamment la complexité polynomiale) et à l'implémentation numérique d'une méthode de points intérieurs de trajectoire centrale de type primal-dual pour résoudre les problèmes de la programmation convexe à contraintes linéaires (PCCL). Dans notre étude, nous proposons de nouveaux paramètres qui décrivent le paramètre barrière et le seuil qui mesure la taille du voisinage de la trajectoire centrale. À travers ces derniers, un algorithme primal-dual à pas court et d'itération de Newton complet bien défini est présenté et de plus sa complexité est calculée. L'efficacité numérique de cet algorithme est confirmée par des tests numériques.

Introduction

Les méthodes de points intérieurs sont apparues dans les années cinquante. Ces méthodes reposent sur la fonction barrière logarithmique définie en 1955, par Frisch ([24]). Cette fonction barrière logarithmique était largement étudiée dans les années soixante, par Fiacco et McCormick. Les itères générés par les méthodes basées sur cette fonction sont strictement réalisables: ils restent à l'intérieur du domaine réalisable, d'où la terminologie de ces méthodes. L'engouement pour ces méthodes dans les années soixante est dû au développement de méthodes performantes en optimisation sans contrainte, citons notamment le code SUMT développé par Fiacco et McCormick, en 1968. Le problème du mauvais conditionnement des matrices hessiennes des fonctions barrières d'abord étudié par Murray, en 1969-1971 puis par Lootsma, en 1972 ralentit le développement de ces méthodes au début des années soixante-dix. On redécouvra ces méthodes plus tard grâce à l'optimisation linéaire. Avant 1984, tout problème d'optimisation linéaire se résolvait par la méthode du simplexe développé par Dantzig, en 1963 ou par une variante de celle-ci. Des recherches ont été menées pour mettre au point une autre méthode mais aucune de celles proposées n'améliorait celle du simplexe. Aussi, pendant une quarantaine d'années, cette méthode domina l'optimisation linéaire. Puis, dans les années soixante-dix, la théorie de la complexité devient une part intégrante de l'optimisation linéaire, si bien qu'on demanda aux méthodes développées de converger en un temps polynômial, c'est-à-dire de résoudre le problème en un nombre d'opérations qui doit être borné par un polynôme en fonction de la taille du problème([27]). Mais la méthode du simplexe n'a pas cette propriété, comme l'ont montré Klee et Minty, en 1972. On se demanda alors si un algorithme d'optimisation linéaire avait cette propriété. En 1979, Khachian proposa un algorithme de programmation linéaire appelé méthode des ellipses de Khachian. Bien que convergeant polynômialement en théorie, cet algorithme convergeait en pratique moins vite que le simplexe([27]). Toutefois, Khachian montra théoriquement l'existence d'algorithmes à complexité polynômiale. Il restait maintenant à en trouver qui soient efficaces en pratique. Ce que fit Karmarkar en 1984 ([35]), il proposa en effet un algorithme

de réduction du potentiel de points intérieurs à complexité polynômiale pour résoudre les problèmes d'optimisation linéaire. Cela provoqua un regain d'intérêt pour les méthodes de points intérieurs, aussi bien en optimisation linéaire qu'en optimisation non linéaire. Dans les années 90, la naissance des méthodes primal-dual de points intérieurs de trajectoire centrale pour résoudre (PL). Elles sont souvent les méthodes de points intérieurs les plus efficaces pour résoudre les problèmes (PL). En théorie, leurs algorithmes associés admettant la meilleure complexité polynomiale et d'autre part ils sont de type Newton ce qui mène donc à une efficacité numérique. Leurs extensions pour d'autres problèmes d'optimisation et de la programmation mathématiques à savoir la programmation quadratique convexe (PQC), la programmation semi définie (SDO) et les problèmes de complémentarité linéaire (PCL) sont d'une importance capitale.

Notre but dans cette thèse est de traiter la programmation convexe à contraintes linéaires (PCCL) par les méthodes de points intérieurs de (TC) et de type primal-dual et à pas court. La programmation PCCL est un cas particulier de la programmation non linéaire et elle généralise la programmation linéaire et la programmation quadratique convexe. Notre but exactement est de proposer de nouveaux paramètres notamment le paramètre barrière et le seuil qui mesure la taille du voisinage de la trajectoire centrale. À travers ces deux nouveaux paramètres un algorithme de TC primal-dual à pas de Newton complet est présenté, et sa complexité est aussi calculée. L'efficacité numérique de cet algorithme est confirmée par des tests numériques.

La thèse est présentée en **quatre chapitres**.

Dans le **premier chapitre**, on présente les notions de base de l'analyse convexe et de la programmation mathématique et une idée générale sur quelques méthodes de points intérieurs pour la programmation convexe.

Dans le **deuxième chapitre**, nous montrons l'existence et l'unicité de la trajectoire centrale (suivi de chemin) pour PCCL et ainsi le principe des méthodes primal-dual de TC pour résoudre les problèmes de la programmation convexe.

Dans le **troisième chapitre**, un algorithme de points intérieurs de type primal-dual de

trajectoire centrale à pas de Newton complet et à pas court est proposé pour PCCL. Une étude théorique complète est faite. La convergence quadratique des itérations de Newton complet vers la trajectoire centrale est démontrée ainsi est calculée de la complexité polynomiale de l'algorithme qui est de l'ordre $\mathcal{O}(\sqrt{n} \log \frac{n}{\epsilon})$.

Dans le **dernier chapitre**, on concentre sur les expériences numériques de cet algorithme sur quelques problèmes de la programmation convexe inclut les problèmes de la programmation linéaire et quadratique, convexe. Enfin, nous clôturons cette thèse par une conclusion et des perspectives.

Notations et terminologie

(PM)	:	Programmation mathématique.
(PL)	:	Programmation linéaire.
(PQ)	:	Programmation quadratique.
(SDP)	:	Programmation semi-définie.
(PM)	:	Un problème mathématique primal.
(DM)	:	Le problème dual de (PM) .
(PL)	:	Le problème linéaire primal.
(DL)	:	Le problème dual de (PL) .
(PQ)	:	Le problème quadratique primal.
(DQ)	:	Le problème dual de (PQ) .
(\mathcal{P})	:	Le problème convexe primal.
(\mathcal{D})	:	Le problème dual de (\mathcal{P}) .
(Tc)	:	Trajectoire centrale.
<i>Algo</i>	:	Algorithme.
<i>sdp</i>	:	Semi définie positive.
<i>dp</i>	:	Définie positive.
$(PCCL)$:	La programmation non linéaire convexe à contraintes linéaires.
<i>CPI</i>	:	Condition de points intérieurs.
<i>K.K.T</i>	:	<i>Karush-Kuhn-Tucker</i> .
Faisable	:	Réalisable.
c.à.d	:	C'est à dire.
s à	:	sujet à.

Opérations

- $x \geq 0$: Les composantes de x , $x_i \geq 0$ pour tout i ;
- $x > 0$: Les composantes de x , $x_i > 0$ pour tout i ;
- $X \succeq 0$: X est une matrice symétrique semi-définie positive;
- $X \succ 0$: X est une matrice symétrique définie positive;
- xz = $(x_1z_1, \dots, x_nz_n)^T$ (produit d'Hadamard);
- $\frac{x}{z}$ = $(\frac{x_1}{z_1}, \dots, \frac{x_n}{z_n})^T$ ($z \neq 0$);
- \sqrt{x} = $(\sqrt{x_1}, \dots, \sqrt{x_n})^T$ ($x > 0$);
- x^{-1} = $(\frac{1}{x_1}, \dots, \frac{1}{x_n})^T$ ($x \neq 0$);
- $\langle x, z \rangle$ = $x^T z = e^T(xz) = \sum_{k=1}^n x_k z_k$, (le produit scalaire de deux vecteurs);
- $\|x\|_\infty$ = $\max_{1 \leq i \leq n} |x_i|$, (norme infinie);
- $\|x\|_2$ = $\sqrt{\sum_{i=1}^n x_i^2}$, (norme euclidienne).

Vecteurs

Les vecteurs sont désignés par des lettres minuscules. Si x est un vecteur de \mathbb{R}^n , on désigne par :

- 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.

Des vecteurs sont libres c.à.d. sont linéairement indépendants.

Matrices

Les matrices sont désignées par des lettres majuscules.

Si A est une matrice, on désigne par :

- 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 la matrice 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;
- X^{-2} = $\text{diag}(x^{-2})$ avec $X_{ii}^{-2} = \frac{1}{x_i^2}$ ($x > 0$).

$A \in \mathbb{R}^{n \times n}$, est Symétrique : si $A = A^T$, c.à.d. $a_{i,j} = a_{j,i}$ pour tout $i = 1 \dots n$, est $j = 1 \dots n$;

$H \in \mathbb{R}^{n \times n}$, est dite (**sdp**) : si pour tout vecteur $x \in \mathbb{R}^n$, $x^T H x \geq 0$, et notée $H \succeq 0$;

$H \in \mathbb{R}^{n \times n}$, est dite (**dp**) : si pour tout vecteur $x \in \mathbb{R}^n$, $x^T H x > 0$, et notée $H \succ 0$.

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 ;
- $\mathcal{F}(\mathcal{P})$ = $\{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$,
l'ensemble des solutions réalisables de (\mathcal{P}) ;
- $ri\mathcal{F}(\mathcal{P})$ = $\{x \in \mathbb{R}^n : Ax = b, x > 0\}$, l'intérieur relatif de $\mathcal{F}(\mathcal{P})$ et
l'ensemble des solutions strictement réalisables de (\mathcal{P}) ;
- $\mathcal{F}(\mathcal{D})$ = $\{(y, z) \in \mathbb{R}^{m \times n} : A^T y - \nabla f(x) + z = 0, x \geq 0, z \geq 0\}$,
l'ensemble des solutions réalisables de (\mathcal{D}) ;
- $ri\mathcal{F}(\mathcal{D})$ = $\{(y, z) \in \mathbb{R}^{m \times n} : A^T y - \nabla f(x) + z = 0, x > 0, z > 0\}$,
l'intérieur relatif de $\mathcal{F}(\mathcal{D})$ et l'ensemble des solutions
strictement réalisables de (\mathcal{D}) .
- $\bar{\mathbb{R}}$ = $[-\infty, +\infty] = \mathbb{R} \cup \{\pm\infty\}$.

Fonctions

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

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

$$\frac{\partial f}{\partial x_i}(x) : \text{ les dérivées partielles au point } x;$$

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

$$\mathcal{C}^1(\mathcal{F}) : \text{ Espace vectoriel des fonctions continues et à dérivée continue sur } \mathcal{F};$$

$$\mathcal{C}^q(\mathcal{F}) : \text{ Espace vectoriel des fonctions } q \text{ fois continûment différentiables sur } \mathcal{F}.$$

Chapter 1

Analyse convexe et programmation mathématique

Afin de rendre facile la compréhension de cette thèse, nous avons besoin de rappeler quelques notions et propriétés d'analyse convexe, et de la programmation mathématique qui seront utiles par la suite.

1.1 Éléments d'analyse convexe

1.1.1 Définitions

Soit $f : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$

Epigraphe de f :

$$\text{epi}(f) = \{(x, \alpha) \in \mathbb{R}^{n+1} \mid f(x) \leq \alpha\}.$$

Domaine effectif de f :

$$\text{ed}(f) = \{x \in \mathbb{R}^n \mid f(x) < +\infty\}.$$

Ensembles de niveau $\alpha \in \mathbb{R}$:

- Inférieur (large) :

$$\Gamma_\alpha = \left\{ x \in \mathbb{R}^n : \tilde{f}(x) \leq \alpha \right\}.$$

- Supérieur (large) :

$$\Gamma_\alpha = \left\{ x \in \mathbb{R}^n : \tilde{f}(x) \geq \alpha \right\}.$$

1.1.2 Ensemble affine

Définition 1.1.1 ([37]) *Un sous-ensemble \mathcal{F} de \mathbb{R}^n est dit affine si :*

$$\forall x, y \in \mathcal{F}, \forall \lambda \in \mathbb{R} : (1 - \lambda)x + \lambda y \in \mathcal{F}.$$

Définition 1.1.2 ([37]) *Soit \mathcal{F} une partie de \mathbb{R}^n , alors il existe une partie affine unique $C \subseteq \mathbb{R}^n$ contenant \mathcal{F} appelée enveloppe affine de \mathcal{F} et notée $\text{aff}(\mathcal{F})$, c'est la plus petite partie affine de \mathbb{R}^n contenant \mathcal{F} ,*

$$\text{aff}(\mathcal{F}) = C = \bigcap \{ C_{\mathcal{F}} : C_{\mathcal{F}} \supset \mathcal{F} \text{ et } C_{\mathcal{F}} \text{ affine} \}.$$

1.1.3 Ensemble convexe

Définition 1.1.3 ([37]) *Un sous-ensemble \mathcal{F} de \mathbb{R}^n est dit convexe si :*

$$(1 - \lambda)x + \lambda y \in \mathcal{F}, \forall x, y \in \mathcal{F} \text{ et } \forall \lambda \in [0, 1].$$

Définition 1.1.4 ([37]) *On désigne par $\mathcal{B}(0, 1)$, la boule unité euclidienne de \mathbb{R}^n :*

$$\mathcal{B}(0, 1) = \{ x \in \mathbb{R}^n : \|x\| \leq 1 \},$$

$\mathcal{B}(0, 1)$ est un convexe fermé et borné. $\forall a \in \mathbb{R}^n$, et pour tout $\epsilon > 0$, la boule de centre a

et de rayon ϵ s'écrit :

$$\begin{aligned}\mathcal{B}(a, \epsilon) &= a + \{x \in \mathbb{R}^n : \|x\| \leq \epsilon\} \\ &= a + \epsilon\mathcal{B}.\end{aligned}$$

Pour tout $\mathcal{F} \subseteq \mathbb{R}^n$, l'ensemble des x dont la distance à \mathcal{F} ne dépasse pas ϵ est :

$$\{x \in \mathbb{R}^n : \exists y \in \mathcal{F}, \|x - y\| \leq \epsilon\} = \cup\{y + \epsilon\mathcal{B} : y \in \mathcal{F}\} = \mathcal{F} + \epsilon\mathcal{B}.$$

La clôture de \mathcal{F} notée $\bar{\mathcal{F}}$ ou $cl(\mathcal{F})$, et son intérieur noté \mathcal{F}^0 ou $int(\mathcal{F})$, peuvent donc s'écrire sous les formes suivantes :

$$\begin{aligned}cl(\mathcal{F}) &= \cap\{\mathcal{F} + \epsilon\mathcal{B} : \epsilon > 0\}, \\ int(\mathcal{F}) &= \{x \in \mathcal{F} : \exists \epsilon > 0, (x + \epsilon\mathcal{B}) \subseteq \mathcal{F}\}.\end{aligned}$$

Définition 1.1.5 ([37]) L'intérieur relatif d'un sous-ensemble non vide \mathcal{F} de \mathbb{R}^n noté $ri(\mathcal{F})$,

$$ri(\mathcal{F}) = \{x \in aff(\mathcal{F}) : \exists \epsilon > 0, (x + \epsilon\mathcal{B}) \cap aff(\mathcal{F}) \subseteq \mathcal{F}\}.$$

Propriétés :

- 1) $int(\mathcal{F}) \subseteq ri(\mathcal{F}) \subseteq \mathcal{F} \subseteq cl(\mathcal{F})$.
- 2) \mathcal{F} relativement ouvert $\Leftrightarrow ri(\mathcal{F}) = \mathcal{F}$ (exemple : \mathcal{F} affine).

1.1.4 Fonction convexe

Définition 1.1.6 ([37]) Une fonction $f: \mathbb{R}^n \rightarrow \bar{\mathbb{R}}$ est dite convexe si :

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

Si l'inégalité précédente est stricte pour tout $x \neq y$ et $\lambda \in]0, 1[$, alors f est dite strictement convexe.

Définition 1.1.7 ([37]) f est dite *cœrcive* sur \mathcal{F} si $\lim_{\|x\| \rightarrow \infty} f(x) = +\infty$.

Remarque 1.1.1 Une fonction $f \in \mathcal{C}^2(\mathcal{F})$ est convexe \iff la matrice Hessienne

$$H(x) = \nabla^2 f(x)$$

est (**sdp**) pour tout $x \in \mathcal{F}$.

f est strictement convexe $\iff H$ est (**dp**) pour tout $x \in \mathcal{F}$ et $x \neq 0$, ([37]).

Remarque 1.1.2 Si f est une fonction quadratique :

$$f(x) = \frac{1}{2}x^T Qx + c^T x,$$

avec $Q \in \mathbb{R}^{n \times n}$: $Q \succ 0$ et $c \in \mathbb{R}^n$, alors f est strictement convexe.

Définition 1.1.8 ([37]) f est dite *propre* si $\text{ed}(f) \neq \emptyset$ et $f(x) > -\infty, \forall x \in \mathbb{R}^n$.

Définition 1.1.9 ([37]) $f : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$ est une fonction fermée dont l'épigraphe $\text{epi}(f)$ est fermé, ce qui est équivalent à sa semi-continuité inférieure. La convexité ne joue aucun rôle ici.

Définition 1.1.10 ([37]) $f : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$ est convexe sur $\text{epi}(f) \iff \text{epi}(f)$ est un ensemble convexe sur \mathbb{R}^{n+1} de plus $\text{ed}(f)$ et $\Gamma_\alpha(f)$ ($\forall \alpha \in \mathbb{R}$) sont des convexes de \mathbb{R}^n .

1.1.5 Cônes Convexes

Définition 1.1.11 $K \subseteq \mathbb{R}^n$ est un cône $\iff \mathbb{R}_+^* K \subseteq K$ c.à.d :

$$\forall \lambda > 0, \forall x \in K, \lambda x \in K.$$

Définition 1.1.12 Un cône K est un convexe $\iff K + K \subseteq K$.

Proposition 1 Le produit cartésien de cônes est un cône.

Proposition 2 *L'intersection quelconque de cônes est un cône.*

Cônes de récession ([37])

Définition 1.1.13 *Soit C un convexe non vide $\Rightarrow \mathcal{O}^+C = \{v \in \mathbb{R}^n \mid C + \lambda v \subseteq C, \forall \lambda > 0\}$ est un convexe non vide, ce dernier est appelé cône de récession de C .*

Proposition 3 *C un convexe non vide, fermé $\Rightarrow \mathcal{O}^+C(a) = \mathcal{O}^+C(b) \forall a, b \in C$.*

Pour toute autre notion d'analyse convexe qui n'aurait pas été précisée ci-dessus, on pourra se référer à ([37]).

1.2 Programmation Mathématique

Un programme mathématique est en général défini comme suit :

$$(PM) \quad \min_x f(x) \quad \text{s à : } \mathcal{F} = \{x \in \mathbb{R}^n : g_i(x) = 0, i = 1 \dots n, U_j(x) \leq 0, j = 1 \dots p\}.$$

La fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est appelée fonction objectif (ou économique) et \mathcal{F} est l'ensemble des contraintes ou bien l'ensemble des solutions réalisables.

Définition 4 *Une contrainte d'inégalité $U_j(x) \leq 0 \forall j$, est dite **saturée** (ou active) en $x^* \in \mathcal{F}$ si $U_j(x^*) = 0$.*

Remarque 1.2.1 *Une contrainte d'égalité $g_i(x) = 0$, est par définition **saturée** en tout point x de \mathcal{F} .*

1.2.1 Solutions optimales locales et globales

• Un point $x^* \in \mathcal{F}$ est une solution optimale locale pour PM , s'il existe un voisinage $\mathcal{V}(x^*)$ de x^* tel que :

$$f(x^*) \leq f(x), \text{ pour tout } x \in \mathcal{F} \cap \mathcal{V}(x^*),$$

et on note l'ensemble de ces solutions par $\text{loc min}_{\mathcal{F}} f(x)$.

- Un point $x^* \in \mathcal{F}$ est une solution optimale globale pour PM si :

$$f(x^*) \leq f(x), \text{ pour tout } x \in \mathcal{F},$$

et on le note par $\arg \min_{\mathcal{F}} f(x)$, l'ensemble des solutions optimales globales de PM et nous avons toujours :

$$\arg \min_{\mathcal{F}} f(x) \subset \text{loc min}_{\mathcal{F}} f(x).$$

Si PM est un problème convexe, les deux ensembles sont égaux.

1.2.2 Classification d'un programme mathématique

On classifie un programme mathématique à partir de deux propriétés fondamentales à savoir la convexité et la différentiabilité de la fonction objectif et les contraintes.

- PM est un problème différentiable si les fonctions f, g_i, U_j sont toutes différentiables.
- PM est un problème convexe si et seulement si f est convexe et l'ensemble \mathcal{F} des contraintes est convexe.

La convexité et la différentiabilité jouent un rôle très important dans la programmation mathématique.

1.2.3 Qualification des contraintes

La condition de qualification est satisfaite en tout $x^* \in \mathcal{F}$ dans les cas suivants :

- Les contraintes sont affines.
- Les gradients des contraintes saturées en x^* sont libres.
- \mathcal{F} est convexe et $\text{int}(\mathcal{F}) \neq \emptyset$ (**condition de Slater**).

Remarque 1.2.2 On dit que le point x^* est régulier si les contraintes sont qualifiées en x^* .

1.2.4 Résolution d'un programme mathématique

La solution complète de PM est traitée dans l'ordre des points suivants :

- L'existence (et éventuellement l'unicité) d'une solution optimale.
- Caractérisation de la solution.
- Élaboration d'algorithmes pour calculer numériquement cette solution.

Existence et unicité d'une solution optimale d'un programme mathématique

Théorème 1.2.1 (*Weierstrass*) *Si $\mathcal{F} \subset \mathbb{R}^n$ est compact (fermé et borné) et f est une fonction continue sur \mathcal{F} , alors PM admet au moins une solution optimale globale $x^* \in \mathcal{F}$.*

corollaire 1.2.1 *Si $\mathcal{F} \subset \mathbb{R}^n$ est non vide et fermé et si f est continue et coercive sur \mathcal{F} alors PM admet au moins une solution optimale.*

Proposition 5 *Si f est strictement convexe et \mathcal{F} est convexe alors si PM admet une solution optimale, elle est unique.*

Remarque 1.2.3 *La stricte convexité n'assure pas l'existence de la solution mais tout simplement l'unicité.*

Condition d'optimalité

Reprenons encore une fois le programme mathématique :

$$(PM) \quad \min_x f(x) \quad \text{s à} : \mathcal{F} = \{x \in \mathbb{R}^n : g_i(x) = 0, i = 1 \dots m, U_j(x) \leq 0, j = 1 \dots n\},$$

où f, U_j, g_i sont continûment différentiable. La théorie de $(K.K.T)$ permet d'écrire les conditions nécessaires d'optimalité pour tout problème d'optimisation avec contraintes, (Voir [42]).

Théorème 1.2.2 *Soit $x^* \in \mathcal{F}$ un point régulier alors :*

$$x^* \in \text{loc} \min_{\mathcal{F}} f(x) \implies \exists \lambda \in \mathbb{R}^m \text{ et } v \in \mathbb{R}_+^n,$$

(λ, v) des multiplicateurs de Lagrange tels que :

$$(K.K.T) \quad \begin{cases} \nabla f(x^*) + \langle \lambda, \nabla g(x^*) \rangle + \langle v, \nabla U(x^*) \rangle = 0, \\ \lambda^T g(x^*) = 0 \text{ (condition d'optimalité)}, \\ v^T U(x^*) = 0 \text{ (condition de complémentarité)}. \end{cases}$$

- Si PM est convexe et $x^* \in \mathcal{F}$ un point régulier alors :

$$x^* \in \arg \min_{\mathcal{F}} f(x) \Leftrightarrow x^* \text{ satisfait } (K.K.T).$$

- Si la condition de régularité n'est pas satisfaite, les conditions de $(K.K.T)$ ne s'appliquent pas : on peut trouver des solutions optimales ne vérifient pas les conditions, comme on peut trouver des points non optimaux satisfaisant ces conditions. En fait, l'existence des multiplicateurs dans ce cas n'est pas assurée.

1.3 Programmation convexe

Un problème convexe \mathcal{P} consiste à minimiser (ou maximiser) une fonction convexe sous contraintes linéaires. Il s'agit donc d'un programme mathématique de la forme :

$$(\mathcal{P}) \quad \min_x f(x) \text{ s à : } x \in \mathcal{F}.$$

Ici, \mathcal{F} est un polyèdre convexe fermé de la forme :

$$\mathcal{F} = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\} \Leftrightarrow \{x \in \mathbb{R}^n : g(x) = b - Ax = 0, U(x) = -x \leq 0\}.$$

$A \in \mathbb{R}^{m \times n}$ est une matrice de plein rang c.a.d $rg(A) = m < n$, $b \in \mathbb{R}^m$, \mathcal{F} l'ensemble des solutions réalisables et $f : \mathbb{R}^n \rightarrow \mathbb{R}$, est la fonction objectif.

- \mathcal{P} est un problème différentiable et nous avons en particulier la fonction f deux fois différentiable sur \mathbb{R}^n c.à.d son gradient et son hessien $\nabla f(x)$ et $\nabla^2 f(x)$ existent et de

plus on a :

$$\begin{aligned}\nabla g(x) &= -A^T, \\ \nabla U(x) &= -e.\end{aligned}$$

- \mathcal{P} est un problème convexe $\iff f$ et \mathcal{F} sont convexes.

1.3.1 Existence (unicité) d'une solution optimale

1► Si le polyèdre \mathcal{F} est borné (donc compact) donc d'après *Weirstrass*, \mathcal{P} admet au moins une solution optimale globale.

2► Si f est strictement convexe c.à.d. $\nabla^2 f$ est une matrice (dp) et si \mathcal{F} est non vide, alors \mathcal{P} possède une solution optimale unique, (voir [42]).

1.3.2 Conditions d'optimalité

Les contraintes de \mathcal{P} sont linéaires alors pour :

$$x^* \in \mathcal{F} = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\},$$

les contraintes sont qualifiées en x^* alors :

$$x^* \in \text{loc min } f(x) \implies \exists \lambda \in \mathbb{R}^m \text{ et } v \in \mathbb{R}_+^n$$

tels que :

$$(K.K.T) \quad \begin{cases} \nabla f(x^*) - A^T \lambda - v &= 0, \\ \lambda^T (b - Ax^*) &= 0, \\ v^T x^* &= 0. \end{cases}$$

Lorsque la fonction f est convexe alors les conditions de $(K.K.T)$ d'ordre un, sont nécessaires et suffisantes.

1.3.3 Le dual d'un programme convexe

Le dual d'un programme convexe au sens de *Wolfe* est un programme convexe important, (voir le monographe de Minoux [41]). En général, le dual d'un programme mathématique au sens de *Wolfe*, est aussi un programme mathématique de la forme suivante :

$$(D) \quad \begin{cases} \max_{(y,z)} \mathcal{L}(x, y, z) \text{ s à :} \\ y \in \mathbb{R}^m, z \in \mathbb{R}_+^n, z \geq 0, \\ \nabla_x \mathcal{L}(x, y, z) = 0, \end{cases}$$

où \mathcal{L} est dite la fonction de *Lagrange* définie par :

$$\mathcal{L}(x, y, z) = f(x) + y^T g(x) + z^T U(x).$$

Alors dans le cas de \mathcal{P} , \mathcal{L} est donnée par :

$$\mathcal{L}(x, y, z) = f(x) + y^T (b - Ax) - z^T x$$

et

$$\nabla_x \mathcal{L}(x, y, z) = \nabla f(x) - A^T y - z.$$

Donc, le dual de \mathcal{P} est donné par :

$$(D) \quad \begin{cases} \max_{(y,z)} [b^T y - x^T \nabla f(x) + f(x)] \text{ s à :} \\ A^T y + z - \nabla f(x) = 0, \quad x \geq 0, z \geq 0, y \in \mathbb{R}^m, \end{cases}$$

avec y est le vecteur de multiplicateur de *Lagrange* associé à la contrainte $(b - Ax)$, z est le vecteur de multiplicateur de *Lagrange* associé à la contrainte d'inégalité $(-x \leq 0)$.

1.4 Algorithme d'optimisation

Nous allons présenter un algorithme permettant de converger vers une solution optimale du problème PM . La plupart des algorithmes d'optimisation avec contraintes exploitent les conditions d'optimalité pour déterminer des minima locaux. Nous donnerons ici quelques définitions, (pour plus de détails voir [15]).

1.4.1 Description

Un algorithme est défini par une application $Algo$, de $\mathcal{F}(\mathcal{P})$ dans $\mathcal{F}(\mathcal{P})$, permettant la génération d'une suite d'éléments de $\mathcal{F}(\mathcal{P})$ par la formule :

$$\begin{cases} x_0 \in \mathcal{F}(\mathcal{P}) \text{ donné, } k = 0 & \text{Etape d'initialisation,} \\ x_{k+1} = Algo(x_k), k = k + 1 & \text{Itération.} \end{cases}$$

Si on remplace $\mathcal{F}(\mathcal{P})$ par son intérieur, en supposant que $int\mathcal{F}(\mathcal{P}) \neq \emptyset$, l'algorithme est dit un algorithme de points intérieurs. Définir un algorithme n'est autre que construire une suite (x_k) , $k \in \mathbb{N}$ de $\mathcal{F}(\mathcal{P})$ et réaliser une étude pour montrer sa convergence.

1.4.2 Convergence

Définition 1.4.1 *On dit que l'algorithme $Algo$ est convergent si la suite (x_k) $k \in \mathbb{N}$ engendrée par l'algorithme converge vers une limite x^* .*

Taux de convergence

Un critère de mesure de la vitesse (ou le taux) de convergence est l'évolution de l'erreur commise à chaque itération donnée par :

$$e_k = \|x_k - x^*\|.$$

Avant de donner les notations des convergences, nous donnons les définitions des notations asymptotiques suivantes :

Définition 1.4.2 (Notation O) Soient deux fonctions $f, g : \mathbb{N} \rightarrow \mathbb{R}^+$.

- On note :

$$f(n) = O(g(n)),$$

lorsqu'il existe des entiers c et n_0 tels que pour tout $n \geq n_0$,

$$f(n) \leq cg(n).$$

Intuitivement, cela signifie que la valeur de la fonction f est inférieure à celle de g à une constante multiplicative près, pour les instances (données) de tailles suffisamment grandes. De même on définit :

Définition 1.4.3 (Notations O, Ω, Θ) Soient deux fonctions $f, g : \mathbb{N} \rightarrow \mathbb{R}^+$.

- On note :

$$f(n) = O(g(n)),$$

lorsque pour tout réel c , il existe un entier n_0 tel que pour tout $n \geq n_0$,

$$f(n) \leq cg(n).$$

- On note :

$$f(n) = \Omega(g(n))$$

lorsqu'il existe des entiers c et n_0 tels que pour tout $n \geq n_0$,

$$cg(n) \leq f(n),$$

- On note :

$$f(n) = \Theta(g(n))$$

lorsque

$$f(n) = O(g(n)) \text{ et } f(n) = \Omega(g(n)).$$

Définition 1.4.4 Soit (x_k) $k \in \mathbb{N}$ une suite donnée par l'algorithme Algo et convergente vers x^* . La classification de la vitesse de convergence d'une suite est basée sur les notions de comparaison des fonctions au voisinage de $+\infty$.

En effet, si on suppose que l'erreur e_k ne s'annule pas, la vitesse de la convergence pourra être :

Linéaire : si l'erreur

$$\|e_k\| = \Omega(\|e_{k+1}\|) \text{ et } \frac{\|e_{k+1}\|}{\|e_k\|} < 1,$$

pour k assez grand. On dit aussi que l'erreur e_k décroît linéairement c.à.d :

$$\exists c \in [0, 1[, \exists k_0 \in \mathbb{N}, \forall k \geq k_0, e_{k+1} \leq ce_k.$$

Super-linéaire : si l'erreur

$$\|e_{k+1}\| = O(\|e_k\|),$$

où l'erreur décroît de la manière suivante : il existe $\{\alpha_k\}$ une suite positive qui converge vers 0 telle que :

$$e_{k+1} \leq \alpha_k e_k.$$

D'ordre γ avec $\gamma > 1$, si :

$$\|e_{k+1}\| = O(\|e_k\|^\gamma) \text{ et } \frac{\|e_{k+1}\|}{\|e_k\|} < 1,$$

pour k assez grand, où l'erreur décroît de la manière suivante :

$$\exists c \in [0; 1[, \exists k_0 \in \mathbb{N}, \forall k \geq k_0; e_{k+1} \leq c(e_k)^\gamma.$$

Dans le cas $\gamma = 2$, la convergence est dite **quadratique**, ([19]).

Définition 1.4.5 Si nous disons le coût d'un algorithme est $O(f(L))$, nous signifions que pour L suffisamment grand, le nombre d'opérations arithmétiques est borné par la quantité $cf(L)$ c.à.d. $\leq cf(L)$ où c est une constante positive, L est la taille de la mémoire nécessaire pour stocker les données, et f est une fonction quelconque. Autrement l'algorithme Algo est de complexité polynômiale c.à.d résoudre le problème en un nombre d'opérations qui doit être borné par un polynôme en fonction de la taille du problème.

1.5 Méthode de Newton-Raphson pour un système non linéaire

Parmi les méthodes les plus populaires appliquées pour la résolution du système d'équations non linéaire (K.K.T), est la méthode de Newton, dans ce qui suit nous décrivons son principe. Soit $F : \mathbb{R}^n \mapsto \mathbb{R}^n$ une fonction non linéaire différentiable telle que :

$$F(x) = (f_1(x), f_2(x), \dots, f_n(x))^T.$$

Cette méthode est une procédure itérative qui a pour objectif de trouver un point $x \in \mathbb{R}^n$ telle que $F(x) = 0$. Si J est la matrice Jacobienne de F , et à l'itération k on écrit :

$$F(x^k + \Delta x^k) \approx F(x^k) + J(x^k)\Delta x^k,$$

où :

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

et la direction de Newton Δx^k , est choisie de telle façon que cette approximation linéaire est égale à zéro. On pose donc

$$x^{k+1} = x^k + \Delta x^k,$$

avec

$$\Delta x^k = -J(x^k)^{-1}F(x^k).$$

La convergence vers une solution est garantie à partir du moment où le point initial x^0 se trouve dans un voisinage suffisamment proche d'un des zéros de F . Le choix de la méthode de Newton est très important au point de vue des méthodes des points intérieurs à cause de son efficacité numérique, (voir [27]).

1.6 Méthodes de résolution d'un programme mathématique

On peut classer les méthodes de résolution d'un programme mathématique en trois catégories :

1.6.1 Méthodes de type gradient

Gradient conjugué

Cette méthode a été proposée par **Hestenes** (1952) pour résoudre un système linéaire à matrice définie positive, puis généralisée par **Fletcher** et **Reeves** (1964) pour résoudre des problèmes d'optimisation non linéaires, elle est connue par son efficacité pour minimiser une fonction quadratique sans contraintes, ([18]).

Gradient projeté

Cette méthode a été proposée par **Ros** en 1960. Le principe de cette méthode est de projeter à chaque itération le gradient sur la frontière du domaine réalisable. Il faut signaler que cette méthode est conçue pour un programme plus général de la forme :

$$\min_x f(x) \text{ s à : } Ax = b, x \geq 0,$$

où f est différentiable non nécessairement convexe. (Pour plus de détails le lecteur consultera le monographe [18]).

1.6.2 Méthodes simpliciales

Parmi les méthodes simpliciales, on cite celle de gradient réduit due à *Wolfe*. C'est une extension directe de la méthode du simplexe, appliquée à la programmation quadratique. De ce fait elle présente les mêmes inconvénients à savoir cyclage et complexité exponentielle.

1.6.3 Méthodes de points intérieurs

Les méthodes de points intérieurs forment une classe d'algorithmes qui permettent de résoudre des problèmes d'optimisation mathématiques. Elles ont l'intérêt d'être polynomiales lorsqu'on les applique aux problèmes d'optimisation linéaire, quadratique convexe, semi-défini et plus généralement aux problèmes d'optimisation convexe. Ces algorithmes sont des extensions des méthodes développées pour la programmation linéaire (affines, projectives et de trajectoire centrale). Les méthodes de points intérieurs se répartissent en plusieurs familles :

Méthodes Affines (optimisation sur des ellipsoïdes)

La méthode de l'ellipsoïde, est une méthode itérative utilisée pour minimiser des fonctions convexes. Cette méthode est connue comme étant le premier algorithme de complexité polynomiale découvert pour résoudre les problèmes d'optimisation linéaire. L'algorithme construit une suite d'ellipsoïdes de plus en plus petits, qui contiennent le minimum.

Arkadi Nemirovski, David B. Yudin, Shor développent en 1972, la méthode des ellipsoïdes pour des problèmes d'optimisation (non linéaire) convexe. Il s'agit pratiquement de l'algorithme de **Karmarkar** sans fonction potentielle et sans transformation projective, on utilise une transformation affine et on remplace la contrainte de non-négativité

par un ellipsoïde qui contient le nouveau itéré. L'algorithme est d'une structure simple, malheureusement, il n'est pas facile de démontrer la polynômialité, ([53]).

Méthodes de réduction du potentiel

La fonction potentiel joue un grand rôle dans le développement des méthodes de points intérieurs. L'algorithme de **Karmarkar** ([35]) appliqué au programme linéaire sous forme standard utilise une fonction potentielle de la forme :

$$\rho \log(c^T x - Z) - \sum_{i=1}^n \log(x_i),$$

où $\rho = n+1$ et Z est une borne inférieure de la valeur optimale de l'objectif. **Karmarkar** prouve la convergence et la polynômialité de son algorithme par monter que cette fonction est réduite à chaque itération par au moins une constante. Depuis 1987, les chercheurs introduisent des fonctions du potentiel de type primal-dual, parmi les quelles, celle de **Tanabe, Todd** et **Ye** définie par :

$$\Phi_\rho(x; z) = \rho \log(x^T z) - \sum_{i=1}^n \log(x_i z_i),$$

Pour $\rho > n$. Cette fonction a joué un rôle très important dans le développement des algorithmes de réduction du potentiel après 1988. Les algorithmes correspondants à ces méthodes possèdent une complexité polynômiale, ils nécessitent $O(\sqrt{n} |\log \varepsilon|)$ itérations pour réduire le saut de dualité ($x^T z \leq \varepsilon$, ε est une précision donnée). (Les livres de Roos et al [49], Ye [54], et Wright [53] sont importants pour le développement des méthodes modernes de points intérieurs).

Méthodes de trajectoire centrale (TC)

Ces méthodes sont le fruit direct d'une grande partie des études acharnées menées par plusieurs chercheurs vers la fin des années 80, et pleinement développées au début des

années 90. Elles possèdent les propriétés théoriques les plus esthétiques, complexité polynomiale et une convergence super linéaire et des fois quadratique . Ces qualités de confort placent cette classe de méthodes au centre de l'intérêt primordial des chercheurs, pour résoudre effectivement des programmes mathématiques avec contraintes. Les algorithmes de TC restreintes les itérés à un voisinage du chemin central, ce dernier est un arc continu de points strictement réalisables. Décrivons maintenant ces méthodes de points intérieurs de TC pour PCCL.

Chapter 2

Méthodes de points intérieurs de type de TC pour (PCCL)

Dans ce chapitre, on s'intéresse à l'étude de la trajectoire centrale (suivi de chemin) pour un problème d'optimisation convexe sous contraintes linéaires (PCCL).

2.1 Existence et unicité de la trajectoire centrale (suivi de chemin) pour PCCL

Dans cette partie, nous considérons pour une deuxième fois le programme convexe primal sous la forme standard suivante :

$$(\mathcal{P}) \quad \min_x f(x) \quad \text{s à} : Ax = b, x \geq 0,$$

et son dual :

$$(\mathcal{D}) \quad \max_{(y, z)} [b^T y - x^T \nabla f(x) + f(x)] \quad \text{s à} : A^T y + z - \nabla f(x) = 0, x \geq 0, z \geq 0,$$

où $f : \mathbb{R}^n \mapsto \overline{\mathbb{R}}$ est une fonction deux fois continument différentiable, et A est une matrice de type (m, n) , $b \in \mathbb{R}^m$, $x \in \mathbb{R}^n$, $z \in \mathbb{R}^n$ et $y \in \mathbb{R}^m$. Le domaine effectif de f sera noté par :

$$\mathbf{ed}(f) = \{x \in \mathbb{R}^n \mid f(x) < +\infty\}.$$

On suppose que les problèmes \mathcal{P} et \mathcal{D} vérifient les hypothèses suivantes :

Hypothèse 1. Le domaine effectif $\mathbf{ed}(f)$ est un ensemble ouvert.

Hypothèse 2. La matrice A est de plein rang, c.à.d $rg(A) = m$, ($m < n$).

Hypothèse 3. (CPI) il existe (x^0, y^0, z^0) tel que x^0 dans $\mathbf{ed}(f)$ c.à.d :

$$Ax^0 = b, A^T y^0 - \nabla f(x^0) + z^0 = 0, x^0 > 0, z^0 > 0 \text{ et } f(x^0) < +\infty$$

Hypothèse 4. f est une fonction au moins deux fois continûment différentiable sur $\mathbf{ed}(f)$ et convexe donc la matrice Hessienne $\nabla^2 f(x)$ est (*sdp*). Notez que puisque f est une fonction convexe fermée propre, à partir de ces hypothèses il s'ensuit que f tends vers $+\infty$ sur la frontière de $\mathbf{ed}(f)$, ([29]).

Remarque 2.1.1 L'hypothèse 3, est équivalente à : $ri\mathcal{F}(\mathcal{P}) \times ri\mathcal{F}(\mathcal{D}) \neq \emptyset$ où :

$ri\mathcal{F}(\mathcal{P}) \times ri\mathcal{F}(\mathcal{D}) = \{(x, y, z) \in \mathbb{R}^{2n+m} : Ax = b, A^T y + z - \nabla f(x) = 0, x > 0, z > 0, f(x^0) < +\infty\}$, désigne l'ensemble des points strictement réalisables des deux problèmes \mathcal{P} et \mathcal{D} .

Sous nos hypothèses, et car les contraintes sont linéaires donc sont qualifiées alors les conditions d'optimalité du premier ordre dites conditions de (*K.K.T*) sont nécessaires et suffisantes pour les deux problèmes convexes \mathcal{P} et \mathcal{D} . Alors résoudre les problèmes \mathcal{P} et \mathcal{D} revient donc à résoudre le système d'équations non linéaire suivant :

$$(K.K.T) \quad \begin{cases} A^T y + z - \nabla f(x) & = 0, z \geq 0, \\ Ax & = b, x \geq 0, \\ xz & = 0. \end{cases}$$

2.1.1 Le problème perturbé

La méthode de barrière logarithmique appliquée à \mathcal{P} génère une famille de problèmes perturbés définis comme suit :

$$(\mathcal{P})_\mu \quad \min_x B_\mu(x) \text{ s à : } Ax = b, x > 0 \quad (2.1)$$

avec B_μ est donnée par :

$$B_\mu(x) = f(x) - \mu \sum_{i=1}^n \ln x_i,$$

où $\mu > 0$, est un scalaire positif dit paramètre barrière associé au problème \mathcal{P}_μ .

Propriétés de la fonction B_μ

- Pour tout $\mu > 0$, la fonction B_μ est continûment différentiable et strictement convexe car la matrice *Hessienne* :

$$\nabla^2 B_\mu(x) = \nabla^2 f(x) + \mu X^{-2},$$

est *(dp)* pour tout $x > 0$, car $X^{-2} = \text{diag}(x^{-2})$ avec $X_{ii}^{-2} = \frac{1}{x_i^2}$, $x_i > 0$.

Sous nos hypothèses le problème (2.1), admet les propriétés suivantes :

- Pour tout $\mu > 0$ (fixé), le problème (2.1) admet une solution unique notée $x(\mu)$, appelée un point central du problème \mathcal{P}_μ .
- La fonction $\mu \mapsto x(\mu)$ définit la trajectoire centrale de \mathcal{P}_μ .
- Quand $\mu \rightarrow 0$, $x(\mu) \rightarrow x^*$ (où x^* est une solution optimale du problème \mathcal{P}).
- La solution $x(\mu)$ est définie d'une façon unique par les conditions d'optimalité de $(K.K.T)_\mu$.

2.1.2 Les conditions d'optimalité de $(K.K.T)_\mu$

Le système suivant :

$$\begin{cases} A^T y - \nabla f(x) + \mu X^{-1}e = 0, \\ Ax = b, \\ x > 0, \end{cases} \quad (2.2)$$

désigne les conditions d'optimalité de $(K.K.T)_\mu$. On pose :

$$z = \mu X^{-1}e,$$

alors, le système (2.2) devient :

$$\begin{cases} A^T y + z - \nabla f(x) = 0, \quad z > 0, \\ Ax = b, \quad x > 0, \\ xz = \mu e. \end{cases} \quad (2.3)$$

Le système (2.3), désigne aussi les conditions d'optimalité du problème perturbé dual \mathcal{D}_μ suivant :

$$(\mathcal{D})_\mu \quad \begin{cases} \max_{(y,z)} [b^T y - x^T \nabla f(x) + f(x) + \mu \sum_{i=1}^n \ln z_i], \\ \text{s à : } A^T y + z - \nabla f(x) = 0, \quad z > 0. \end{cases}$$

La solution du système (2.3) est appelée le point central correspondant à $\mu > 0$. Il est noté par $(x(\mu), y(\mu), z(\mu))$. Pour chaque $\mu > 0$, la trajectoire centrale est bien définie et dépend de l'existence et de l'unicité de la solution mentionnée dans ce qui précède.

Remarque 2.1.2

- 1- Le système (2.3), admet une solution unique $(x(\mu), y(\mu), z(\mu))$ pour tout $\mu > 0$.
- 2- L'ensemble suivant :

$$\mathcal{C} = \{(x(\mu), y(\mu), z(\mu)) : \mu > 0\},$$

définit la trajectoire centrale des deux problèmes \mathcal{P}_μ et \mathcal{D}_μ .

2.1.3 Le chemin central est bien défini

Afin d'étudier que la trajectoire centrale est bien définie, nous définissons la fonction $\Phi_\mu(x)$ par :

$$\Phi_\mu(x) = \begin{cases} f(x) - \mu \sum_{i=1}^n \ln x_i, & \text{si } Ax = b, x > 0 \text{ et } x \in \mathbf{ed}(f), \\ +\infty & \text{autrement,} \end{cases}$$

et

$$\tilde{f}(x) = \begin{cases} f(x), & \text{si } Ax = b, x > 0 \text{ et } x \in \mathbf{ed}(f), \\ +\infty & \text{autrement.} \end{cases}$$

Il est facile de voir que Φ_μ et \tilde{f} sont des fonctions propres convexes et fermées et de plus Φ_μ est strictement convexe dans son domaine effectif. À partir de maintenant $\Gamma(\alpha, \mu)$ et Γ_α désignent les ensembles du niveau de Φ_μ et \tilde{f} respectivement, correspondant à $\alpha \in \mathbb{R}$, c.à.d :

$$\Gamma(\alpha, \mu) = \{x \in \mathbb{R}^n \mid \Phi_\mu(x) \leq \alpha\},$$

et

$$\Gamma_\alpha = \{x \in \mathbb{R}^n \mid \tilde{f}(x) \leq \alpha\}.$$

$\Gamma(\alpha, \mu)$ et Γ_α sont des sous-ensembles fermés (convexes) de \mathbb{R}^n car Φ_μ et \tilde{f} sont des fonctions fermées (convexes). Nous rappelons que le cône de récession d'un ensemble convexe $C \subset \mathbb{R}^n$, est donné par :

$$\mathcal{O}^+C = \{v \in \mathbb{R}^n : C + \lambda v \subseteq C, \forall \lambda > 0\}.$$

Nous rappelons aussi quelques résultats très connus de l'analyse convexe qui seront utiles par la suite.

Lemme 2.1.1 ([29]) *Soit C un convexe fermé non vide de \mathbb{R}^n , alors C est borné si et seulement si son cône de récession $\mathcal{O}^+C = \{0\}$.*

Preuve. Voir Proposition 3.5, ([37]). □

Lemme 2.1.2 ([29]) *Les ensembles de niveau non vides d'une fonction convexe propre fermée sont soit tous bornés.*

Preuve. Voir Corollaire 8.7.1, ([47]). □

corollaire 2.1.1 ([29]) *Une fonction convexe propre fermée à un ensemble de niveau borné non vide définit si et seulement si l'ensemble de ses minimums est non vide et borné .*

2.1.4 Le Théorème principal de la trajectoire centrale

Maintenant, nous prouvons le théorème principal qui donne les conditions nécessaires et suffisantes, pour que la trajectoire centrale associée au problème \mathcal{P} est bien définie, ([29]).

Théorème 2.1.1 *Les conditions suivantes sont équivalentes :*

C1) *L'ensemble des solutions réalisables de \mathcal{P} , $\mathcal{F}_{(\mathcal{P})}$ n'est pas vide et borné.*

C2) *Le chemin central $\mathcal{C} = \{(x(\mu), y(\mu), z(\mu)) : \mu > 0\}$, est bien défini.*

C3) *Pour certains $\mu_0 > 0$, le point central $(x(\mu_0), y(\mu_0), z(\mu_0))$ est bien défini.*

C4) *Il existe un point intérieur réalisable dual $(\bar{y}, \bar{z}) \in \mathbb{R}^{m \times n}$:*

$$A^T \bar{y} - \nabla f(\bar{x}) + \bar{z} = 0, \quad \bar{x} \geq 0, \quad \bar{z} \geq 0.$$

Preuve. Supposons que la condition C1 est vérifiée, et nous montrons que la condition C2 est vraie. On à $\mu > 0$ en utilisant x^0 comme dans **Hypothèse 3**, on pose $\hat{\alpha} = \Phi_\mu(x^0)$. Supposons que :

$$\mathcal{O}^+ \Gamma(\hat{\alpha}, \mu) = \{0\},$$

soit $v \in \mathbb{R}^n$ un élément de $\mathcal{O}^+ \Gamma(\hat{\alpha}, \mu)$, donc pour chaque $x \in \mathcal{O}^+ \Gamma(\hat{\alpha}, \mu)$ et $t \geq 0$ on a :

$$\Phi_\mu(x + tv) \leq \hat{\alpha} < +\infty,$$

alors, en vue de la définition de Φ_μ , on a :

$$\begin{aligned} A(x + tv) &= b, \\ x + tv &\geq 0 \end{aligned} \tag{2.4}$$

et

$$\begin{aligned} \hat{\alpha} &\geq \Phi_\mu(x + tv) = f(x + tv) - \mu \sum_{i=1}^n \ln(x_i + tv_i) \\ &\geq f(x) + t \nabla f(x)^T v - \mu \sum_{i=1}^n \ln(x_i + tv_i), \end{aligned}$$

où la seconde inégalité est satisfaite puisque f est convexe, donc

$$\nabla f(x)^T v \leq 0, \text{ pour tout } x \in \Gamma(\hat{\alpha}, \mu)$$

est vraie, parce que $v_i \geq 0$ par la faisabilité de $x + tv$, et le fait que le logarithme croît plus lentement qu'une fonction linéaire de t . Ainsi puisque $x_0 + tv \in \Gamma(\hat{\alpha}, \mu)$ pour tout $t \geq 0$, nous avons :

$$\nabla f(x_0 + tv)^T v \leq 0, \text{ pour tout } t \geq 0.$$

Par conséquent $f(x_0 + tv)$ est une fonction non croissante de $t \geq 0$, de sorte que :

$$f(x_0 + tv) \leq f(x_0) \text{ pour tout } t \geq 0. \tag{2.5}$$

On a $x_0 \in \Gamma(\hat{\alpha}, \mu)$. Donc, en utilisant **(2.4)** avec $x = x_0$, il s'ensuit que $x_0 + tv$ est réalisable pour tout $t \geq 0$. De la faisabilité de $\{x_0 + tv \mid t \geq 0\}$ et **(2.3)**, nous concluons que

$$\{x_0 + tv \mid t \geq 0\} \subseteq \Gamma_{\hat{\omega}} \text{ où } \hat{\omega} = f(x_0).$$

D'autre part, par la condition $C1$ et Lemme **2.1.2**, l'ensemble non vide $\Gamma_{\hat{\omega}}$ est borné, donc $v = 0$ et ainsi

$$O^+\Gamma(\hat{\alpha}, \mu) = \{0\}.$$

En vue du Lemme **2.1.2**, il s'ensuit que $\Gamma(\hat{\alpha}, \mu)$ est un (non vide) borné. Donc Φ_{μ} atteint son minimum $x(\mu)$, qui est unique en raison de la stricte convexe de Φ_{μ} dans son domaine effectif. Notez que par **Hypothèse 2**, la matrice AA^T est régulière. Par conséquent, en prenant :

$$z(\mu) = \mu X^{-1}(\mu)e,$$

et

$$y(\mu) = (AA^T)^{-1} A(\nabla f(x(\mu)) - z(\mu)).$$

Alors, on déduit que $(x(\mu), y(\mu), z(\mu))$ est une solution unique du système (2.2), donc, la condition $C2$ est vérifiée. La condition $C3$ est une conséquence évidente de la condition $C2$. Si la condition $C3$ satisfait, alors $(x(\mu_0), y(\mu_0), z(\mu_0))$ satisfait le système (2.2) pour $\mu = \mu_0$. Ainsi, la condition $C4$ est vraie avec :

$$(\bar{x}, \bar{y}, \bar{z}) = (x(\mu_0), y(\mu_0), z(\mu_0)).$$

Supposons maintenant que la condition $C4$ est vérifiée et nous vérifions la condition $C1$, soit $x \in \mathbb{R}^n$ tel que :

$$Ax = b, \quad x \geq 0. \tag{2.6}$$

Puis en utilisant le fait que la fonction f est convexe, **(2.6)** et la condition $C4$ nous obtenons :

$$\begin{aligned} f(x) &\geq f(\bar{x}) + \nabla f(\bar{x})^T(x - \bar{x}) \\ &= f(\bar{x}) + (A^T\bar{y} + \bar{z})^T(x - \bar{x}) \\ &= f(\bar{x}) - (A^T\bar{y} + \bar{z})^T\bar{x} + b^T\bar{y} + \bar{z}^T x. \end{aligned} \tag{2.7}$$

De la condition $C4$, il existe un $\rho \in \mathbb{R}$ tel que :

$$\bar{z}_i \geq \rho > 0, \text{ pour } i = 1, \dots, n. \quad (2.8)$$

En utilisant (2.7) et (2.8) nous obtenons :

$$\tilde{f}(x) = f(x) \geq W + \rho \|x\|_1, \quad (2.9)$$

où

$$W = f(\bar{x}) - (A^T \bar{y} + \bar{z})^T \bar{x} + b^T \bar{y} \text{ et } \|x\|_1 = \sum_{i=1}^n x_i.$$

Par conséquent, de (2.9) et Lemme 2.1.2 nous concluons que tous les ensembles de niveau de \tilde{f} sont bornée. Puisque \tilde{f} est une fonction convexe propre fermée, du Corollaire 2.1.1 il s'ensuit que l'ensemble des solutions strictement réalisables de \mathcal{P} est non vide et borné. Dans ce qui suit le principe des algorithmes primal-dual de suivi de trajectoire. \square

2.2 Principe des méthodes de trajectoire centrale

Le principe de ces méthodes est basé sur l'idée d'approximer la trajectoire centrale c.à.d. résoudre le système (2.3) approximativement en utilisant un pas de Newton. D'après, la dernière équation du système (2.3), on déduit que : $x^T z = n\mu$,

ou bien :

$$x^T z = x^T \nabla f(x) - b^T y.$$

La quantité $x^T \nabla f(x) - b^T y$ désigne le saut de dualité des deux problèmes \mathcal{P} and \mathcal{D} , donc il suffit de trouver approximativement des points sur la trajectoire centrale \mathcal{C} quand $\mu \rightarrow 0$. Dans ce qui suit la forme générique d'algorithme primal-dual de type suivi de trajectoire.

Forme générique des algorithmes primal-dual

Données : $\epsilon > 0$, $\mu^0 > 0$ (des paramètres fixés);

$k := 0$;

Répéter

Trouver approximativement $(x(\mu^k), y(\mu^k), z(\mu^k))$ par la méthode de Newton;

Choisir $\mu^{k+1} < \mu^k$;

$k := k + 1$;

Jusqu'à : $\mu^k < \epsilon$.

Remarque 2.2.1

1- La résolution de (2.3) est approximative puisqu'il est impossible de calculer le point central $(x(\mu^k), y(\mu^k), z(\mu^k))$, d'une façon exacte c.à.d. qui vérifie l'équation :

$$\frac{x(\mu^k)z(\mu^k)}{\mu^k} = e,$$

on se contente d'une solution approchée vérifiant un certain critère de proximité.

2- Cet algorithme dépend de la mise à jour du paramètre μ (choix de μ^{k+1} à chaque itération k).

2.2.1 Concept de proximité ou de centralisation

La proximité est une distance qui mesure le lointain d'un point strictement réalisable (x, y, z) par rapport à la trajectoire centrale et on la note par :

$$\delta(xz; \mu).$$

Cette quantité est donnée en différentes formes, (voir [49], [53], [54]).

2.2.2 Mise à jour du paramètre barrière μ

La façon traditionnelle de mettre à jour le paramètre est de le multiplier par un facteur $\sigma \in [0, 1]$ c.à.d. $\mu^{k+1} = \sigma\mu^k$. Ce paramètre σ peut dépendre de n . La manière de varier ce paramètre à chaque itération entraîne :

a) **Les méthodes primal-dual de suivi de trajectoire à petit pas** (si σ dépend de n c.à.d. $\sigma = \Theta(n)$).

b) **Les méthodes primal-dual de suivi de trajectoire à grand pas** (si σ est un constant indépendant de n la dimension du problème c.à.d. $\sigma = \Theta(1)$).

2.2.3 Itération de Newton

Soit $(x, y, z) \in ri\mathcal{F}(\mathcal{P}) \times ri\mathcal{F}(\mathcal{D})$, $\mu > 0$ et $\bar{\mu} = \sigma\mu$. En suivant ce qui précède, notre but est de se rapprocher du point central $(x(\mu), y(\mu), z(\mu))$ pour $\sigma \in [0, 1]$, on veut plus précisément trouver $(x^+, y^+, z^+) = (x + \Delta x, y + \Delta y, z + \Delta z)$ tel que (x^+, y^+, z^+) soit strictement réalisable. La méthode de **Newton** résout approximativement le système (2.3) par le système suivant :

$$\begin{cases} A^T y^+ + z^+ - \nabla f(x^+) & = 0, \\ Ax^+ & = b, \\ x^+ z^+ & = \bar{\mu}e. \end{cases}$$

En négligeant le terme quadratique $\Delta x \Delta z$, on aura le système d'équations linéaires suivant :

$$\begin{cases} \nabla^2 f(x) \Delta x - A^T \Delta y - \Delta z & = 0, \\ A \Delta x & = 0, \\ X \Delta z + Z \Delta x & = \bar{\mu}e - Xz. \end{cases} \quad (2.10)$$

Le nouvel itéré est alors défini par :

$$(x^+, y^+, z^+) = (x, y, z) + (\Delta x, \Delta y, \Delta z).$$

2.2.4 Pas de déplacement

Après la résolution du système **(2.10)**, on obtient les directions $(\Delta x, \Delta y, \Delta z)$, et le nouvel itéré de Newton est calculé en prenant $\alpha = 1$, mais ce choix du pas de déplacement ne donne pas toujours la stricte faisabilité. Notre but est de préserver la stricte faisabilité et donc on doit trouver un α tel que le nouvel itéré soit strictement réalisable c.à.d.

$$(x^{k+1}, y^{k+1}, z^{k+1}) = (x^k, y^k, z^k) + \alpha_k(\Delta x^k, \Delta y^k, \Delta z^k)$$

tels que $x_{k+1} > 0$ et $z_{k+1} > 0$.

Remarque 2.2.2 Rappelons que le nouvel itéré est défini par :

$$x^+ = x + \alpha\Delta x, y^+ = y + \alpha\Delta y, z^+ = z + \alpha\Delta z$$

pour tout $\alpha \in]0, 1]$ et d'après les systèmes **(2.3)** et **(2.10)** on a :

$$\begin{aligned} Ax^+ &= b, \\ A^T y^+ + z^+ - \nabla^2 f(x^+) &= 0. \end{aligned}$$

Donc pour $\Delta x \geq 0, \Delta z \geq 0$, le point (x^+, y^+, z^+) est strictement réalisable, alors si $\alpha = 1$, on obtient une itération de Newton complète strictement réalisable, ([5]).

2.3 Algorithme générale primal-dual de suivi de trajectoire

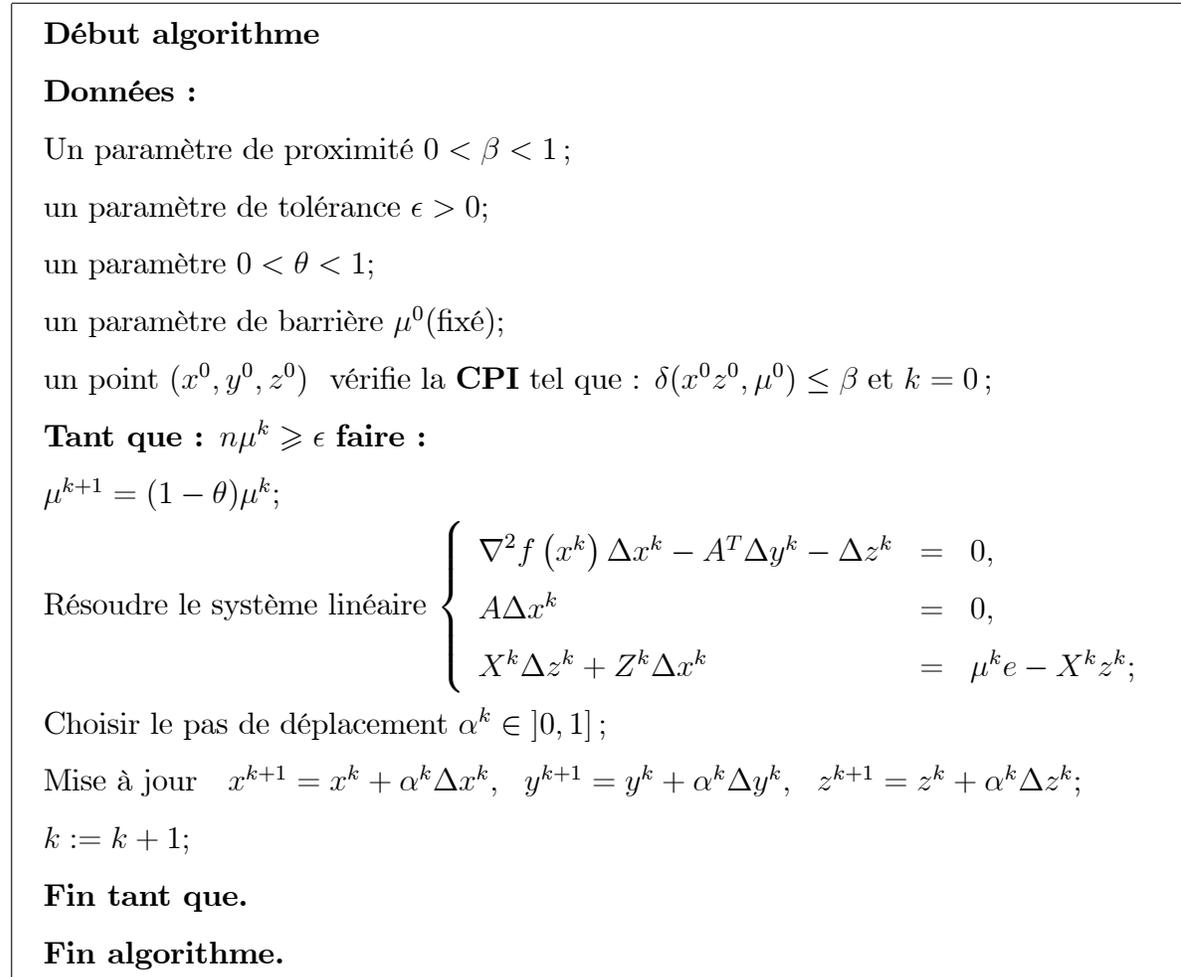


Fig.1. Algo 2.3

Chapter 3

Méthodes primal-dual pour (PCCL)

Rappelons que les solutions $(x(\mu), y(\mu), z(\mu))$ du problème \mathcal{P}_μ et \mathcal{D}_μ sont données dans le Chapitre 2, par les conditions d'optimalité données par le système d'équations non linéaires (2.3), qui est équivalent à :

$$\begin{cases} A^T y - \nabla f(x) + z = 0, \\ Ax = b, \\ \frac{xz}{\mu} = e, \quad x > 0, \quad z > 0. \end{cases} \quad (3.1)$$

Dans ce qui suit, on développe les directions de Newton et ainsi de suite des algorithmes primal-dual pour tracer approximativement la trajectoire centrale.

3.1 Directions de Newton

Cette section représente, la partie la plus importante de notre chapitre. On s'intéresse essentiellement au développement du pas de Newton.

Le nouvel itéré $(x + \Delta x, y + \Delta y, z + \Delta z)$ est obtenu par le système d'équations linéaires

suivant :

$$\begin{cases} A\Delta x & = 0, \\ A^T\Delta y + \Delta z - \nabla^2 f(x)\Delta x & = 0, \\ x\Delta z + z\Delta x & = \mu e - xz. \end{cases} \quad (3.2)$$

Pour faciliter l'analyse de ces algorithmes proposés (voir [44]), on introduit les notations suivantes :

les vecteurs v et d de \mathbb{R}^n sont définis par :

$$v_i = \sqrt{\frac{x_i z_i}{\mu}}, \text{ et } d_i = \sqrt{\frac{x_i}{z_i}},$$

pour tout $i = 1, \dots, n$. Par la suite, on fait le changement de variable :

$$d_x = \frac{d^{-1}\Delta x}{\sqrt{\mu}}, \quad d_z = \frac{d\Delta z}{\sqrt{\mu}} \text{ et } d_y = \Delta y.$$

Avec ces changements, on obtient :

$$x\Delta z + z\Delta x = \mu v(d_x + d_z), \quad (3.3)$$

et

$$d_x d_z = \frac{\Delta x \Delta z}{\mu}, \quad (3.4)$$

de plus, on a :

$$\bar{A} = \frac{1}{\sqrt{\mu}}AD \quad \text{et} \quad \bar{H} = D\nabla^2 f(x)D,$$

avec

$$D = \text{diag}(d).$$

Avec ces changements, le système linéaire **(3.2)** devient :

$$\begin{cases} \bar{A} d_x & = 0, \\ -\bar{A}^T d_y - d_z + \bar{H} d_x & = 0, \\ d_x + d_z & = v^{-1} - v, \end{cases} \quad (3.5)$$

où \bar{H} est aussi une matrice (*sdp*) et \bar{A} de plein rang et sous nos hypothèses citées au **Chapitre 2**, le système **(3.5)** admet une solution unique (d_x, d_y, d_z) . Maintenant, on observe que :

$$d_x^T d_z = d_x^T (\bar{H} d_x - \bar{A}^T d_y) \quad (3.6)$$

$$= d_x^T \bar{H} d_x \geq 0 \quad (3.7)$$

avec $\bar{A} d_x = 0$ et \bar{H} est une matrice (*sdp*).

Remarque 3.1.1 *Dans le cas de la programmation linéaire, les directions primales et duales sont orthogonales car :*

$$d_x^T d_z = \mu(\Delta x)^T \Delta z = 0,$$

tandis que dans la programmation convexe on a :

$$d_x^T d_z \geq 0,$$

qui rend l'analyse de convergence de l'algorithme un peu différent que la programmation linéaire.

Notre prochain travail, une étude théorique complète de la convergence et la complexité de l'algorithme 3.1.2.

3.1.1 Description algorithmique

Dans ce qui suit, d'après le système (3.2) on va présenter un algorithme primal-dual à petit pas pour tracer la trajectoire centrale. La proximité pour centrer les itérations est donnée par :

$$\delta(xz; \mu) := \frac{\|v^{-1} - v\|}{2},$$

(voir [49]), de plus on a : $\delta(xz; \mu) = 0 \iff v^{-1} = v \iff xz = \mu e$

3.1.2 Algorithme

La représentation générique de cet algorithme est donnée dans la figure 2 comme suit :

Début algorithme

Données:

Un paramètre de tolérance $\epsilon > 0$;

un paramètre de proximité $0 < \beta < 1$ (un seuil $\beta = \frac{1}{\sqrt{2}}$);

un paramètre $0 < \theta < 1$ (défaut $\theta = \frac{1}{2\sqrt{n}}$);

un paramètre de barrière μ^0 (fixé);

un point (x^0, y^0, z^0) vérifie la **CPI** tel que : $\delta(x^0 z^0, \mu^0) \leq \beta$ et $k = 0$;

Tant que: $n\mu^k \geq \epsilon$ **faire :**

$\mu^{k+1} = (1 - \theta)\mu^k$;

Résoudre le système linéaire $\begin{cases} \nabla^2 f(x^k) \Delta x^k - A^T \Delta y^k - \Delta z^k = 0, \\ A \Delta x^k = 0, \\ X^k \Delta z^k + Z^k \Delta x^k = \mu^k e - X^k z^k; \end{cases}$

Mise à jour $x^{k+1} = x^k + \Delta x^k, y^{k+1} = y^k + \Delta y^k, z^{k+1} = z^k + \Delta z^k; k := k + 1$;

Fin tant que.

Fin algorithme.

Fig.2. *Algo.3.1.2.*

3.2 Convergence de l'algorithme et analyse de la complexité

Dans cette section, on va étudier la convergence et la complexité de l'algorithme proposé, pour faciliter l'analyse on définit les vecteurs q_v et p_v de \mathbb{R}^n , (voir [1]) par :

$$q_v = d_x - d_z \quad \text{et} \quad p_v = d_x + d_z,$$

alors

$$d_x d_z = \frac{p_v^2 - q_v^2}{4} \quad (3.8)$$

et

$$\|q_v\| \leq \|p_v\|, \quad \text{car} \quad \|p_v\|^2 = \|q_v\|^2 + 4d_x^T d_z.$$

3.2.1 Résultats préliminaires

Lemme 3.2.1 *Soit (d_x, d_y, d_z) la solution du système (3.5) et si $\delta := \delta(xz; \mu)$, et $\mu > 0$, alors, on a :*

$$0 \leq d_x^T d_z \leq 2\delta^2, \quad (3.9)$$

et

$$\|d_x d_z\|_\infty \leq \delta^2 \quad \text{et} \quad \|d_x d_z\| \leq \sqrt{2}\delta^2. \quad (3.10)$$

Preuve. Puisque $d_x^T d_z \geq 0$ donc, il est facile de vérifier que :

$$\|d_x\|^2 + \|d_z\|^2 + 2d_x^T d_z = \|d_x + d_z\|^2 = \|v^{-1} - v\|^2 = 4\delta^2$$

Pour le deuxième résultat, on utilise **(3.10)**, donc on en peut écrire :

$$\begin{aligned}
\|d_x d_z\|_\infty &\leq \frac{1}{4}(\|(d_x + d_z)^2 - (d_x - d_z)^2\|_\infty) \\
&\leq \frac{1}{4} \max(\|p_v\|_\infty^2, \|q_v\|_\infty^2) \\
&\leq \frac{1}{4} \max(\|p_v\|^2, \|q_v\|^2) \text{ (car } \|\cdot\|_\infty \leq \|\cdot\|), \\
&= \frac{1}{4} \|p_v\|^2 = \frac{1}{4} \|v^{-1} - v\|^2 = \delta^2.
\end{aligned}$$

En utilisant encore les résultats **(3.9)** et **(3.10)**, on obtient :

$$\begin{aligned}
\|d_x d_z\|^2 &= e^T (d_x d_z)^2 \\
&= \frac{1}{16} e^T (p_v^2 - q_v^2)^2 \\
&= \frac{1}{16} \|p_v^2 - q_v^2\|^2 \\
&\leq \frac{1}{16} (\|p_v^2\|^2 + \|q_v^2\|^2) \\
&\leq \frac{1}{16} (\|p_v\|^4 + \|q_v\|^4) \\
&\leq \frac{1}{8} \|p_v\|^4.
\end{aligned}$$

D'où le résultat.

Dans les prochains lemmes, nous commençons en premier à donner des conditions qui assurent la stricte faisabilité d'une itération de *Newton* complète. \square

3.2.2 Analyse de la faisabilité

Lemme 3.2.2 *Soit (x, z) un point primal-dual strictement réalisable. Alors*

$x^+ > 0$ et $z^+ > 0$ si et seulement si :

$$e + d_x d_z > 0.$$

Preuve. Comme $x^+ = x + \Delta x$, et $z^+ = z + \Delta z$ et on a pour chaque $\alpha \in [0, 1]$,

$$x(\alpha) = x + \alpha\Delta x, z(\alpha) = z + \alpha\Delta z.$$

Donc

$$x^0 = x(0) = x, x^1 = x(1) = x^+ \text{ et la même chose pour } z.$$

Alors

$$x^0 z^0 = xz > 0,$$

on a :

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

alors par le résultat :

$$x\Delta z + z\Delta x = \mu e - xz,$$

$$x(\alpha) z(\alpha) = xz + \alpha(\mu e - xz) + \alpha^2\Delta x\Delta z,$$

donc

$$\frac{x^+ z^+}{\mu} = e + d_x d_z, \tag{3.11}$$

et si $e + d_x d_z > 0$, on déduit que $\Delta x\Delta z > -\mu e$ donc

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

Alors $x(\alpha)z(\alpha) > 0$ pour tout $\alpha \in [0, 1]$. Comme $x(\alpha)$ et $z(\alpha)$ sont des fonctions linéaires en α , donc elles ne changent pas de signe sur l'intervalle $[0, 1]$ et on a pour $\alpha = 0$ que $x > 0$ et $z > 0$, alors $x(1) > 0$ et $z(1) > 0$. \square

Lemme 3.2.3 *Si $\delta < 1$, alors l'itération de Newton complet, est strictement réalisable c.à.d. $x^+ > 0$ et $z^+ > 0$.*

Preuve. D'après Lemme 3.2.2, on a vu que :

$$x^+ > 0 \text{ et } z^+ > 0$$

si et seulement si :

$$e + d_x d_z > 0,$$

ou bien si :

$$1 + (d_x)_i (d_z)_i > 0, \text{ pour tout } i.$$

Alors, on a :

$$\begin{aligned} 1 + (d_x)_i (d_z)_i &\geq 1 - |(d_x)_i (d_z)_i| \\ &\geq 1 - \|d_x d_z\|_\infty, \text{ pour tout } i. \end{aligned}$$

Maintenant, d'après (3.11), on déduit que :

$$1 - \|d_x d_z\|_\infty > 1 - \delta^2.$$

Donc, la condition $e + d_x d_z > 0$ est satisfaite si $\delta < 1$. □

Le Lemme suivant analyse l'influence d'une itération complète de *Newton* sur la proximité.

Lemme 3.2.4 *Si $\delta < 1$, alors :*

$$\delta_+(x^+ z^+; \mu) \leq \frac{\delta^2}{\sqrt{2(1 - \delta^2)}}.$$

Si $\delta < \frac{1}{\sqrt{2}}$, alors

$$\delta_+(x^+ z^+; \mu) \leq \delta^2,$$

donc la mesure de proximité converge d'une façon quadratique vers la trajectoire centrale.

Preuve. Par définition, et d'après (3.11) on a :

$$\begin{aligned} 4\delta_+^2 &= \|(v^+)^{-1} - (v^+)\|^2 \\ &= \|(v^+)^{-1}(e - (v^+)^2)\|^2 \\ &= \left\| \frac{d_x d_z}{\sqrt{e + d_x d_z}} \right\|^2. \end{aligned}$$

Maintenant, d'après les deux Lemmes (3.2.1) et (3.2.3), on déduit que :

$$\begin{aligned} 4\delta_+^2 &\leq \frac{\|d_x d_z\|^2}{1 - \|d_x d_z\|_\infty} \\ &\leq \frac{4\delta^4}{2(1 - \delta^2)}. \end{aligned}$$

Il s'ensuit que :

$$\delta_+(x^+ z^+; \mu) \leq \frac{\delta^2}{\sqrt{2(1 - \delta^2)}}.$$

Ce qui achève la démonstration du lemme. □

Dans le lemme suivant, nous analysons l'influence de l'itération de *Newton* par la mise à jour du paramètre $\mu^+ = (1 - \theta)\mu$, sur la mesure de proximité.

Lemme 3.2.5 *Si $\delta < \frac{1}{\sqrt{2}}$ et $\mu^+ = (1 - \theta)\mu$ où $0 < \theta < 1$, alors :*

$$\delta^2(x^+ z^+; \mu^+) \leq (1 - \theta)\delta_+^2 + \frac{\theta^2(n + 1)}{4(1 - \theta)} + \frac{\theta}{2}.$$

De plus, si $\theta = \frac{1}{2\sqrt{n}}$ et $n \geq 2$, alors on a :

$$\delta(x^+ z^+; \mu^+) \leq \frac{1}{\sqrt{2}}.$$

Preuve. Soit $v^+ = \sqrt{\frac{x^+ z^+}{\mu}}$ et $\mu^+ = (1 - \theta) \mu$. Alors on a :

$$\begin{aligned}
4\delta^2(x^+ z^+, \mu^+) &= \left\| \left(\sqrt{\frac{\mu^+}{x^+ z^+}} \right) - \left(\sqrt{\frac{x^+ z^+}{\mu^+}} \right) \right\|^2 \\
&= \left\| \sqrt{1 - \theta} (v^+)^{-1} - \frac{1}{\sqrt{1 - \theta}} v^+ \right\|^2 \\
&= \left\| \sqrt{1 - \theta} ((v^+)^{-1} - v^+) - \frac{\theta}{\sqrt{1 - \theta}} v^+ \right\|^2 \\
&= (1 - \theta) \|(v^+)^{-1} - v^+\|^2 + \frac{\theta^2}{1 - \theta} \|v^+\|^2 - 2\theta ((v^+)^{-1} - v^+)^T v^+ \\
&= 4(1 - \theta)\delta_+^2 + \frac{\theta^2}{1 - \theta} \|v^+\|^2 - 2\theta ((v^+)^{-1} - v^+)^T v^+ \\
&= 4(1 - \theta)\delta_+^2 + \frac{\theta^2}{1 - \theta} \|v_+\|^2 - 2\theta n + 2\theta \|v_+\|^2,
\end{aligned}$$

car $((v^+)^{-1})^T v^+ = n$ (d'après (3.11)) et $(v^+)^T v^+ = \|v_+\|^2$.

Maintenant, par Lemme 3.2.1, rappelons que :

$$0 \leq d_x^T d_z \leq 2\delta^2$$

et si $\delta < \frac{1}{\sqrt{2}}$ on a :

$$\|v_+\|^2 = \frac{1}{\mu} x_+^T z_+ \leq (n + 1), \quad (3.12)$$

par conséquent :

$$\delta^2(x^+ z^+; \mu^+) \leq (1 - \theta)\delta_+^2 + \frac{\theta^2(n + 1)}{4(1 - \theta)} + \frac{\theta}{2}.$$

Ce qui achève la preuve de la première partie du Lemme.

De plus si $\delta < \frac{1}{\sqrt{2}}$, alors $\delta_+^2 < \frac{1}{4}$ donc

$$\delta^2(x^+ z^+; \mu^+) \leq \frac{\theta^2(n + 1)}{4(1 - \theta)} + \frac{\theta}{2} + \frac{(1 - \theta)}{4}.$$

Si on prend $\theta = \frac{1}{2\sqrt{n}}$ alors $\theta^2 = \frac{1}{4n}$, il s'ensuit que :

$$\delta^2(x^+z^+; \mu^+) \leq \frac{1}{4n} \frac{(n+1)}{(1-\theta)} + \frac{\theta}{2} + \frac{(1-\theta)}{4},$$

et puisque $\frac{n+1}{4n} \leq \frac{3}{8}$ pour tout $n \geq 2$, on trouve :

$$\delta^2(x^+z^+; \mu^+) \leq \frac{3}{32(1-\theta)} + \frac{\theta}{2} + \frac{(1-\theta)}{4}.$$

Maintenant, pour $n \geq 2$, on a $0 \leq \theta \leq \frac{1}{2\sqrt{2}}$ posons :

$$f(\theta) = \frac{3}{32(1-\theta)} + \frac{(1-\theta)}{4} + \frac{\theta}{2},$$

alors f est une fonction continue strictement croissante sur l'intervalle $\left[0, \frac{1}{2\sqrt{2}}\right]$, car :

$$f'(\theta) = \frac{96}{(32\theta - 32)^2} + \frac{1}{4} > 0 \text{ avec } 0 < \theta < 1,$$

et par conséquent :

$$f(\theta) \leq f\left(\frac{1}{2\sqrt{2}}\right) \simeq 0.48341 < \frac{1}{2}, \forall \theta \in \left[0, \frac{1}{2\sqrt{2}}\right],$$

donc $\delta(x^+z^+; \mu^+) < \frac{1}{\sqrt{2}}$. Ce qui achève la preuve du Lemme. □

Par Lemme 3.2.5, on déduit que pour les paramètres $\theta = \frac{1}{2\sqrt{n}}$ et $\beta = \frac{1}{\sqrt{2}}$, les conditions $x > 0$ et $z > 0$ et $\delta(x^+z^+; \mu^+) < \frac{1}{\sqrt{2}}$ sont maintenus durant la procédure de la solution, donc **Algo 3.1.2** est bien défini.

Dans le lemme suivant, on analyse l'influence d'un pas de Newton complet sur la mise à jours du saut de dualité.

Lemme 3.2.6 Soient $\delta < \frac{1}{\sqrt{2}}$ et $\mu^+ = (1 - \theta)\mu$, alors le nouveau saut de dualité vérifie :

$$(x^+)^T z^+ \leq \mu(n + 1), \text{ pour tout } n \in \mathbb{N}. \quad (3.13)$$

Preuve. Du Lemme 3.2.5, (3.12) on déduit directement que :

$$\frac{1}{\mu}(x^+)^T z^+ \leq (n + 1).$$

Ce qui achève la preuve du Lemme. □

3.2.3 Analyse de la complexité

La détermination du nombre total d'itérations nécessaires pour trouver une solution optimale primale-duale nécessite le calcul du nombre d'itérations externes.

Théorème de la complexité

Théorème 3.2.1 Soient $\theta = \frac{1}{2\sqrt{n}}$ et $\mu^0 = \frac{1}{2}$. Alors, le nombre des itérations produites par **Algo 3.1.2**

$$\mathcal{O}(\sqrt{n} \log \frac{n}{\epsilon})$$

itérations.

Pour démontré ce Théorème on a le lemme suivant, on donnera une borne supérieure sur le nombre des itérations généré par **Algo 3.1.2** avec une précision.

Lemme 3.2.7 Soient x^{k+1}, z^{k+1} désignent le $(k + 1)$ -ème itéré produit par **Algo 3.1.2** avec $\mu := \mu^k$. Alors

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

si :

$$k \geq \left\lceil \frac{1}{\theta} \log \frac{2\mu^0 n}{\epsilon} \right\rceil.$$

Preuve. Par Lemme 3.2.6, on a vu que :

$$(x^{k+1})^T z^{k+1} \leq \mu^k (n+1) \quad (3.14)$$

et que :

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

Alors :

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

puisque $n+1 \leq 2n$ pour tout $n \geq 1$ alors on a :

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

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

$$(1-\theta)^k 2\mu^0 n \leq \epsilon,$$

on prend le logarithme de cette dernière inégalité, on déduit que :

$$k \log(1-\theta) \leq \log \epsilon - \log 2\mu^0 n$$

maintenant de l'inégalité $-\log(1-\theta) \geq \theta$ pour tout $0 < \theta < 1$, on obtient :

$$k\theta \geq \log \frac{2n\mu^0}{\epsilon}.$$

Ce qui termine la preuve. □

Chapter 4

Simulation numérique

Nous présentons dans ce chapitre quelques résultats numériques issues de la mise en œuvre d'algorithme **Algo**. Cet algorithme est programmé en **Turbo-Pascal**. Le point de départ est désigné par (x^0, y^0, z^0) , la précision $\epsilon \in [10^{-6}, 10^{-8}]$. Les solutions optimales primales et duales sont désignées par $(x_{optimal})$ et $(y_{optimal}, z_{optimal})$, respectivement, et par $f(opt)$ et $g(opt)$ sont les valeurs optimales de l'objectif primaire et duale. Finalement, on note par **Iter**, le nombre des itérations nécessaires pour obtenir une solution optimale primale-duale et **Temps(s)**, le temps de calcul en seconde.

4.1 Mise en œuvre de cet algorithme

4.1.1 Paramètre barrière

Les valeurs du paramètre barrière sont :

$$\mu^{k+1} = (1 - \theta)\mu^k.$$

Les valeurs du paramètre θ sont : $0 < \theta < 1$,

Sont choisis comme suit :

On prend $\theta = \frac{1}{2\sqrt{n}}$ (théorique).

De plus, on a utilisé d'autre choix relaxés de θ (numérique).

Pour $n \geq 2$, on a $0 < \theta \leq \frac{1}{2\sqrt{2}}$:

$$\theta = \frac{1}{3\sqrt{n}}, \theta = \frac{1}{5\sqrt{n}}, \theta = \frac{1}{7\sqrt{n}}, \dots, \frac{1}{k\sqrt{n}}, k \in \mathbb{N} \text{ et } k > 2.$$

4.1.2 Directions de déplacement

La direction de déplacement est calculée par la méthode de **pivotage de Gauss**.

4.1.3 Point d'initialisation

Pour avoir un point de départ, on passe par les étapes suivante :

- 1) On choisit un $x^0 > 0$, tel que $Ax^0 = b$.
- 2) Puis, on prend un y^0 de \mathbb{R}^n arbitraire.
- 3) En remplaçant x^0, y^0 calculés dans les étapes 1 et 2 dans l'équation suivante :

$$A^T y + z - \nabla f(x) = 0,$$

pour obtenir un $z^0 > 0$.

- 4) Finalement, on exige que le point (x^0, y^0, z^0) , vérifie la condition de proximité :

$$\delta(x^0 z^0; \mu^0) \leq \frac{1}{\sqrt{2}}.$$

4.2 Résultats numériques

La programmation linéaire et la programmation quadratique ce sont des cas particuliers de la programmation convexe.

4.2.1 Programmation linéaire

Un problème de programmation linéaire (*PL*) consiste à minimiser une fonction linéaire sous contraintes linéaires. Il s'agit des problèmes d'optimisation les plus célèbres, les plus simples théoriquement. Le problème suivant est un programme linéaire sous forme standard

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

Le dual de \mathcal{P} , est donné par :

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

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

Problème 1. Soit le programme linéaire suivant :

$$A = \begin{pmatrix} 7 & 2 & 3 & 1 & -1 & -2 & 4 \\ -4 & -5 & -2 & 3 & -5 & 9 & 6 \\ 2 & 7 & -6 & 7 & -3 & 4 & 2 \\ 6 & -6 & -1 & 7 & 5 & -5 & 3 \end{pmatrix}, \quad b = \begin{pmatrix} 14 \\ 2 \\ 13 \\ 9 \end{pmatrix},$$
$$c = \left(2.8, 1, 0.7, 2.9, 0.5, 1.4, 2.9 \right)^T.$$

Point de départ (Initialisation) :

$$x^0 = (1, 1, 1, 1, 1, 1, 1)^T,$$

$$z^0 = (1, 1, 1, 1, 1, 1, 1)^T,$$

$$y^0 = (0.2, 0.1, 0.1, 0.1)^T.$$

La solution optimale est :

$$x_{\text{optimal}} = (0.9707, 0.7082, 0, 0.4934, 0, 0, 1.3239)^T,$$

$$z_{\text{optimal}} = (0, 0, 1.3514, 0, 1.7571, 0.3954, 0)^T,$$

$$y_{\text{optimal}} = (0.3421, 0.1388, 0.2189, 0.0871)^T,$$

et

$f(\text{opt})$	$g(\text{opt})$	saut de dualité
8.6961	8.6960	$7.059e - 005$

Les résultats numériques obtenus pour Problème 1 sont donnés dans le tableau suivant :

μ	θ	$\frac{1}{7\sqrt{n}}$	$\frac{1}{5\sqrt{n}}$	$\frac{1}{3\sqrt{n}}$	$\frac{1}{2\sqrt{n}}$
0.95	Iter	52	37	22	12
	Temps	0.04	0.04	0.02	0.00
0.85	Iter	72	52	31	20
	Temps	0.05	0.05	0.06	0.01
0.65	Iter	90	64	38	25
	Temps	0.05	0.05	0.06	0.02

Tableau 1.

Problème 2. Soit le programme linéaire suivant :

$$A = \begin{pmatrix} 0 & 1 & 2 & -1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 2 & 3 & 4 & -1 & 0 & 1 & 0 & 0 \\ -1 & 0 & -2 & 1 & 2 & 0 & 0 & 1 & 0 \\ 1 & 2 & 0 & -1 & -2 & 0 & 0 & 0 & 1 \\ 1 & 3 & 4 & 2 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}, b = \begin{pmatrix} 4 \\ 10 \\ 1 \\ 1 \\ 11 \end{pmatrix},$$

$$c = \left(1.01, 1.09, 1.07, 1.05, 1.04, 1.02, 1.01, 1.02, 1.01 \right)^T.$$

Point de départ :

$$x^0 = (1, 1, 1, 1, 1, 1, 1, 1, 1)^T,$$

$$z^0 = (1, 1, 1, 1, 1, 1, 1, 1, 1)^T,$$

$$y^0 = (0.02, 0.01, 0.02, 0.01, 0.01)^T.$$

La solution optimale est :

$$x_{\text{optimal}} = (0, 1.7, 0.7, 1.2667, 0.5667, 1.6, 0, 0, 0)^T,$$

$$z_{\text{optimal}} = (1.3333, 0, 0, 0, 0, 0, 0.4167, 0.4583, 0.9583)^T,$$

$$y_{\text{optimal}} = (1.02, 0.5933, 0.5617, 0.0517, -0.4067)^T.$$

$f(\text{opt})$	$g(\text{opt})$	saut de dualité
6.1533	6.1532	$1.062e - 004$

Les résultats numériques obtenus pour Problème 2 sont donnés dans le tableau suivant :

μ	θ	$\frac{1}{7\sqrt{n}}$	$\frac{1}{5\sqrt{n}}$	$\frac{1}{3\sqrt{n}}$	$\frac{1}{2\sqrt{n}}$
0.95	Iter	61	44	26	17
	Temps	0.05	0.05	0.00	0.00
0.85	Iter	85	60	36	24
	Temps	0.06	0.02	0.01	0.00
0.75	Iter	96	69	41	27
	Temps	0.05	0.06	0.03	0.00
0.65	Iter	105	74	44	29
	Temps	0.09	0.05	0.03	0.01

Tableau 2.

Problème 3. Soit le programme linéaire suivant :

$$\begin{pmatrix} 2 & 1 & 1 & 0 & 0 \\ 1 & 2 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}, c = \begin{pmatrix} 1.6 \\ 1.8 \\ 1.2 \\ 1.2 \\ 1.2 \end{pmatrix}, b = \begin{pmatrix} 4 \\ 4 \\ 2 \end{pmatrix}.$$

Point de départ :

$$x^0 = (1, 1, 1, 1, 1)^T,$$

$$z^0 = (1, 1, 1, 1, 1)^T,$$

$$y^0 = (0.2, 0.2, 0.2)^T.$$

La solution optimale est :

$$x_{\text{optimal}} = (1.3333, 1.3333, 0, 0, 0.6667)^T,$$

$$z_{\text{optimal}} = (0, 0, 0.3333, 1.3333, 0)^T,$$

$$y_{\text{optimal}} = (0.8667, -0.1333, 1.2)^T.$$

$f(\text{opt})$	$g(\text{opt})$	saut de dualité
5.3333	5.3333	$3.882e - 005$

Les résultats numériques obtenus pour Problème 3 sont donnés dans le tableau suivant :

μ	θ	$\frac{1}{7\sqrt{n}}$	$\frac{1}{5\sqrt{n}}$	$\frac{1}{3\sqrt{n}}$	$\frac{1}{2\sqrt{n}}$
0.95	Iter	42	30	18	12
	Temps	0.05	0.05	0.05	0.00
0.85	Iter	59	42	25	17
	Temps	0.06	0.05	0.05	0.00
0.75	Iter	68	48	29	19
	Temps	0.05	0.05	0.03	0.01
0.65	Iter	74	52	31	20
	Temps	0.05	0.04	0.03	0.01

Tableau 3.

Problème 4. Soit le programme linéaire suivant donné par :

$$A = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}, \quad b = \begin{pmatrix} 5 \\ 5 \\ 4 \\ 4 \\ 4 \\ 3 \\ 3 \end{pmatrix},$$

$$c = (0.5, 0.8, -2, -0.5, 0.82, -1.98, 0.5, 0.82, -1.98, 0.5, 0.82, -1.98, 0.5, 0.82)^T.$$

Point de départ (Initialisation) :

$$x^0 = (1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)^T,$$

$$z^0 = (1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)^T,$$

$$y^0 = (-0.5, -0.2, -3, -1, 0.02, 0.02, 1)^T.$$

La solution optimale est :

$$x_{\text{optimal}} = (1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)^T,$$

$$z_{\text{optimal}} = 10^{-4}(2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)^T,$$

$$y_{\text{optimal}} = (0.5, 0.8, -2, -1, 0.02, 0.02, 1)^T.$$

$f(\text{opt})$	$g(\text{opt})$	saut de dualité
-2.3600	-2.3601	$1.506e - 004$

Les résultats numériques obtenus pour Problème 4 sont donnés dans le tableau suivant :

μ	θ	$\frac{1}{7\sqrt{n}}$	$\frac{1}{5\sqrt{n}}$	$\frac{1}{3\sqrt{n}}$	$\frac{1}{2\sqrt{n}}$
0.95	Iter	80	57	34	23
	Temps	0.09	0.07	0.06	0.06
0.85	Iter	110	78	47	31
	Temps	0.11	0.06	0.06	0.06
0.75	Iter	125	89	53	35
	Temps	0.11	0.11	0.05	0.04
0.65	Iter	135	96	57	38
	Temps	0.17	0.05	0.05	0.05

Tableau 4.

4.2.2 Programmation quadratique

Un problème de programmation quadratique consiste à minimiser une fonction quadratique sous contraintes linéaires.

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

$$(\mathcal{P}) \quad \min_x [c^T x + \frac{1}{2} x^T Q x] \text{ s à : } Ax = b, \quad x \geq 0,$$

et son dual :

$$(\mathcal{D}) \quad \begin{cases} \max_{(y,z)} [b^T y - \frac{1}{2} x^T Q x] \text{ s à : } A^T y + z - Qx = c, \\ x \geq 0, z \geq 0, \end{cases}$$

où Q est une matrice carrée symétrique 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 5. Soit le programme quadratique convexe suivant :

$$Q = \begin{pmatrix} 2 & 1 & 0 & 0 & 0 & 0 \\ 1 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, A = \begin{pmatrix} \frac{5}{12} & -1 & 1 & 0 & 0 & 0 \\ \frac{5}{2} & 1 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}, b = \begin{pmatrix} \frac{35}{12} \\ \frac{35}{2} \\ 5 \\ 5 \end{pmatrix},$$

$$c = \begin{pmatrix} -30, & -30, & 0, & 0, & 0, & 0 \end{pmatrix}^T.$$

Point de départ :

$$x^0 = (4.7399, 4.2328, 5.1745, 1.4173, 9.3799, 0.7672)^T,$$

$$z^0 = (2.2411, 2.5161, 2.0281, 7.4499, 0.9415, 13.8886)^T,$$

$$y^0 = (-2.02809, -7.4499, -0.9415, -13.8886)^T.$$

La solution optimale est :

$$x_{\text{optimal}} = (5.0000, 5.0000, 5.8333, 0.0000, 10.0000, 0.0000)^T,$$

$$z_{\text{optimal}} = (0.0000, 0.0000, 0.0000, 6.0000, 0.0000, 9.0000)^T,$$

$$y_{\text{optimal}} = (0.0000, -6.0000, 0.0000, -9.0000)^T,$$

et

$f(\text{opt})$	$g(\text{opt})$	saut de dualité
-224.9997	-224.9997	$1.6429e - 005$

Les résultats numériques obtenus pour Problème 5 sont donnés dans le tableau suivant :

μ	θ	$\frac{1}{7\sqrt{n}}$	$\frac{1}{5\sqrt{n}}$	$\frac{1}{3\sqrt{n}}$	$\frac{1}{2\sqrt{n}}$
0.95	Iter	34	24	15	10
	Temps	0.00	0.00	0.00	0.00
0.65	Iter	119	85	50	33
	Temps	0.05	0.05	0.05	0.00

Tableau 5.

Problème 6. Soit le programme quadratique convexe suivant :

$$Q = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}, c = \begin{pmatrix} -3 \\ -10 \\ 0 \\ 0 \end{pmatrix}, A = \begin{pmatrix} -1 & 1 & 1 & 0 \\ 2 & 3 & 0 & 1 \end{pmatrix}, b = \begin{pmatrix} 2 \\ 11 \end{pmatrix}.$$

Point de départ :

$$\begin{aligned} x^0 &= (1.5000, 2.0000, 1.5000, 2.0000)^T, \\ z^0 &= (2.0000, 2.0000, 5.0000, 6.0000)^T, \\ y^0 &= (-2.0000, -2.0000)^T. \end{aligned}$$

La solution optimale est :

$$\begin{aligned} x_{\text{optimal}} &= (1.0000, 3.0000, 0.0000, 0.0000)^T, \\ z_{\text{optimal}} &= (0.0000, 0.0000, 1.0000, 1.0000)^T, \\ y_{\text{optimal}} &= (-1.0000, -1.0000)^T. \end{aligned}$$

$f(opt)$	$g(opt)$	saut de dualité
-22.9999	-23.0000	$4.042e - 005$

Les résultats numériques obtenus pour Problème 6 sont donnés dans le tableau suivant :

μ	θ	$\frac{1}{7\sqrt{n}}$	$\frac{1}{5\sqrt{n}}$	$\frac{1}{3\sqrt{n}}$	$\frac{1}{2\sqrt{n}}$
0.95	Iter	69	49	29	19
	Temps	0.01	0.01	0.00	0.00
0.65	Iter	83	59	35	23
	Temps	0.05	0.01	0.01	0.00

Tableau 6.

Problème 7. Soit le programme quadratique convexe suivant :

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

$$c = \left(1, -1.5, 2, 1.5, 3 \right)^T, \quad b = \left(9.31, 5.45, 7.06 \right)^T.$$

Point de départ :

$$x^0 = (2.4200, 1.0000, 1.5500, 2.3000, 1.465)^T,$$

$$z^0 = (3.0600, 15.7190, 8.1750, 7.2250, 7.8700)^T,$$

$$y^0 = (20.0000, 11.0000, 5.0000)^T.$$

La solution optimale est :

$$x_{\text{optimal}} = (2.6604, 0.7035, 1.3244, 2.4894 \ 1.1645)^T,$$

$$z_{\text{optimal}} = (0.0000, 0.0000, 0.0000, 0.0000, 0.0000)^T,$$

$$y_{\text{optimal}} = (25.0011, 12.1538 \ 5.6674).$$

et

$f(\text{opt})$	$g(\text{opt})$	le saut de dualité
175.2458	175.2457	$5.996e - 005$

Les résultats numériques obtenus pour Problème 7 sont donnés dans le tableau suivant :

μ	θ	$\frac{1}{7\sqrt{n}}$	$\frac{1}{5\sqrt{n}}$	$\frac{1}{3\sqrt{n}}$	$\frac{1}{2\sqrt{n}}$
0.95	Iter	70	50	30	19
	Temps	0.05	0.03	0.01	0.00
0.65	Iter	108	76	45	29
	Temps	0.11	0.06	0.02	0.01

Tableau 7.

Problème 8. Soit le programme quadratique convexe suivant :

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

$$A = \begin{pmatrix} 1 & -1 & 1.9 & 1.25 & 1.2 & 0.4 & -0.7 & 1.06 & 1.5 & 1.05 \\ 1.3 & 1.2 & 0.15 & 2.15 & 1.25 & 1.5 & 0.4 & 1.52 & 1.3 & 1 \\ 1.5 & -1.1 & 3.5 & 1.25 & 1.8 & 2 & 1.95 & 1.2 & 1 & -1 \end{pmatrix}, \quad b = \begin{pmatrix} 11.651 \\ 16.672 \\ 21.294 \end{pmatrix},$$

$$c = \left(-0.5, -1, 0, 0, -0.5, 0, 0, -1, -0.5, -1 \right)^T.$$

Point de départ :

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

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

$$y^0 = (5.6566, 16.7085, 3.6733)^T.$$

La solution optimale est donnée par :

$$x_{\text{optimal}} = (0.968, 0.517, 1.746, 1.909, 1.249, 2.588, 1.339, 1.622, 0.825, 0.896)^T,$$

$$z_{\text{optimal}} = (0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000)^T,$$

$$y_{\text{optimal}} = (4.1830, 22.4678, 5.2352)^T,$$

et

$f(\text{opt})$	$g(\text{opt})$	saut de dualité
264.1339	264.1509	$1.702e - 002$

Les résultats numériques obtenus pour Problème 8 sont donnés dans le tableau suivant :

μ	θ	$\frac{1}{7\sqrt{n}}$	$\frac{1}{5\sqrt{n}}$	$\frac{1}{3\sqrt{n}}$	$\frac{1}{2\sqrt{n}}$
0.95	Iter	118	84	50	33
	Temps	0.20	0.09	0.05	0.05
0.65	Iter	157	112	66	43
	Temps	0.15	0.11	0.08	0.06

Tableau 8.

4.2.3 Cas d'une fonction convexe

Soit le programme convexe suivant :

$$\begin{cases} \min f(x) = \sum_{i=1}^n x_i \ln \frac{x_i}{a_i} \text{ s à :} \\ x_i + x_{i+m} = b_i, \quad i = 1, \dots, m \quad (n = 2m) \\ x \geq 0. \end{cases}$$

Où $a_i > 0$ fixés et $b_i \in \mathbb{R}$. On peut varier les dimensions m et n .

Problème 9.

$$a_i = 1, b_i = 1, n = 10, m = 5.$$

Point de départ :

$$x^0 = (0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5)^T,$$

$$z^0 = 1.3068(1, 1, 1, 1, 1, 1, 1, 1, 1, 1)^T,$$

$$y^0 = (-1, -1, -1, -1, -1)^T.$$

La solution optimale est :

$$x_{\text{optimal}} = (0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5)^T,$$

$$z_{\text{optimal}} = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0)^T,$$

$$y_{\text{optimal}} = (0.3068, 0.3068, 0.3068, 0.3068, 0.3068)^T ;$$

$f(\text{opt})$	$g(\text{opt})$	saut de dualité
-3.4657	-3.4659	$1.313e - 004$

Les résultats numériques obtenus pour Problème 9 sont donnés dans le tableau suivant :

μ	θ	$\frac{1}{7\sqrt{n}}$	$\frac{1}{5\sqrt{n}}$	$\frac{1}{3\sqrt{n}}$	$\frac{1}{2\sqrt{n}}$
0.95	Iter	107	76	45	30
	Temps	0.07	0.07	0.05	0.05
0.85	Iter	100	71	42	28
	Temps	0.06	0.06	0.06	0.06
0.75	Iter	86	61	36	24
	Temps	0.05	0.05	0.05	0.01
0.65	Iter	15	11	7	5
	Temps	0.00	0.00	0.00	0.00

Tableau 9.

Problème 10.

$$a_i = 1, b_i = 1, n = 14, m = 7$$

Point de départ :

$$x^0 = (0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5)^T,$$

$$z^0 = 1.3068(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)^T,$$

$$y^0 = (-1, -1, -1, -1, -1, -1, -1)^T.$$

La solution optimale est :

$$x_{\text{optimal}} = (0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5)^T,$$

$$z_{\text{optimal}} = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)^T,$$

$$y_{\text{optimal}} = (0.3068, 0.3068, 0.3068, 0.3068, 0.3068, 0.3068, 0.3068)^T,$$

et

$f(opt)$	$g(opt)$	saut de dualité
-4.8520	-4.8525	$5.407e - 004$

Les résultats numériques obtenus pour Problème 10 sont donnés dans le tableau suivant

μ	θ	$\frac{1}{7\sqrt{n}}$	$\frac{1}{5\sqrt{n}}$	$\frac{1}{3\sqrt{n}}$	$\frac{1}{2\sqrt{n}}$
0.95	Iter	131	94	56	37
	Temps	0.17	0.09	0.07	0.06
0.85	Iter	122	87	52	34
	Temps	0.11	0.06	0.06	0.05
0.75	Iter	106	75	45	30
	Temps	0.11	0.05	0.06	0.03
0.65	Iter	22	16	10	7
	Temps	0.06	0.06	0.00	0.00

Tableau 10.

Problème 11.

$$a_i = 1, b_i = 6, n = 14, m = 7.$$

Point de départ :

$$x^0 = (3, 2, 3, 2, 3, 2, 3, 3, 4, 3, 4, 3, 4, 3)^T$$

$$z^0 = (0.33, 0.4, 0.33, 0.4, 0.33, 0.4, 0.33, 0.33, 1.0938, 0.33, 1.0938, 0.33, 1.0938, 0.33)^T$$

$$y^0 = (1.7686, 1.2931, 1.7686, 1.2931, 1.7686, 1.2931, 1.7686)^T .$$

La solution optimale est :

$$x_{\text{optimal}} = (3, 2.960, 3, 2.960, 3, 2.960, 3, 3, 3.040, 3, 3.040, 3, 3.040, 3)^T$$

$$z_{\text{optimal}} = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)^T$$

$$y_{\text{optimal}} = (2.0986, 2.1306, 2.0986, 2.1306, 2.0986, 2.1306, 2.0986)^T$$

$f(\text{opt})$	$g(\text{opt})$	saut de dualité
46.1433	46.7163	$5.730e - 001$

Les résultats numériques obtenus pour Problème 11 sont donnés dans le tableau suivant

μ	θ	$\frac{1}{7\sqrt{n}}$	$\frac{1}{5\sqrt{n}}$	$\frac{1}{3\sqrt{n}}$	$\frac{1}{2\sqrt{n}}$
0.95	Iter	101	72	43	29
	Temps	0.11	0.06	0.06	0.06
0.85	Iter	103	73	44	29
	Temps	0.16	0.11	0.06	0.06
0.75	Iter	117	83	50	33
	Temps	0.16	0.11	0.05	0.05
0.65	Iter	129	92	55	36
	Temps	0.29	0.09	0.07	0.07

Tableau 11.

Problème 12.

$$a_i = 1, b_i = 1, n = 8, m = 4.$$

Point de départ :

$$x^0 = (0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5)^T,$$

$$z^0 = (0.8068, 0.8068, 0.8068, 0.8068, 0.8068, 0.8068, 0.8068, 0.8068)^T,$$

$$y^0 = (-0.5, -0.5, -0.5, -0.5)^T.$$

La solution optimale est donnée par :

$$x_{\text{optimal}} = (0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5)^T,$$

$$z_{\text{optimal}} = (0, 0, 0, 0, 0, 0, 0, 0)^T,$$

$$y_{\text{optimal}} = (0.3068, 0.3068, 0.3068, 0.3068)^T ;$$

$f(\text{opt})$	$g(\text{opt})$	saut de dualité
-2.7725	-2.7728	$2.9260e - 004$

Les résultats numériques obtenus pour Problème 12 sont donnés dans le tableau suivant

μ	θ	$\frac{1}{7\sqrt{n}}$	$\frac{1}{5\sqrt{n}}$	$\frac{1}{3\sqrt{n}}$	$\frac{1}{2\sqrt{n}}$
0.95	Iter	110	78	46	30
	Temps	0.01	0.05	0.01	0.03
0.85	Iter	107	76	45	30
	Temps	0.01	0.02	0.01	0.01
0.75	Iter	104	74	44	27
	Temps	0.06	0.06	0.02	0.01
0.65	Iter	98	70	42	27
	Temps	0.06	0.06	0.02	0.01

Tableau 12.

Cette implémentation est faite en **MATLAB R2010a** sur un pentium.

Problème.13.

$$a_i = 1, b_i = 1, n = 200, m = 100$$

Point de départ :

$x_i^0 = 0.5$	pour	$i = 1, \dots, 200,$
$z_i^0 = 1.3068$	pour	$i = 1, \dots, 200,$
$y_i^0 = -1$	pour	$i = 1, \dots, 100.$

La solution optimale est donnée par :

$x_{\text{optimal}} = 0.50$	pour	$i = 1, \dots, 200,$
$z_{\text{optimal}} = 0.00$	pour	$i = 1, \dots, 200,$
$y_{\text{optimal}} = 0.3068$	pour	$i = 1, \dots, 100.$

Les résultats numériques obtenus pour Problème 13 sont donnés dans le tableau suivant

μ	θ	$\frac{1}{7\sqrt{n}}$	$\frac{1}{5\sqrt{n}}$	$\frac{1}{2\sqrt{n}}$
0.95	Iter	1356	967	383
	Temps	35,9176	25.3477	10.1947
0.65	Iter	1319	940	372
	Temps	32.4777	24.5611	9,6544

Tableau 13.

4.2.4 Commentaires

À travers les tests numériques qu'on a réalisé sur des exemples des problèmes linéaires, problèmes quadratiques ainsi que des problèmes convexes de différentes dimensions, on remarque que les résultats numériques démontrent l'effet des deux nouveaux paramètres

sur la réduction du nombre d'itérations produit par cet algorithme.

4.3 Perspectives

Dans cette section, on constate qu'à travers d'autres choix de valeurs de paramètres barrières θ , que le nombre d'itérations produit par l'algorithme est réduit significativement. On mentionne que ces choix ne sont pas validés théoriquement, mais elles ont besoin d'une analyse convenable notamment la convergence de l'algorithme correspondant, en appuyant sur la détermination d'un nouveau voisinage c.à.d. le choix du paramètre $\beta > 0$ et par conséquent le choix d'une nouvelle mesure de proximité $\delta > 0$.

Parmi ces choix qu'on a utilisés sont :

$$\theta = \frac{1}{\sqrt{n}}, \theta = \frac{1}{\sqrt{n/2}}, \theta = \frac{1}{\sqrt{n/3}}, \theta = \frac{1}{\sqrt{n/5}}, \theta = \frac{1}{\sqrt{n/7}}, \dots, \frac{1}{\sqrt{n/n'}} \text{ tel que } n' < n,$$

tel que : $\frac{1}{2\sqrt{2}} < \theta < 1, n \geq 2$.

Les résultats numériques obtenus par ces valeurs de θ , pour les Problèmes précédents sont donnés dans les tableaux suivants :

Problème 1.

μ	θ	$\frac{1}{\sqrt{n}}$	$\frac{1}{\sqrt{n/2}}$	$\frac{1}{\sqrt{n/3}}$	$\frac{1}{\sqrt{n/5}}$
0.95	Iter	7	5	4	3
	Temps	0.01	0.00	0.00	0.00
0.85	Iter	10	7	5	4
	Temps	0.00	0.00	0.00	0.00
0.65	Iter	12	8	6	4
	Temps	0.00	0.00	0.00	0.00

Tableau 1*

Problème 2.

μ	θ	$\frac{1}{\sqrt{n}}$	$\frac{1}{\sqrt{n/2}}$	$\frac{1}{\sqrt{n/3}}$
0.95	Iter	9	6	5
	Temps	0.00	0.00	0.00
0.85	Iter	11	8	6
	Temps	0.00	0.00	0.00
0.75	Iter	13	9	7
	Temps	0.00	0.00	0.00
0.65	Iter	14	9	7
	Temps	0.01	0.00	0.00

Tableau 2*

Problème.3

μ	θ	$\frac{1}{\sqrt{n}}$	$\frac{1}{\sqrt{n/2}}$	$\frac{1}{\sqrt{n/3}}$
0.95	Iter	6	4	3
	Temps	0.00	0.00	0.00
0.85	Iter	8	5	4
	Temps	0.00	0.00	0.00
0.75	Iter	9	6	4
	Temps	0.00	0.00	0.00
0.65	Iter	10	6	5
	Temps	0.00	0.00	0.00

Tableau 3*

Problème.4

μ	θ	$\frac{1}{\sqrt{n}}$	$\frac{1}{\sqrt{n/2}}$	$\frac{1}{\sqrt{n/3}}$	$\frac{1}{\sqrt{n/5}}$	$\frac{1}{\sqrt{n/7}}$
0.95	Iter	11	8	6	5	4
	Temps	0.02	0.00	0.00	0.00	0.00
0.85	Iter	15	10	8	6	5
	Temps	0.06	0.05	0.00	0.00	0.00
0.75	Iter	17	12	9	7	5
	Temps	0.00	0.00	0.00	0.00	0.00
0.65	Iter	18	12	10	7	6
	Temps	0.02	0.00	0.00	0.00	0.00

Tableau 4*

Problème.5

μ	θ	$\frac{1}{\sqrt{n}}$	$\frac{1}{\sqrt{n/2}}$	$\frac{1}{\sqrt{n/3}}$	$\frac{1}{\sqrt{n/5}}$
0.95	Iter	5	4	3	2
	Temps	0.00	0.00	0.00	0.00
0.65	Iter	15	10	7	4
	Temps	0.00	0.00	0.00	0.00

Tableau 5*

Problème 6

μ	θ	$\frac{1}{\sqrt{n}}$	$\frac{1}{\sqrt{n/2}}$	$\frac{1}{\sqrt{n/3}}$
0.95	Iter	9	6	4
	Temps	0.00	0.00	0.00
0.65	Iter	10	6	5
	Temps	0.00	0.00	0.00

Tableau 6*

Problème.7

μ	θ	$\frac{1}{\sqrt{n}}$	$\frac{1}{\sqrt{n/2}}$	$\frac{1}{\sqrt{n/3}}$	$\frac{1}{\sqrt{n/4}}$
0.95	Iter	9	6	5	4
	Temps	0.00	0.00	0.00	0.00
0.65	Iter	13	9	6	5
	Temps	0.01	0.00	0.00	0.00

Tableau 7*

Problème.8

μ	θ	$\frac{1}{\sqrt{n}}$	$\frac{1}{\sqrt{n/2}}$	$\frac{1}{\sqrt{n/3}}$	$\frac{1}{\sqrt{n/5}}$	$\frac{1}{\sqrt{n/7}}$
0.95	Iter	16	11	8	6	4
	Temps	0.02	0.00	0.00	0.00	0.00
0.65	Iter	20	14	11	7	5
	Temps	0.00	0.00	0.00	0.00	0.00

Tableau 8*

Problème.9

μ	θ	$\frac{1}{\sqrt{n}}$	$\frac{1}{\sqrt{n/2}}$	$\frac{1}{\sqrt{n/3}}$	$\frac{1}{\sqrt{n/5}}$	$\frac{1}{\sqrt{n/7}}$
0.95	Iter	14	10	8	5	4
	Temps	0.01	0.00	0.00	0.00	0.00
0.85	Iter	13	9	7	5	4
	Temps	0.05	0.01	0.00	0.00	0.00
0.75	Iter	12	8	6	5	4
	Temps	0.00	0.00	0.00	0.00	0.00
0.65	Iter	3	3	2	2	2
	Temps	0.00	0.00	0.00	0.00	0.00

Tableau 9*

Problème.10

μ	θ	$\frac{1}{\sqrt{n}}$	$\frac{1}{\sqrt{n/2}}$	$\frac{1}{\sqrt{n/3}}$	$\frac{1}{\sqrt{n/5}}$	$\frac{1}{\sqrt{n/7}}$
0.95	Iter	18	12	10	7	6
	Temps	0.02	0.00	0.00	0.00	0.00
0.85	Iter	17	11	9	7	5
	Temps	0.01	0.01	0.00	0.00	0.00
0.75	Iter	15	10	8	6	5
	Temps	0.05	0.01	0.00	0.00	0.00
0.65	Iter	4	3	3	2	2
	Temps	0.00	0.00	0.00	0.00	0.00

Tableau 10*

Problème.11

μ	θ	$\frac{1}{\sqrt{n}}$	$\frac{1}{\sqrt{n/2}}$	$\frac{1}{\sqrt{n/3}}$	$\frac{1}{\sqrt{n/5}}$	$\frac{1}{\sqrt{n/7}}$
0.95	Iter	14	10	8	6	5
	Temps	0.00	0.00	0.00	0.00	0.00
0.65	Iter	18	12	10	7	6
	Temps	0.01	0.00	0.00	0.00	0.00

Tableau 11*

Problème.12

μ	θ	$\frac{1}{\sqrt{n}}$	$\frac{1}{\sqrt{n/2}}$	$\frac{1}{\sqrt{n/3}}$	$\frac{1}{\sqrt{n/5}}$	$\frac{1}{\sqrt{n/7}}$
0.95	Iter	14	10	7	5	4
	Temps	0.01	0.00	0.00	0.00	0.00
0.65	Iter	13	9	7	5	3
	Temps	0.00	0.00	0.00	0.00	0.00

Tableau 12*

Problème.13

μ	θ	$\frac{1}{\sqrt{n/25}}$	$\frac{1}{\sqrt{n/50}}$
0.95	Iter	32	20
	Temps	0.8521	0.5748
0.65	Iter	31	20
	Temps	0,8077	0.5336

Tableau 13*

4.4 Conclusion générale

Dans cette thèse, on a présenté une étude théorique et numérique d'une méthode de point intérieur de type primale-duale de trajectoire centrale, pour résoudre un programme convexe à contraintes linéaires. En effet, nous avons proposé de nouveaux paramètres notamment le paramètre barrière et le seuil qui mesure la taille du voisinage de la trajectoire centrale. A travers ces deux nouveaux paramètres un algorithme de TC primal-dual à pas de Newton complet bien défini est présenté et sa complexité est aussi calculée. L'efficacité numérique de cet algorithme est confirmée par des tests numériques. Les résultats obtenus sont encourageants, ce qui motivent à généraliser nos travaux à plusieurs problèmes d'optimisation tels que les problèmes de la programmation semi-définie, les problèmes de complémentarité linéaire standard et généralisé et pour les problèmes de complémentarité semi-défini. De plus notre travail dans le future est d'étudier la programmation convexe à contraintes linéaires sur les cônes symétriques notamment le cône de Lorentz.

Bibliography

- [1] M. Achache. A new primal-dual path-following method for Convex Quadratic Programming. *Comput. Appl. Math.* 25: 97 -110, (2006).
- [2] M. Achache. A weighted path-following method for the Linear Complementarity Problem. *Universitatis Babes. Bolyai. Series Informatica.* 49 (1): 61-73, (2004).
- [3] M. Achache. Multidimensional path-following interior point methods for linear programming and linear complementarity problems. *Thèse de Doctorat d'Etat. Université Ferhat Abbas - Sétif 1. Algérie,* (2005).
- [4] M. Achache. Complexity analysis and numerical implementation of a short-step primal-dual algorithm for linear complementarity problems. *Applied Mathematics and Computation.* (216): 1889–1895, (2010).
- [5] M. Achache and M. Goutali. A primal-dual interior point algorithm for Convex Quadratic Programs. *Studia Univ. Babes-Bolyai. Ser. Inform.* LVII (1): 48–58, (2012).
- [6] M. Achache and M. Goutali. Complexity analysis and numerical implementation of a full-Newton step interior-point algorithm for LCCO. *Numerical Algorithms.* 70 (2): 393–405, (2015).
- [7] L. Adler and R.D.C. Monteiro. Interior path-following primal-dual algorithms. Part **II**: convex quadratic programming. *Mathematical Programming.* 44: 43-66, (1989).

- [8] F. Alizadeh. Interior point methods in Semidefinite Programming with applications to Combinatorial Optimization. SIAM. J. OPTim. 5: 13-51, (1995).
- [9] E.D. Andersen. Solution of linear and convex optimization problems with emphasis on interior-point methods. CAMO.ICI. RO, (1996).
- [10] M.S. Bazaraa, H.D. Sherali and C.M. Shetty. Nonlinear Programming, Theory and Algorithms. John Wiley & Sons, New Yourk, NY, second ed, (1993).
- [11] A. Ben-Tal and A. Nemirovski. Lectures on Modern convex optimization. Analysis, Algorithms and Engineering Applications. Volume 2 of MPS-SIAM. Philadelphia. USA, (2000).
- [12] D. Benterki and Z. Kebbiche. A weighted path-following method for linearly constrained convex programming. Roumaine Math. Pures Appl. 57 (3): 245-256, (2012).
- [13] J.F. Bonnans, J.C. Gilbert, C. Le Maréchal, et C. Sogastizabal. Optimisation Numérique. Aspects théoriques et pratiques. Springer-Verlag Berlin Heidelberg, (1997).
- [14] J. M. Borwein and A. S. Lewis. Convex Analysis and Noulnear Optimization. Canadian Mathematical Society, (2006).
- [15] M. Bouafia. Étude asymptotique des méthodes de points intérieurs pour la programmation linéaire. Thèse de Doctorat, Université du Havre, (2016).
- [16] S. Boyd and L. Vandenberghe. Convex Optimization. Cambridge University Press, (2009).
- [17] Z. Y. Cheng and J. E. Mitchell. A Primal-Dual Interior-Point Method for Linear Programming based on a weighted Barrier Function. Optimization Theory and Applications. 87 (2): 301-321, (1995).
- [18] C. Culioli. Introduction à l'optimisation. Edition Marketing, (1994).

- [19] Z. Darvay. New interior-point algorithms in linear optimization. [J]. *Advanced Modeling and Optimization*. 5 (1): 51-92, (2003).
- [20] Z. Darvay. A predictor-corrector algorithm for linearly constrained convex optimization. *Studia Univ. Babes-Bolyai. Ser. Inform. LIV* (2): 121–138, (2009).
- [21] Z. Darvay. A weighted-path-following method for linear optimization, *Studia Universitatis Babes-Bolyai. Series Informatica*. 47 (1): 3-12, (2002).
- [22] M. El Ghami. *New Primal-dual Interior-Point Methods Based on Kernel Functions*. Phd thesis. Delft University. Netherlands, (2005).
- [23] M. El Ghami, I. D. Ivanov, C. Roos and T. Steihaug. A polynomial-time algorithm for LO based on generalized logarithmic barrier functions. *International Journal of Applied Mathematics*. 21 (1): 99–115, (2008).
- [24] K.R. Frisch. *The logarithmic potential method of convex programming*. Technical report. University Institute of Economics, Oslo, Norway, (1955).
- [25] J.C. Gilbert. *El'ements d'Optimisation Différentiable. Théorie et Algorithmes*. INRIA Rocquencourt, (2002).
- [26] P.E. Gill, W. Murray, D.B. Pongceleon and M.A. Saunders. Primal-dual methods for linear programming. *Mathematical Programming*. 70 (3): 251-277, (1995).
- [27] F. Glineur. INMA2471. *Méthodes et Modèles d'Optimisation*. Notes de cours. Université Catholique de Louvain, (2006).
- [28] M. D. Gonzalez-Lima and C. Roos. On Central-Path Proximity Measures in Interior-Point Methods. *Journal of Optimisation. Theory and Applications*. Vol 127 (2): 303-328, (2005).
- [29] L.M. Grana Drummond and B.F. Svaiter. On well definiteness of the central path. *J. Optim. Theory Appl.* 102 (2): 223–237, (1999).

- [30] N. W. Henderson. Arc search Methods for Linearly Constrained Optimization. Doctoral Thesis. Stanford University, (2012).
- [31] J.L. Jardin. Analyse, Algorithmes et Programmes en Pascal. Dunod Paris, (1989).
- [32] F. Jarre. On the convergence of the method of analytic centers when applied to convex quadratic problems. *Mathematical Programming*. 49 (3): 341-358, (1991).
- [33] A. Kadiri et A. Yassine. Une procédure de purification pour les problèmes de complémentarité linéaire monotones. *RAIRO Operations Research*. 38 (1): 63-83, (2004).
- [34] P. K. Kaluarachchi. Full-Newton-step interior-point method for the linear Complementarity Problems Master of Science. Georgia Southern University, (2011).
- [35] N. K. Karmarkar. A new Polynomial time Algorithm for Linear Programming. *Combinatorica* 4 (4): 373-395, (1984).
- [36] Z. Kebbiche. Etude et extensions d'algorithmes de points intérieurs pour la programmation non linéaire. Thèse de doctorat d'état, Université Ferhat Abbas - Sétif 1, (2007).
- [37] A. Keraghel. Analyse convexe. Théorie fondamentale et exercices. Laboratoire de Mathématique Fondamentale et Numérique. Université de Sétif 1, (2001).
- [38] A. Keraghel. Etude adaptative et comparative des principales variantes dans l'algorithme de Karmarkar. Thèse de Doctorat, Université Joseph Fourier Grenoble, (1989).
- [39] S. Kettab. Généralisation d'une méthode de trajectoire centrale de points intérieurs pour la programmation semi-définie. Thèse de doctorat, Université Ferhat Abbas - Setif 1, (2014).
- [40] L. Mahgoub and Y. Mohamed. Interior-Point-Methods for Solving Linear Programming Problems. University of the Western Cape Town. South Africa, (2005).

- [41] T. Messaoudi. Etude qualitative et comparative de quelques méthodes de points intérieurs pour la résolution d'un programme quadratique convexe. Thèse de Magister. Université Ferhat abbas - Sétif 1, (2001).
- [42] M. Minoux. Programmation Mathématique. Théorie et Algorithmes, (1990).
- [43] Y. E. Nesterov and A. Nemirovskii. Interior point polynomial methods in convex programming. SIAM, Philadelphia, (1994).
- [44] J. Peng, C. Roos and T. Terlaky. New Complexity Analysis of the Primal-Dual Newton Method for Linear Optimization. *Annals of Operations Research*. **99** (4): 23-39, (2000).
- [45] J. Peng, C. Roos and T. Terlaky. A new class of polynomial primal-dual methods for linear and semi definite optimization. *European Journal of Operational Research-Elsevier*, (2002).
- [46] J. Peng, C. Roos and T. Terlaky. Self-Regular Proximities and New Search Directions for Linear and Semidefinite optimization. *Mathematical Programming*, Springer, (2002).
- [47] R. T Rockafellar. *Convex Analysis*. Princeton University Press, New Jersey, (1970).
- [48] O. Romanko. An Interior Point Approach to Quadratic and Parametric Quadratic Optimization. Master of Science McMaster, (2004).
- [49] C. Roos, T. Terlaky, and J. Ph. Vial . Theory and algorithms for linear optimization. An interior point approach. John-Wiley. Sons, Chichester, UK, (1997).
- [50] H. Roumili. Méthodes des points intérieurs non réalisables en optimisation. Théorie, Algorithme et Applications. Thèse de Doctorat, Université Ferhat abbas - Sétif 1, (2007).

- [51] P. Segala. Méthodes de points intérieurs et de quasi-Newton. Thèse de Doctorat. Université de Limoges, France, (2002).
- [52] T. Terlaky. Interior Point Methods of Mathematical Programming. Applied Optimization series. Kluwer Academic Publishers, (1996).
- [53] S.J. Wright. Primal-dual interior point methods. Copyright by SIAM, (1997).
- [54] Y. Ye. Interior point algorithms. Theory and Analysis. John-Wiley Sons, Chichester, (1997).
- [55] M. Zhang, Y.Q. Bai and G. Wang. A new primal-dual path-following interior-point algorithm for Linearly Constrained Convex Optimization. J. Shanghai University. 12 (6): 475–780, (2008).

ملخص

في هذه الأطروحة، نختتم بالدراسة النظرية (لا سيما التعقيد الحدودي) و بالمعالجة العددية لطريقة النقاط الداخلية ذات المسار المركزي من نوع الأولي-المرادف لحل مسائل البرمجة المحدبة غير الخطية بقيود خطية. في دراستنا هذه نقترح معاملين جديدين؛ معامل الحاجز ومعامل العتبة الذي يقيس طول جوار المسار المركزي. حيث، يتم تقديم خوارزمية أولية-مرادفة ذات تكرارة نيوتن الكاملة و بخطوة صغيرة معرفة بشكل جيد ويتم حساب تعقيدها. ولتأكيد الكفاءة العددية لهذه الخوارزمية قمنا بتجارب عددية وهي مشجعة.

الكلمات المفتاحية: البرمجة المحدبة بقيود خطية، البرمجة التربيعية المحدبة، البرمجة الخطية؛ طرق النقاط الداخلية، خوارزميات المسار المركزي من نوع الأولي-المرادف، تعقيد الخوارزمي.

Résumé

Dans cette thèse, on s'intéresse à l'étude théorique (notamment la complexité polynomiale) et à l'implémentation numérique d'une méthode de points intérieurs de trajectoire centrale de type primal-dual pour résoudre les problèmes de la programmation non linéaire convexe à contraintes linéaires. Dans notre étude, nous proposons de nouveaux paramètres qui décrivent le paramètre barrière et le seuil qui mesure le voisinage de la trajectoire centrale. A travers ces derniers, un algorithme primal-dual à pas court et d'itération de Newton complet bien défini est présenté et de plus sa complexité est calculée. L'efficacité numérique de cet algorithme est confirmée par des tests numériques qui sont encouragés.

Mots clés: Programmation Convexe a contraintes linéaires, Programmation Quadratique Convexe, Programmation linéaire; Méthode de Points Intérieurs, Algorithmes de trajectoire centrale de type primal-dual, Complexité des Algorithmes.

Abstract

In this thesis, we are interested in the theoretical study (in particular the polynomial complexity) and in the numerical implementation of a primal-dual path-following method to solve the linearly constrained convex optimization. In our study, we propose new parameters which describe the barrier parameter and the threshold which measures the neighborhood of the central path. Through this latter, a well-defined short-step with a full-Newton primal-dual algorithm is presented and its complexity is computed. The numerical efficiency of this algorithm is confirmed by some numerical tests, and they are very encouraging.

Keywords: Linearly Constrained Convex Programming, Convex quadratic programming, Linear Programming, Interior Point Methods, Primal-dual Algorithms, Complexity of Algorithms.