

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université 8 Mai 1945 Guelma

Faculté des Mathématiques et de l'Informatique  
et des Sciences de la Matière  
Département de Mathématiques



## Mémoire de Master en Mathématiques

Option : EDP et Analyse Numérique

Présenté par :

Mr. Malek Abderraouf

M<sup>elle</sup>. Debbar Khouloud

**Intitulé**

**Etude théorique sur les fonctions noyau pour  
la programmation linéaire**

Dirigé par :

Mr. BOUAFIA Mousaab

Devant les jurys :

PRESIDENT

Dr. LAKHAL Fahim

MCA

Univ-Guelma

RAPPORTEUR

Dr. BOUAFIA Mousaab

MCA

Univ-Guelma

EXAMINATRICE

Dr. MELLAL Roumaissa

MCB

Univ-Guelma

Session Juillet 2019

# Etude théorique sur les fonctions noyau pour la programmation linéaire

Présenté par : Malek Abderraouf et Debbar Khouloud

Encadrées par : Dr. Bouafia Moussaab

30 juin 2019



## Remerciement :

Aujourd'hui, après des années d'attente et de patience, nous sommes au point de terminer une étape majeure de notre parcours d'étude, c'était une des expériences les plus merveilleuses, les plus belles que nous avons vécues, remplie d'amour de science, d'apprentissage et de récolte des connaissances.

Nous sommes très déçus d'avoir terminé cette période avant de prendre plus d'une gorgée de la mer des mathématiques. Avant de commencer la présentation de ce mémoire, nous profitons l'occasion pour remercier du fond du cœur toutes les personnes qui nous ont aidés dans la réalisation de ce mémoire de master mathématique, spécialité EDP et Analyse numérique.

En premiers lieux, nous remercions tous les professeurs de mathématique de l'université de 08 mai Guelma pour les solides notions théoriques qu'ils nous ont enseignés pendant toutes ces années et sur lesquelles nous nous sommes appuyés pour élaborer ce mémoire.

Tout d'abord grands remerciements à monsieur BOUAFIA Mousaab, tuteur de ce mémoire, pour son aide précieuse et pour son temps qu'il a bien voulu nous consacrer. Nous adressons aussi nos sincères remerciements aux membres de jurés monsieur LEKHEL Fahim le président des jurés et l'examinatrice M.MELLAL Roumaïssa, qui ont accepté la lourde tâche de lire, commenter et juger ce mémoire.

Nos vifs remerciements vont également à tous les professeurs qui nous ont enseigné qui par leurs compétences nous ont soutenus dans la poursuite de nos études.

Nos derniers remerciements, et pas les moindres vont à nos parents et familles qui nous ont tant apporté d'amour, d'encouragement, sans eux nous n'aurons pas pu accomplir tels succès. Finalement nous remercions tous ceux qui nous aident à réaliser ce travail de près ou de loin.





## Dédicace :

Malek H. Raouf

*A mes chers parents, pour tous leurs sacrifices, amour, tendresse, soutien et leurs prières tout au long de mes études.*

*A mes chères sœurs pour leurs encouragements permanents, et soutien moral.*

*A mes chers frères Saïd et Khalid pour leurs appui et encouragement.*

*A ma future femme pour leur support continue.*

*A toute ma famille pour leur soutien tout au long de mon parcours universitaire,*

*Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit de votre soutien infailible,*

*Merci d'être toujours là pour moi.*





## Dédicace :

Khouloud Debbar

*C'est avec un grand plaisir que je dédie ce travail à :*

*Mes parents, qui ont consacré leur existence à bâtir la mienne, quoi que je dise ou que je fasse, je ne saurais point vous remercier comme il se doit. Votre affection me couvre, votre bienveillance me guide et votre présence à mes côtés a toujours été ma source de force pour affronter les différents obstacles. Que ce travail traduit ma gratitude et mon affection.*

*Mes très chers frères Antara, Aïssa, Bilal et Mohamed Elkhalifa et mes plus belles sœurs Asma, Zineb et Hadjer avec qui j'ai partagé les plus beaux moments, pour leur soutien infinis et leurs aide incessantes, puisse Allah vous donne santé, bonheur, courage et surtout réussite.*

*Ma cousine Wiam et mes amies de toujours: Meryem, Souha, Sarra et Nour et Houda, pour leur sincère et profonde amitié, leur confiance, leur encouragement et pour les moments agréables que nous avons passés ensemble. Et j'espère conserver à jamais les souvenirs et les lies qui nous unissent.*

*Mon binôme et sa famille.*

*Tous mes enseignants qui ont contribué à ma formation.*

*Tous ceux qui me sont chers et que j'ai omis de citer.*



## Résumé :

Dans ce mémoire nous présentons une étude théorique et algorithmique sur la méthode de trajectoire centrale via une fonction noyau qui résout les problèmes de la programmation linéaire (PL). Cette méthode est un prolongement de la méthode de trajectoire centrale classique.

Les avantages de notre méthode est d'assurer la convergence théorique et pratique, mais il reste de trouver un point de départ à l'intérieur et au voisinage de la trajectoire centrale est difficile.

## Mots Clés :

Programmation Linéaire, Méthode de Points Intérieurs, Fonctions Noyaux, Complexité algorithmique, Méthode de Trajectoire Centrale.

## ملخص :

من خلال هذه المذكرة نقدم دراسة نظرية و خوارزمية حول طريقة المسار المركزي المعتمدة على دالة النواة لحل مشاكل البرمجة الخطية, هذه الطريقة هي تمديد لطريقة المسار المركزي الكلاسيكية. من محاسن هذه الطريقة ضمان التقارب النظري والعملي, الا انه يبقى ايجاد نقطة بداية في الداخلية و بجوار المسار المركزي امر صعب.

## كلمات مفتاحية :

البرمجة الخطية, طريقة النقطة الداخلية, دالة النواة, التعقد الخوارزمي, طريقة المسار المركزي .

# Table des matières

<b>Introduction</b>	<b>3</b>
<b>1 Notions de bases sur l'analyse convexe</b>	<b>5</b>
1.1 Élément d'analyse convexe . . . . .	5
1.1.1 Fonction convexe différentiable . . . . .	8
1.2 Programmation mathématiques . . . . .	9
1.2.1 Classification et résolution d'un programme mathématique . . . . .	10
1.2.2 Existence et Unicité de solution . . . . .	10
1.2.3 Dualité Lagrangienne : . . . . .	11
1.2.4 Programmation linéaire (PL) . . . . .	12
1.3 Méthodes barrières . . . . .	15
1.4 Algorithme d'optimisation . . . . .	17
1.4.1 Description . . . . .	17
1.4.2 Notations asymptotiques . . . . .	18
<b>2 Fonction noyau et sa qualification</b>	<b>19</b>
2.1 Fonction noyau . . . . .	19
2.2 Qualification d'une fonction noyau . . . . .	22
2.3 Propriétés et relation entre les conditions de qualification . . . . .	22

<b>3</b>	<b>Méthode de Trajectoire Centrale via une fonction noyau</b>	<b>29</b>
3.1	Présentation de la méthode . . . . .	29
3.2	Générique Primal-dual IPMs pour l'optimisation linéaire . . . . .	33
3.3	Etude de la convergence de la méthode de Trajectoire Centrale via une fonction noyau . . . . .	35
3.3.1	Borne supérieure de $\Phi(v)$ pour chaque itération externe . . . . .	35
3.4	Analyse de la décroissance de la fonction barrière de proximité $\Phi$ . . . . .	36
3.4.1	Analyse de la complexité algorithmique de l'algorithme . . . . .	39
	<b>Conclusion</b>	<b>43</b>
	<b>Bibliographie</b>	<b>44</b>

# Introduction

La programmation mathématique se propose pour objet l'étude théorique des problèmes d'optimisation ainsi que la conception et la mise en œuvre des algorithmes de résolution qui utilisent une fonction à plusieurs variables. On distingue deux types de programmation mathématique linéaire et non-linéaire (fonctions et de contraintes linéaire, non-linéaire).

En 1947 George Dantzig [10] a créé une méthode de résolution pour la programmation linéaire s'appelle l'algorithme du simplexe. mais théoriquement la méthode n'a pas autant de succès, elle est plutôt jugée inefficace de par sa complexité arithmétique exponentielle qui est de l'ordre de  $O(2^n)$  opérations.

Cette méthode a connu depuis lors de nombreuses améliorations, elle est développées par plusieurs chercheurs jusqu'à l'application de la méthode de l'ellipsoïde par L.G.Kachian en 1978 [20] et il prouve qu'elle est de complexité polynomiale.

En 1984 Narendra Karmarkar [13] a découvert des méthodes de point intérieurs qui résout les problèmes d'optimisation linéaire en temps polynomiale concurrente a celle de l'ellipsoïde mais plus efficace. L'idée de base de cette méthode est d'utiliser des fonctions barrières qui sont convexes par définition du problème. à l'opposé de l'algorithme du simplexe, cette méthode atteint l'optimum du problème en passant par l'intérieur de l'ensemble des solutions réalisables.

Parmi leurs avantages est l'efficacité théorique et pratique, traitement de très grands problèmes en un temps acceptable et l'adaptation au cas non linéaire.

Depuis l'annonce de Karmarkar, plus de 3000 articles de recherche portant sur les

méthodes de point intérieur ont été publiés par la communauté scientifique en quelques années, Et a donné comme résultat une grande variété d'algorithmes de ce type parmi eux **la méthode de trajectoire centrale (TC)**, qu'elle a été introduite au début des années 90. Cette méthode est basée sur le suivi d'une trajectoire centrale

Elles possèdent de bonnes propriétés théoriques, une complexité polynômiale. Les algorithmes de trajectoire centrale restreintes les itérés à un voisinage de la trajectoire centrale, cette dernière est une courbe de points strictement réalisables.

L'objectif de ce mémoire est de proposer une étude théorique et algorithmique sur les fonctions noyau pour la programmation linéaire

Ce mémoire est reparti en trois chapitres :

Le premier contient un rappel introductif sur l'analyse convexe et quelques définitions de base, et il rassemble toutes les notions et résultats que nous utiliserons par la suite. Le deuxième chapitre concerne les fonctions noyaux, leurs propriétés et leurs qualifications. Et le dernier chapitre est consacré à l'étude théorique pour résoudre un programme linéaire primal-dual par les méthodes de points intérieurs basées sur la méthode de trajectoire centrale, Où nous avons injecté des fonctions noyau, en outre, nous formulons un algorithme de points intérieurs primal-dual pour (*IPMs*) en utilisant la fonction de proximité et de donner son analyse de complexité. De plus, la complexité obtenue par l'algorithme de long pas et petit pas. Finalement, on finira le mémoire par une conclusion et perspectives.

# Chapitre 1

## Notions de bases sur l'analyse convexe

Dans ce chapitre, nous rappelons rapidement certaines propriétés de l'analyse convexe, et quelques notions fondamentales de la programmation linéaire et l'étude asymptotique de la convergence d'un algorithme d'optimisation.

Ces notions sont utiles pour démontrer les résultats théoriques dans les chapitres suivants.

### 1.1 Élément d'analyse convexe

La notion de convexité prend deux formes : un ensemble convexe, et une fonction convexe.

**Definition 1** • *Un objet géométrique est dit convexe lorsque, chaque fois qu'on y prend deux points  $x$  et  $y$ , le segment  $[x, y]$  qui les joint y est entièrement contenu. Ainsi un cube plein, un disque ou une boule sont convexes, mais un objet creux ou bosselé ne l'est pas. On dit qu'un ensemble  $D$  de  $\mathbb{R}^n$  est convexe c'est-à-dire : le segment de droite reliant*

toute paire de points  $x$  et  $y$  appartenant à  $D$  est entièrement contenu dans  $D$ .

$$\lambda x + (1 - \lambda) y \in D, \forall x, y \in \mathbb{C}, \forall \lambda \in [0, 1] \quad (1.1.1)$$

• Un ensemble  $D$  est dit convexe affine si :

$$\lambda x + (1 - \lambda) y \in D, \forall x, y \in D, \forall \lambda \in \mathbb{R} \quad (1.1.2)$$

• Un polyèdre est une forme géométrique à trois dimensions ayant des faces planes polygonales qui se rencontrent selon des segments de droite qu'on appelle arêtes.

On dit que  $D$  est un polyèdre convexe si tout point de tout segment joignant deux points quelconques du polyèdre appartient au polyèdre. Autrement dit, un polyèdre est convexe si toutes ses diagonales sont entièrement contenues dans son intérieur. Et il est de la forme suivante :

$$D = \{x \in \mathbb{R}^n : A_i^t \leq b_i, i = 1, \dots, m\} \quad (1.1.3)$$

où  $A_i$  est un vecteur non nul de  $\mathbb{R}^n$  et  $b_i$  un scalaire pour  $i = 1, \dots, m$ .

$D$  peut s'écrire aussi sous la forme matricielle suivante :

$$D = \{x \in \mathbb{R}^n / Ax \leq b\} \quad (1.1.4)$$

où  $A$  est une matrice de  $\mathbb{R}^{m \times n}$  et  $b$  un vecteur de  $\mathbb{R}^m$ .

•  $S_n$  est un  $n$ -simplexe ( un simplexe ou  $n$ -simplexe est l'analogie à  $n$  dimensions du triangle ) s'il est de la forme :

$$S_n = \left\{ x \in \mathbb{R}_+^n : \sum_{i=1}^n x_i = 1 \right\} \quad (1.1.5)$$

Un point  $x \in S_n$  est dit extrémal (ou sommet de  $S_n$ ) si l'on a :

$$\forall t \in [0, 1], \forall (y, z) \in S_n^2 : x = (1 - t) y + tz \Rightarrow x = y = z \quad (1.1.6)$$

- Une fonction réelle d'une variable réelle est dite convexe si :
  - quels que soient deux points  $x$  et  $y$  du graphe de la fonction, le segment  $[x, y]$  est entièrement situé au-dessus du graphe, ou
  - l'épigraphe de la fonction (l'ensemble des points qui sont au-dessus de son graphe) est un ensemble convexe, ou
  - vu d'en dessous, le graphe de la fonction est en bosse.

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

★  $f$  est dite convexe sur  $D$  si :

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y), \forall \lambda \in [0, 1], \forall x, y \in D \quad (1.1.7)$$

★  $f$  est strictement convexe sur  $D$  si :

$$f(\lambda x + (1 - \lambda)y) < \lambda f(x) + (1 - \lambda)f(y), \forall \lambda \in ]0, 1[, \forall x, y \in D \text{ et } x \neq y \quad (1.1.8)$$

★  $f$  est fortement convexe ( $\alpha$ -convexe) sur  $D$  si :

$$\exists \alpha > 0, \forall \lambda \in ]0, 1[, \forall x, y \in D \text{ et } x \neq y, \quad (1.1.9)$$

tel que

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y) - \frac{1}{2}\alpha\lambda(1 - \lambda)\|x - y\|^2 \quad (1.1.10)$$

★  $f$  est mid-convexe sur  $D$  si :

$$f\left(\frac{x + y}{2}\right) \leq \frac{f(x) + f(y)}{2}, \forall x, y \in D \quad (1.1.11)$$

★  $f$  est quasi-convexe sur  $D$  si :

$$f(\lambda x + (1 - \lambda)y) \leq \max(f(x), f(y)), \forall \lambda \in [0, 1], \forall x, y \in D \quad (1.1.12)$$

**Remarque 1.1.1** Si  $f$  est une fonction continue sur un convexe  $D$ , on a :

- $f$  est convexe sur  $D \Leftrightarrow f$  est mid-convexe  $D$ .
- $f$  est  $\alpha$ -convexe sur  $D$  équivaut à :

$$\forall x, y \in D, f\left(\frac{x+y}{2}\right) \leq \frac{f(x) + f(y)}{2} - \frac{1}{8}\alpha\|x - y\|^2 \quad (1.1.13)$$

### 1.1.1 Fonction convexe différentiable

Les deuxièmes dérivées peuvent servir à déterminer la convexité ou la concavité d'une fonction.

**Definition 2** Soit  $f \in C^1(D)$ , une fonction continûment différentiable sur un domaine convexe  $D$ . Alors on a les équivalences suivantes :

$f$  est convexe si et seulement si :

$$f(y) - f(x) \geq \langle \nabla f(x), y - x \rangle, \forall x, y \in D \quad (1.1.14)$$

où bien

$$\langle \nabla f(x) - \nabla f(y), y - x \rangle \geq 0, \forall x, y \in D \quad (1.1.15)$$

De plus,  $f$  est dite strictement convexe si l'une ou l'autre des inégalités précédentes sont strictes pour  $x \neq y$ .

$f$  est fortement convexe si et seulement s'il existe  $\alpha > 0$ , tel que :

$$f(y) - f(x) \geq \langle \nabla f(x), y - x \rangle + \frac{\alpha}{2}\|y - x\|^2, \forall x, y \in D \quad (1.1.16)$$

Si  $f \in C^2(D)$ , alors :

$f$  est une fonction convexe sur  $D$  si et seulement si la matrice Hessienne est semi-définie positive, (c'est à dire que  $y^t \nabla^2 f(x) y \geq 0, \forall x, y \in D$  où encore toutes les valeurs propres de  $\nabla^2 f(x)$  sont positives).

De même, si la matrice Hessienne est définie positive, (c'est à dire que  $y^t \nabla^2 f(x) y > 0, \forall x, y \in D$  et  $y \neq 0$  où encore toutes les valeurs propres de  $\nabla^2 f(x)$  sont positives), alors  $f$  est une fonction strictement convexe sur  $D$

## Formules de Taylor

La formule de Taylor, du nom du mathématicien Brook Taylor qui l'établit en 1712, permet l'approximation d'une fonction  $n$  fois dérivable au voisinage d'un point, par un polynôme dont les coefficients dépendent uniquement des dérivées de la fonction en ce point.

**Théorème 1.1.1 (Taylor-Young)** *Supposons que  $f$  soit de classe  $C^n$  sur  $I$ . Alors, pour tout  $h \in \mathbb{R}$  tel que  $x_0 + h$  appartienne à  $I$  on peut écrire :*

$$\begin{aligned} f(x_0 + h) &= f(x_0) + hf'(x_0) + \frac{h^2}{2!} f^{(2)}(x_0) + \dots + \frac{h^n}{n!} f^{(n)}(x_0) + h^n \epsilon(h) \\ &= \sum_{k=0}^n \frac{h^k}{k!} f^{(k)}(x_0) + h^n \epsilon(h) \end{aligned} \tag{1.1.17}$$

Où  $\epsilon(h)$  est une fonction qui tend vers 0 quand  $h$  tend vers 0.

## 1.2 Programmation mathématiques

La programmation mathématique, et plus particulièrement l'optimisation vise à résoudre des problèmes où l'on cherche à déterminer parmi un grand nombre de solutions candidates celle qui donne le meilleur rendement. Plus précisément, on cherche à trouver une solution satisfaisant un ensemble de contraintes qui minimise (maximise) une fonction

donnée. L'application de la programmation mathématique est en expansion croissante et se retrouve dans plusieurs domaines.

### 1.2.1 Classification et résolution d'un programme mathématique

Un programme mathématique ( $P$ ) est un problème d'optimisation avec contraintes de la forme :

$$(P) \quad \begin{cases} \min f(x) \\ x \in D \end{cases} \quad \text{où } D = \begin{cases} x \in \Omega \subseteq \mathbb{R}^n / g_i(x) \leq 0, \quad i = 1, \dots, m \\ h_j(x) = 0, \quad j = 1, \dots, p \end{cases}$$

où  $f, g_i, h_j$  sont des fonctions définies de  $\mathbb{R}^n$  dans  $\mathbb{R}$ .

On appelle  $f$  la fonction objectif et  $D$  l'ensemble des solutions réalisables ou l'ensemble des contraintes.

La classification de ( $P$ ) et son traitement numérique sont établis à partir des propriétés fondamentaux des fonctions  $f, g_i, h_j$  à savoir la convexité, la différentiabilité et la linéarité.

Parmi les cas particuliers les plus étudiés on note :

- La programmation linéaire ( $f$  linéaire,  $g_i, h_j$  affines).
- La programmation convexe ( $f, g_i$  et  $D$  convexes,  $h_j$  affines).
- La programmation en nombres entiers ( $D$  est un ensemble discret, c'est à dire les variables sont entières).

### 1.2.2 Existence et Unicité de solution

**Théorème 1.2.1 (Weierstrass) [19]** *Si  $D$  est compact non vide de  $\mathbb{R}^n$  et si  $f$  est continue sur  $D$  alors ( $PM$ ) admet au moins une solution optimale globale  $x^* \in D$ .*

**Théorème 1.2.2 [19]** *Si  $D$  est fermé non vide de  $\mathbb{R}^n$ ,  $f$  est continue et coercive sur  $D$*

(c'est-à-dire  $\lim_{\|x\| \rightarrow +\infty} f(x) = +\infty$ ) alors (PM) admet au moins une solution optimale globale.

**Théorème 1.2.3 (d'existence [3])** Si  $D$  est convexe non vide de  $\mathbb{R}^n$ ,  $f$  est strictement convexe sur  $D$  alors (PM) admet une solution optimale au plus.

**Démonstration :** Dans le cas où il existe une solution, on va démontrer qu'elle est unique c'est pour ça on suppose qu'il existe deux solutions différentes  $x_1$  et  $x_2$  minimum de la fonction  $f$ , où  $f$  est strictement convexe alors  $\forall \lambda \in ]0, 1[$  on a :

$$f(\lambda x_1 + (1 - \lambda)x_2) < \lambda f(x_1) + (1 - \lambda)f(x_2) \quad (1.2.1)$$

mais  $f(x_1) = f(x_2)$  alors :

$$f(\lambda x_1 + (1 - \lambda)x_2) < f(x_1) \quad (1.2.2)$$

Contradiction avec la supposition, alors on déduit que s'il existe une solution elle est unique.

■

### 1.2.3 Dualité Lagrangienne :

Les programmes fractionnaires convexes consistent en la maximisation d'un objectif rapport d'une fonction convexe soumis à des contraintes convexes. Pour ces modèles à objectif non linéaire et non convexe, une nouvelle décomposition Lagrangienne basée sur une réécriture fractionnaire des contraintes, est proposée. Elle combine la résolution d'un programme en variables 0 – 1 à objectif linéaire sous les contraintes initiales et celle d'un programme fractionnaire sans contraintes. Lorsque les contraintes sont linéaires, le résultat de dominance de la décomposition Lagrangienne sur la relaxation Lagrangienne est prouvé.

On considère le problème primal suivant :

$$m = \inf_x [f(x), x \in S] \quad (1.2.3)$$

Avec

$$S = \{x \in D \subset \mathbb{R}^n : g_i(x) \leq 0, i = 1, \dots, k, h_j(x) = 0, j = 1, \dots, m\} \quad (1.2.4)$$

Le Lagrangien associé à ce problème est la fonction  $L : D \times [0, +\infty[^k \times \mathbb{R}^m \longrightarrow \mathbb{R}$ , définie par :

$$L(x, \lambda, \mu) = f(x) + \sum_{i=1}^k \lambda_i g_i(x) + \sum_{j=1}^m \mu_j h_j(x) \quad (1.2.5)$$

On pose :

$$\alpha(x) = \sup_{\lambda, \mu} [L(x, \lambda, \mu) : \lambda \geq 0] = \begin{cases} f(x) \text{ si } g_i(x) \leq 0 \text{ et } h_j(x) = 0 \\ +\infty \text{ sinon} \end{cases} \quad (1.2.6)$$

Et on prend

$$\bar{\alpha} = \inf_{x \in D} \alpha(x) = \inf_x [f(x) : x \in S] \quad (1.2.7)$$

Le problème dual associé au problème primal est :

$$\bar{\beta} = \sup_{(\lambda, \mu)} \inf_{x \in D} [L(x, \lambda, \mu)] \quad (1.2.8)$$

on a l'inégalité de dualité :

$$-\infty \leq \bar{\beta} \leq \bar{\alpha} \quad (1.2.9)$$

## 1.2.4 Programmation linéaire (PL)

La programmation linéaire dans  $\mathbb{R}^n$  est constituée de la famille de programmes qui traitent la résolution des problèmes d'optimisation pour lesquels la fonction objectif et

toutes les contraintes sont linéaires. Il s'agit des problèmes d'optimisation les plus célèbres, les plus simples théoriquement et les plus étudiés en recherche opérationnelle.

**Definition 3** *On appelle un programme linéaire tout programme mathématique, où la fonction objectif est linéaire et l'ensemble des contraintes est affine.*

*Il existe trois formes pour écrire un programme linéaire qui sont : la forme canonique, la forme standard et la forme générale.*

On s'intéresse au problème  $(P)$  sous forme standard suivante :

$$(P) \begin{cases} \min c^t x \\ s.c \\ Ax = b \\ x \geq 0 \end{cases} \quad (1.2.10)$$

tel que  $A$  est une matrice réelle de type  $(m, n)$  supposée de plein rang (c'est-à-dire :  $\text{Rang}(A) = m < n$ ),  $b$  un vecteur de  $\mathbb{R}^n$ .

Bien évidemment, n'importe quel  $(P)$  se ramène facilement à cette forme. Le dual  $(D)$  de  $(P)$  est un programme linéaire défini par :

$$(D) \begin{cases} \max c^t x \\ s.c \\ A^t y + s = c \\ s \geq 0, s \in \mathbb{R}^n \\ y \in \mathbb{R}^m \end{cases} \quad (1.2.11)$$

on notera par la suite :

$$F_{(P)} = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\} \quad (1.2.12)$$

Ensemble des solutions primales réalisables de  $(P)$ .

- Un vecteur  $x \in F_{(P)}$  est appelé solution réalisable de  $(P)$ .
- Un vecteur  $x^* \in F_{(P)}$  minimisant la fonction objectif de  $(P)$  s'appelle solution optimale de  $(P)$ .
- Un programme linéaire  $(P)$  réalisable est borné si la fonction objectif est bornée sur  $F_{(P)}$ .

$$\hat{F}_{(P)} = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\} \quad (1.2.13)$$

Ensemble des solutions primales strictement réalisables de  $(P)$ .

On note par la suite l'ensemble des solutions duales réalisables de  $(D)$  par :

$$F_{(D)} = \{y \in \mathbb{R}^m : A^t y + s = c, s \geq 0\} \quad (1.2.14)$$

- Un vecteur  $y^* \in F_{(D)}$  maximisant la fonction objectif de  $(D)$  s'appelle solution optimale de  $(D)$ .

**Remarque 1.2.1** • *Si l'un des problèmes  $(P)$  et  $(D)$  admet une solution optimale, il en est de même pour l'autre, et leurs valeurs optimales correspondantes sont égales.*

- • *Si l'un des problèmes a une valeur optimale non finie, l'autre n'a pas de solution optimale.*

**Théorème 1.2.4 (Dualité faible)** *Si  $x$  et  $(y, s)$  sont respectivement des solutions réalisables pour  $(P)$  et  $(D)$  alors,  $c^t x \geq b^t y$ .*

**Théorème 1.2.5 (Dualité forte)** *Si  $\bar{x}$  et  $(\bar{y}, \bar{s})$  sont respectivement des solutions réalisables correspondant une valeur optimale finie pour  $(P)$  et  $(D)$  tel que :*

$$c^t \bar{x} = b^t \bar{y} \quad (1.2.15)$$

*alors  $\bar{x}$  est une solution primale optimale de  $(P)$  et  $\bar{y}$  est une solution duale optimale de  $(D)$ .*

**Remarque 1.2.2** *On peut déduire facilement que si  $\bar{x}$  et  $(\bar{y}, \bar{s})$  sont respectivement des solutions réalisables de (P) et (D), alors on a la propriété suivante :*

$$c^t \bar{x} = b^t \bar{y} \Leftrightarrow \bar{x} \bar{s} = 0 \Leftrightarrow \bar{x}^t \bar{s} = 0 \quad (1.2.16)$$

## 1.3 Méthodes barrières

Nous considérons le problème

$$\min f(x) \quad \text{sous contraintes} \quad x \in D, g(x) \leq 0 \quad (1.3.1)$$

où  $f : \mathbb{R}^n \rightarrow \mathbb{R}, g : \mathbb{R}^n \rightarrow \mathbb{R}^m$  et  $D$  est un ensemble fermé. L'ensemble des points

admissibles est

$$F = \{x \in \mathbb{R}^n \mid x \in D, g(x) \leq 0\} \quad (1.3.2)$$

Dans ce cadre-ci, nous appellerons l'ensemble des points intérieurs l'ensemble  $S$  défini par

$$S = \{x \in \mathbb{R}^n \mid x \in D, g(x) < 0\} \quad (1.3.3)$$

Noter qu'il s'agit d'un abus de langage, et que les points ne sont intérieurs que lorsque l'on se restreint au sous-espace  $\{x \mid x \in D\}$ . Nous supposons que

$S$  est non vide,

tout point admissible peut être approché arbitrairement par un point intérieur, c'est-à-dire, pour tout  $x \in F$ , et pour tout  $\varepsilon > 0$ , il existe  $\tilde{x} \in S$  tel que

$$\|\tilde{x} - x\| \leq \varepsilon \quad (1.3.4)$$

Si  $D$  est un ensemble convexe, et  $g$  une fonction convexe, cette hypothèse est toujours vérifiée

**Lemme 1.3.1 ([3])** soit  $D \subset \mathbb{R}^n$  un ensemble convexe fermé. soit  $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$  une fonction convexe. Soit  $F$  défini par (1.3.2) et  $S$  défini par (1.3.3). Pour tout  $x \in F$ , et pour tout  $\varepsilon > 0$ , il existe  $\tilde{x} \in S$  tel que

$$\|\tilde{x} - x\| \leq \varepsilon \quad (1.3.4)$$

**Remarque 1.3.1** Les méthodes de points intérieurs utilisent des fonctions dites fonctions barrières afin d'obliger les algorithmes à rester dans  $S$ .

**Definition 4 (fonction barrière)** Soit  $D \subset \mathbb{R}^n$  un ensemble fermé. Soit  $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$  une fonction convexe. Soit  $S$  défini par (1.3.4). Une fonction  $\Phi : S \rightarrow \mathbb{R}$  est une fonction barrière si elle est continue et si :

$$\lim_{x \in S, g(x) \rightarrow 0} \Phi(x) = +\infty \quad (1.3.5)$$

Le théorème suivant assure la convergence de la méthode barrière

**Théorème 1.3.1 ([3])** soient  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , continûment différentiable,  $D \subset \mathbb{R}^n$  fermé et  $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$  continûment différentiable. Soient les ensembles  $F$  et  $S$  définies par (1.3.2) et (1.3.3) respectivement. Supposons que  $S \neq \emptyset$ , et que tout point admissible peut être approché arbitrairement par un point intérieur, c'est à dire, pour tout  $x \in F$ , et pour tout  $\varepsilon > 0$ , il existe  $\tilde{x} \in S$  tel que

$$\|\tilde{x} - x\| \leq \varepsilon \quad (1.3.6)$$

Considérons la suite  $(x_k)_k$  où

$$x_k = \arg \min_{x \in S} f(x) + \varepsilon_k \Phi(x) \quad (1.3.7)$$

avec  $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}$  est une fonction barrière, et  $(\varepsilon_k)_k$  est telle que  $\varepsilon_k > \varepsilon_{k+1}, \forall k$ , et

$\lim_{k \rightarrow \infty} \varepsilon_k = 0$ . Alors, tout point limite de la suite  $(x_k)_k$  est un minimum global du problème d'optimisation

$$\min f(x) \text{ s.c } x \in F \quad (1.3.8)$$

## 1.4 Algorithme d'optimisation

Un algorithme d'optimisation est une procédure mathématique qui permet d'obtenir les minimums (ou maximums) d'une fonction réelle  $f$  (que l'on appelle fonction objective)  $\min_{x \in \mathbb{R}^n} f(x)$ .

En général la solution est un sous-espace  $A \in \mathbb{R}^n$  qui est soumis à un ensemble de contraintes, qui sont exprimées comme un système d'équations et inéquations. Les éléments de  $A$  sont appelés solutions admissibles et souvent ont des bornes supérieures et inférieures.

Les algorithmes d'optimisation sont des processus itératifs que génèrent une séquence de valeurs  $x_{n+1}$  à partir d'un point de départ  $x_0$ . Un algorithme est convergent quand pour n'importe quel point de départ, la séquence arrive à la solution.

Les algorithmes d'optimisation ont besoin en général des dérivées de premier et deuxième degré de la fonction pour le calcul du gradient d'une fonction, par exemple la méthode de descente de gradient a besoin juste des premières dérivées, la méthode de Newton nécessite les deuxièmes dérivées de la fonction objective, et sans dérivée on peut trouver les méthodes d'algorithme du simplexe..etc.

Dans cette section nous allons présenter un algorithme permettant de converger vers une solution optimale du problème ( $P$ ). Et nous donnerons quelques définitions.

### 1.4.1 Description

Un algorithme est défini par une application  $A$  de  $C$  dans  $C$ , où  $C$  est l'ensemble des solutions réalisables, permettant la génération d'une suite d'éléments de  $C$  par la formule :

$$\begin{cases} x_0 \in C \text{ donné, } k = 0 & \text{Etape d'initialisation} \\ x_{k+1} = A(x_k), k = k + 1 & \text{Itération} \end{cases} \quad (1.4.1)$$

Si on remplace  $C$  par son intérieur, en supposant que  $\text{int}(C) \neq \emptyset$ , 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  $C$  et réaliser une étude pour montrer sa convergence.

## 1.4.2 Notations asymptotiques

Nous donnons les définitions des notations asymptotiques suivantes :

**Definition 5** (*Notation  $\mathbf{O}$* ) Soient deux fonctions  $f, g : \mathbb{N} \rightarrow \mathbb{R}^+$ . On note  $f(n) = \mathbf{O}(g(n))$  lorsqu'il existe des entiers  $c$  et  $n_0$  tels que pour tout  $n \leq n_0$ ,

$$f(n) \leq cg(n) \tag{1.4.2}$$

*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 :*

(*Notations  $\mathbf{o}, \Omega, \theta$* ) Soient deux fonctions  $f, g : \mathbb{N} \rightarrow \mathbb{R}^+$ . On note  $f(n) = \mathbf{o}(g(n))$  lorsque pour tout réel  $c$ , il existe un entier  $n_0$  tel que pour tout  $n \geq n_0$ ,

$$f(n) \leq cg(n) \tag{1.4.3}$$

On note  $f(n) = \Omega(g(n))$  lorsqu'il existe des entiers  $s$  et  $n_0$  tels que pour tout  $n \geq n_0$ ,

$$f(n) \geq cg(n) \tag{1.4.4}$$

On note  $f(n) = \theta(g(n))$  lorsque  $f(n) = \mathbf{o}(g(n))$  et  $f(n) = \Omega(g(n))$  .

Soit  $(x_k)_{k \in \mathbb{N}}$  une suite donnée par l'algorithme  $A$  et convergente vers  $x^*$ .

# Chapitre 2

## Fonction noyau et sa qualification

### 2.1 Fonction noyau

**Definition 6** Soit  $f : \mathbb{R}_+^* \rightarrow \mathbb{R}_+$  une fonction deux fois continument différentiable. Alors  $f$  est dite fonction noyau strictement convexe, si elle vérifie les conditions suivantes :

$$\begin{aligned} f(1) = f'(1) &= 0 \\ f''(t) &> 0, \forall t > 0 \\ \lim_{t \rightarrow 0^+} f(t) &= \lim_{t \rightarrow \infty} f(t) = +\infty \end{aligned} \tag{2.1.1}$$

Les deux premiers conditions montrent que  $f$  est strictement convexe et minimale en 1 avec  $f(1) = 0$ , donc  $f$  s'écrit comme suit :

$$f(t) = \int_1^t \int_1^\xi f''(\tau) d\tau d\xi \tag{2.1.2}$$

**Lemme 2.1.1** Soit  $f$  une fonction noyau alors :

$$1 - t f'(t) \geq f(t), \forall t \geq 1,$$

2- Si  $f(t_1) = f(t_2)$ , tel que  $t_1 < 1 < t_2$  alors :

$$a- f'(t_1) < 0, f'(t_2) > 0$$

$$b- f(\alpha t_1) < f(\alpha t_2), \forall \alpha > 1$$

**Démonstration :** 1- On définit la fonction  $g$  par :  $g(t) = tf'(t) - f(t), \forall t \geq 1$  On

a :

$$g(1) = 0, \text{ car } f(1) = f'(1) = 0.$$

$$g'(t) = f'(t) + tf''(t) - f'(t)$$

$$= tf''(t) \geq 0, \text{ car } f'' > 0, \forall t \geq 1 \text{ (d'après la définition)}$$

Comme  $g'(t) \geq 0, \forall t \geq 1$  alors  $g(t)$  est croissante, en plus  $g(t) \geq g(1) = 0, \forall t \geq 1$ .

$$\text{lorsque } g(t) \geq 0 \Leftrightarrow tf'(t) - f(t) \geq 0$$

$$\Leftrightarrow tf'(t) \geq f(t)$$

2-a- Supposons que  $f(t_1) = f(t_2)$ , avec  $t_1 < 1 < t_2$  alors on a :

$$f''(t) \geq 0 \Rightarrow f''(t_1) \geq 0, \text{ alors } f' \text{ est croissante}$$

$$\text{donc } f'(t_1) < f'(1) = 0, \text{ avec } t_1 < 1$$

$$\text{et } f'(t_2) > f'(1) = 0, \text{ avec } t_2 > 1$$

donc

$$f'(t_1) < 0$$

$$f'(t_2) > 0$$

2-b-

$$t_2 f'(t_2) > f(t_2), \text{ (d'après 1)}$$

$$\lim_{t \rightarrow 1} f(t) = f(t_1) + f'(t_1)(t - t_1) + \frac{f''(\xi)}{2!}(t - t_1)^2, \text{ (formule de Taylor au voisinage de } t_1)$$

$$\lim_{t \rightarrow 1} f(t) > f(t_1) + f'(t_1)(t - t_1)$$

$$f(1) > f(t_1) + f'(t_1)(1 - t_1)$$

$$0 > f(t_1) + f'(t_1)(1 - t_1)$$

$$f(t_1) < f'(t_1)(t_1 - 1)$$

D'autre part, on définit  $u(\alpha) = f(\alpha t_1) - f(\alpha t_2)$ , pour  $\alpha \geq 1$  On a :

$$u(1) = 0$$

$$u'(\alpha) = t_1 f'(\alpha t_1) - t_2 f'(\alpha t_2)$$

comme  $f'' \geq 0$  ( $f'$  est croissante), et  $\alpha t_1 < \alpha t_2$ , pour  $\alpha > 1$ , alors

$$f'(\alpha t_1) < f'(\alpha t_2), \text{ pour } \alpha > 1$$

D'autre part,  $f'(\alpha t_2) > 0, \forall t_2 > 1, \forall \alpha > 1$ , donc

$$u'(\alpha) < t_1 f'(\alpha t_2) - t_2 f'(\alpha t_2) < 0$$

$$u'(\alpha) < (t_1 - t_2) f'(\alpha t_2) < 0, \text{ pour } \alpha > 1$$

alors

$$u(\alpha) = f(\alpha t_1) - f(\alpha t_2) < 0$$

donc

$$f(\alpha t_1) < f(\alpha t_2)$$

Ce qui termine la preuve ■

## 2.2 Qualification d'une fonction noyau

**Definition 7** Une fonction noyau  $f$  est dite être qualifiée si elle satisfait les propriétés suivantes :

$$tf''(t) + f'(t) > 0, \forall t < 1 \quad (2.2.1)$$

$$f''(t) - f'(t) > 0, \forall t > 1 \quad (2.2.2)$$

$$f'''(t) < 0, \forall t > 0 \quad (2.2.3)$$

$$2[f''(t)]^2 - f'(t)f'''(t) > 0, \forall t < 1 \quad (2.2.4)$$

$$f''(t)f'(\alpha t) - \alpha f'(t)f''(\alpha t) > 0, \forall t > 1, \forall \alpha > 1 \quad (2.2.5)$$

## 2.3 Propriétés et relation entre les conditions de qualification

Les trois prochains lemmes sont vérifiés si les trois conditions (2.2.1), (2.2.2) et (2.2.3) sont satisfaites.

**Lemme 2.3.1** ([15]) Soit  $f$  une fonction deux fois différentiable, alors les propriétés suivantes sont équivalentes :

(i)  $f(t)$  est dite exponentiellement convexe car :

$$f(\sqrt{t_1 t_2}) \leq \frac{f(t_1) + f(t_2)}{2}, \text{ pour } t_1, t_2 > 0 \quad (2.3.1)$$

(ii) la fonction  $\psi$  définie par  $\psi(\xi) = f(e^\xi)$  est convexe.

(iii)  $tf''(t) + f'(t) \geq 0, t > 0$ .

**Démonstration :** Supposons que la fonction  $g$  est continue donc pour qu'elle soit convexe, il suffit qu'elle soit mid-convexe.

Montrons que (i)  $\iff$  (ii)

On a :

$$f(\sqrt{t_1 t_2}) \leq \frac{f(t_1) + f(t_2)}{2}, \forall t_1, t_2 > 0$$

On suppose que :

$$\begin{aligned} g\left(\frac{\zeta_1 + \zeta_2}{2}\right) &= f\left(e^{\frac{\zeta_1 + \zeta_2}{2}}\right) = f(\sqrt{e^{\zeta_1} e^{\zeta_2}}) \\ &\leq \frac{f(e^{\zeta_1}) + f(e^{\zeta_2})}{2} = \frac{g(\zeta_1) + g(\zeta_2)}{2}, \forall \zeta_1, \zeta_2 \in \mathbb{R} \text{ tel que } t_1 = e^{\zeta_1}, t_2 = e^{\zeta_2} \\ &\iff g \text{ est mid-convexe et continue donc convexe} \end{aligned}$$

Concernant l'équivalence (ii)  $\iff$  (iii)

$$\begin{aligned} g \text{ convexe} &\iff g''(\xi) = [f(e^\xi)]'' = [f'(e^\xi) + e^\xi f''(e^\xi)]e^\xi \geq 0, \forall \xi \in \mathbb{R} \\ &\iff [t f''(t) + f'(t)]t \geq 0, \forall t > 0 \text{ où } t = e^\xi \\ &\iff t f''(t) + f'(t) \geq 0, \forall t > 0. \end{aligned}$$

Ce qui termine la preuve ■

**Lemme 2.3.2 ([7])** Soit  $f$  une fonction deux fois différentiable, alors les propriétés suivantes sont équivalentes :

(i)  $f\left(\sqrt{\frac{\zeta_1^2 + \zeta_2^2}{2}}\right) \leq \frac{f(\zeta_1) + f(\zeta_2)}{2}, \zeta_1, \zeta_2 > 0.$

(ii) la fonction  $\phi$  définie par  $\phi(t) = f(\sqrt{t})$  est convexe,  $t > 0.$

(iii)  $t f''(t) - f'(t) \geq 0, t > 0.$

**Démonstration :** Signalons que la fonction  $\phi$  est continue donc pour qu'elle soit convexe, il suffit qu'elle soit mid-convexe. Montrons que (i)  $\iff$  (ii)

$$f\left(\sqrt{\frac{\zeta_1^2 + \zeta_2^2}{2}}\right) \leq \frac{1}{2}(f(\zeta_1) + f(\zeta_2)), \forall \zeta_1, \zeta_2 \in \mathbb{R}_+^*$$

on suppose que  $\zeta_1 = \sqrt{t_1}, \zeta_2 = \sqrt{t_2} \in \mathbb{R}_+^*$  alors

$$f\left(\sqrt{\frac{t_1 + t_2}{2}}\right) \leq \frac{1}{2}(f(\sqrt{t_1}) + f(\sqrt{t_2}))$$

$$\phi\left(\frac{t_1+t_2}{2}\right) = f\left(\sqrt{\frac{t_1+t_2}{2}}\right) \leq \frac{f(\sqrt{t_1}) + f(\sqrt{t_2})}{2} = \frac{\phi(t_1) + \phi(t_2)}{2}, \forall t_1, t_2 \in \mathbb{R}_+^* \text{ tel que } t_1 = \zeta_1^2, t_2 = \zeta_2^2$$

$\iff \phi$  est mid-convexe et continue donc elle est convexe

Concernant l'équivalence (ii)  $\iff$  (iii)

$$\phi \text{ convexe} \iff \phi''(\xi) = [f(\sqrt{t})]'' = \frac{1}{4t^{\frac{3}{2}}}(tf''(t) - f'(t)) \geq 0, \text{ pour } t > 0$$

$$\iff tf''(t) - f'(t) \geq 0, \forall t > 0$$

Ce qui termine la preuve. ■

**Lemme 2.3.3** ([7]) *Si  $f$  vérifie (2.2.2) et (2.2.3) de la définition 5, alors  $f$  vérifie (2.2.5) :*

$$f''(t)f'(\alpha t) - \alpha f'(t)f''(\alpha t) > 0, \forall \alpha > 1 \quad (2.3.2)$$

**Démonstration :** Supposons que  $f$  vérifie (2.2) et (2.3) de la définition 5. Soit  $t > 1$ ,

on considère  $u(\alpha) = f''(t)f'(\alpha t) - \alpha f'(t)f''(\alpha t), \forall \alpha > 1$ , alors :

$$u'(\alpha) = tf''(\alpha t)f''(t) - f'(t)f''(\alpha t) - \alpha tf'(t)f'''(\alpha t), \forall \alpha > 1$$

$$u'(\alpha) = f''(\alpha t)[tf''(t) - f'(t)] - \alpha tf'(t)f'''(\alpha t), \forall \alpha > 1$$

Mais on a :

$$f''(\alpha t) > 0 \text{ d'après la convexité de } f$$

$$tf''(t) - f'(t) > 0 \text{ d'après (2.2)}$$

$$f'(t) > 0, \forall t > 1$$

$$f'''(\alpha t) < 0, \text{ d'après (2.3)}$$

Donc  $u'(\alpha) > 0$  c'est à dire la fonction  $u$  est strictement croissante et  $u(1) = 0$ , alors  $u(\alpha) > 0, \forall \alpha > 1$ .

Donc

$$f''(t)f'(\alpha t) - \alpha f'(t)f''(\alpha t) > 0, \forall \alpha > 1$$

Ce qui termine la preuve . ■

**Lemme 2.3.4 ([7])** Soit  $f$  satisfait (2.2.3) alors :

1-  $f$  admet les propriétés suivantes, en distinguant deux cas différents :

Le premier cas :  $\forall t < 1$ , on a :

$$\begin{aligned} f(t) &< \frac{f'(t)}{2}(t-1) \\ f'(t) &< f''(t)(t-1) \\ \frac{1}{2}(t-1)^2 f''(1) &< f(t) < \frac{f''(t)}{2}(t-1)^2 \end{aligned}$$

Le deuxième cas :  $\forall t > 1$ , on a :

$$\begin{aligned} \frac{f'(t)}{2}(t-1) &< f(t) \\ f''(t)(t-1) &> f'(t) \\ \frac{1}{2}(t-1)^2 f''(t) &< f(t) < \frac{f''(1)}{2}(t-1)^2 \end{aligned}$$

2- Si  $f(t_1) = f(t_2)$ , telle que  $t_1 < 1 < t_2$  alors :

$$f'(t_2) < -f'(t_1)$$

**Démonstration :** 1- On définit  $u(t) = 2f(t) - (t-1)f'(t)$ , pour  $t > 0$ . On a :

$$\begin{aligned} u(1) &= 0 \\ u'(t) &= f'(t) - (t-1)f''(t) \\ u'(1) &= 0 \\ u''(t) &= -(t-1)f'''(t) \end{aligned}$$

Donc nous distinguons les deux cas suivante

**cas 1 :** si  $0 < t < 1$ , alors

$$u''(t) = -(t-1)f'''(t) < 0$$

car

$$(t - 1) < 0 \text{ pour } 0 < t < 1$$

$$f'''(t) < 0$$

donc  $u'$  est décroissante et  $u'(1) = 0$ , alors

$$u'(t) = f'(t) - (t - 1)f''(t) > 0$$

donc  $u$  est croissante et  $u(1) = 0$ , alors

$$u(t) = 2f(t) - (t - 1)f'(t) < 0$$

donc

$$f'(t) < (t - 1)f''(t)$$

$$f(t) < \frac{1}{2}(t - 1)f'(t)$$

En utilisant le développement de Taylor au voisinage de  $t_0 = 1$ , on a :

$$f(t) = f(1) + f'(1)(t - 1) + \frac{f''(1)}{2!}(t - 1)^2 + \frac{f'''(\xi)}{3!}(t - 1)^3, t < \xi < 1$$

$$f(t) > \frac{f''(1)}{2}(t - 1)^2 \text{ d'après (2.3) comme } f'''(t) < 0$$

On a :

$$f(t) < \frac{f'(t)}{2}(t - 1) < \frac{f''(t)(t-1)}{2}(t - 1) < \frac{f''(t)}{2}(t - 1)^2$$
$$\frac{f''(1)}{2}(t - 1)^2 < f(t) < \frac{f''(t)}{2}(t - 1)^2$$

alors

$$f(t) < \frac{f'(t)}{2}(t - 1)$$

$$f'(t) > f''(t)(t - 1)$$

$$\frac{1}{2}(t - 1)^2 f''(1) < f(t) < \frac{f''(t)}{2}(t - 1)^2$$

**cas 2 :** si  $t > 1$ , alors

$$u''(t) = -(t - 1)f'''(t) > 0$$

ceci implique que  $u'$  est croissante avec  $u'(1) = 0$ , donc

$$u'(t) = f'(t) - (t-1)f''(t) > 0$$

comme  $u'(t) > 0$ , alors  $u$  est croissante avec  $u(1) = 0$ , alors

$$u(t) = 2f(t) - (t-1)f'(t) > 0$$

donc

$$f'(t) > (t-1)f''(t)$$

$$f(t) > \frac{1}{2}(t-1)f'(t)$$

alors  $\frac{1}{2}(t-1)^2 f''(t) < f(t)$ , et en utilisant le développement de Taylor au voisinage de  $t_0 = 1$ , on a :

$$f(t) = \frac{f''(1)}{2}(t-1)^2 + \frac{f'''(\xi)}{6}(t-1)^3, \xi > 1$$

comme  $f'''(t) < 0$ , donc  $f(t) < \frac{1}{2}(t-1)^2 f''(1)$ , alors

$$\frac{f'(t)}{2}(t-1) < f(t)$$

$$f''(t)(t-1) < f'(t)$$

$$\frac{f''(t)}{2}(t-1)^2 < f(t) < \frac{f''(1)}{2}(t-1)^2$$

2- Suppose que  $f(t_1) = f(t_2)$ , telle que  $t_1 < 1 < t_2$ , et utilisant les cas 1 et cas 2 alors :

$$\frac{f''(1)}{2}(t_1-1)^2 < f(t_1) = f(t_2) < \frac{f''(1)}{2}(t_2-1)^2$$

comme  $f''(1) > 0$ , alors :  $1 - t_1 < t_2 - 1$ . D'autre part, supposons que  $-f(t_1) < f'(t_2)$ , on a :

$$\begin{aligned} f(t_2) &> \frac{f'(t_2)}{2}(t_2 - 1) \\ &> \frac{f'(t_2)}{2}(1 - t_1) \\ &> -\frac{f'(t_1)}{2}(1 - t_1) \\ &> \frac{f'(t_1)}{2}(t_1 - 1) \\ &> f(t_1) \end{aligned}$$

Contradiction avec l'hypothèse  $f(t_2) = f(t_1)$ , donc  $f'(t_2) < -f'(t_1)$ .

Ce qui termine la preuve. ■

# Chapitre 3

## Méthode de Trajectoire Centrale via une fonction noyau

### 3.1 Présentation de la méthode

Dans ce chapitre on donne un aperçu sur les méthodes primales-duales de trajectoire basée sur l'approche barrière logarithmique. Les méthodes primales-duales de trajectoire centrale, ont été introduites à la fin des années 80, comme variante de l'approche de Karmarkar, et furent pleinement développées au début des années 90.

La méthode barrières logarithmique de type primal-dual de *TC* pour *PL* pour son description, on considère une deuxième fois le programme linéaire sous la forme suivante :

$$\begin{aligned} (P) \min_x \{c^t x : Ax = b, x \geq 0\} \\ (D) \max_{(y,s)} \{b^t y : A^t y + s = c, y \in \mathbb{R}^m, s \in \mathbb{R}^n, s \geq 0\} \end{aligned} \tag{3.1.1}$$

Où  $(P)$  représente le problème primale et  $(D)$  son problème dual associé. Avec  $A \in \mathbb{R}^{m \times n}$  une matrice donnée,  $b \in \mathbb{R}^m$  et  $c, x \in \mathbb{R}^n$ .

Tout au long de ce chapitre, nous formulons les hypothèses suivantes :

**Hypothèse 1** : La matrice  $A$  est de plein rang (c'est à dire  $Rang(A) = m \leq n$ ).

**Hypothèse 2** : Condition de points intérieurs (**IPC**), c'est à dire qu'il existe  $(x^0, y^0, s^0)$  tel que :

$$Ax^0 = b, x^0 > 0, A^t y^0 + s^0 = c, s^0 > 0$$

On applique le théorème de la dualité forte aux problèmes duaux  $(P)$  et  $(D)$ , et on obtient le problème suivant :

$$\begin{cases} Ax & = b, x \geq 0 \\ A^t y + s & = c, s \geq 0 \\ xs & = 0 \end{cases} \quad (3.1.2)$$

La résolution de ce système est équivalente à la résolution des systèmes  $(P)$  et  $(D)$ .

L'idée principale d'**IPMs** est de perturber la condition de complémentarité et spécifiquement la partie droite de la troisième équation du système (3.1.2)

$$\begin{cases} Ax & = b, x \geq 0 \\ A^t y + s & = c, s \geq 0 \\ xs & = \mu e, \mu > 0 \end{cases} \quad (3.1.3)$$

D'après la dernière équation du système (3.1.3),  $x_i(\mu)s_i(\mu) = \mu, \forall i : \overline{1..n}$ , donc  $x^t s = n\mu$  où  $x^t s$  désigne le saut dualité, donc il suffit de trouver (approximativement) des points sur la trajectoire centrale quand  $\mu \rightarrow 0$ .

Nous remarquons que ce système représente une méthode barrière. Alors il admet une unique solution  $(x(\mu), y(\mu), s(\mu))$ , pour chaque  $\mu > 0$ , et nous pouvons supposer que  $x^0 = s^0 = e$  sans perte de généralité, on suppose aussi que  $(x(\mu), y(\mu), s(\mu))$  est connu pour certaines valeurs positives de  $\mu$ .

Par exemple :  $\mu = 1, x(1) = s(1) = e$ . avec  $\mu = (1 - \theta), \forall \theta \in ]0, 1[$ , et on obtient le système de Newton suivant :

$$\begin{cases} A\Delta x & = 0 \\ A^t\Delta y + \Delta s & = 0 \\ s\Delta x + x\Delta s & = \mu e - xs \end{cases} \quad (3.1.4)$$

Le système (3.1.4) admet une solution unique dite la direction de Newton classique pour l'optimisation linéaire désignée par  $(\Delta x, \Delta y, \Delta s)$ . Ensuite,  $\mu$  est réduit par le facteur  $(1 - \theta)$ , et nous appliquons de nouveau la méthode de Newton ciblant les nouveaux  $\mu$ -centres. Ce processus est répété jusqu'à ce que  $\mu$  devient assez petit (c'est à dire  $n\mu \leq \varepsilon$ ), alors dans ce cas nous obtenons une  $\varepsilon$ -solution des problèmes  $(P)$  et  $(D)$ .

Le nouvel itéré de la méthode de Newton avec un pas fixé est donné par :

$$x_+ = x + \alpha\Delta x, \quad y_+ = y + \alpha\Delta y, \quad s_+ = s + \alpha\Delta s, \quad \text{où } \alpha \in ]0, 1]. \quad (3.1.5)$$

Maintenant, nous introduisons le vecteur réduit  $v$  et les directions de Newton mise à l'échelle  $(d_x, d_s)$  comme suit :

$$v = \sqrt{\frac{xs}{\mu}}, \quad v^{-1} = \sqrt{\left(\frac{xs}{\mu}\right)^{-1}}, \quad d_x = \frac{v\Delta x}{x}, \quad d_s = \frac{v\Delta s}{s}. \quad (3.1.6)$$

On utilise (3.1.6) dans la troisième équation du système (3.1.4) on obtient :

$$\begin{cases} A\Delta x = 0 \\ A^t\Delta y + \Delta s = 0 \\ s\Delta x + x\Delta s = \mu v(v^{-1} - v) \end{cases} \quad (3.1.7)$$

Maintenant on pose  $\bar{A} = \frac{1}{\mu}AV^{-1}X$ ,  $V = \text{diag}(v)$ ,  $X = \text{diag}(x)$ . Alors (3.1.7) devient :

$$\left\{ \begin{array}{l} \bar{A}d_x = 0 \\ \bar{A}^t\Delta y + d_s = 0 \\ d_x + d_s = v^{-1} - v \end{array} \right. \quad (3.1.8)$$

On remarque que  $v^{-1} - v$  n'est autre que l'opposé du gradient d'une fonction barrière logarithmique  $\Phi$ , où  $d_x + d_s = -\nabla\Phi(v)$ .

Donc le système (3.1.8) peut être réécrit comme suit :

$$\left\{ \begin{array}{l} \bar{A}d_x = 0 \\ \bar{A}^t\Delta y + d_s = 0 \\ d_x + d_s = -\nabla\Phi(v) \end{array} \right. \quad (3.1.9)$$

Où la fonction barrière  $\Phi(v) : (\mathbb{R}_+^*)^n \longrightarrow \mathbb{R}_+$  est définie comme suit :

$$\Phi(v) = \Phi(x, s, \mu) = \sum_{i=1}^n f(v_i) \quad (3.1.10)$$

Avec

$$f(v_i) = \frac{v_i^2 - 1}{2} - \log v_i \quad (3.1.11)$$

**Remarque 3.1.1** On a d'après le système (3.1.8)

$$\bar{A}^t\Delta y + d_s = 0 \Leftrightarrow d^t x \bar{A}^t \Delta y + d^t x d_s = 0 \quad (3.1.12)$$

mais

$$d^t x \bar{A}^t = \bar{A}d_x = 0 \quad (3.1.13)$$

Alors

$$\overline{A}^t \Delta y + d_s = 0 \Leftrightarrow d^t x d_s = 0 \quad (3.1.14)$$

Nous remarquons que si  $(d_x, \Delta y, d_s)$  est une solution du système (3.1.9), alors  $d_x, d_s$  sont orthogonaux.

Nous utilisons la fonction de proximité  $\Phi_l(v)$  (la fonction barrière logarithmique) pour mesurer la distance entre l'itéré  $\mu$ -centre et la trajectoire centrale pour  $\mu > 0$  donné. Nous définissons également la mesure de proximité, basée sur la norme,  $\delta(v) : (\mathbb{R}_+^*)^n \rightarrow \mathbb{R}_+$ , comme suit :

$$\delta(v) = \frac{1}{2} \|\nabla \Phi(v)\| = \frac{1}{2} \|d_x + d_s\| \quad (3.1.15)$$

## 3.2 Générique Primal-dual IPMs pour l'optimisation linéaire

Dans cette section on remplace la fonction noyau  $f$  de la fonction barrière logarithmique  $\Phi$  par une nouvelle fonction  $u$  qui est défini dans la partie suivante et supposons que  $\tau \geq 1$ . Alors l'algorithme se déroule comme suit :

Supposons que l'on donne un point strictement réalisable  $(x, y, s)$  qui est dans un  $\tau$ -voisinage de  $\mu$ -centre. Ensuite, nous diminuons  $\mu$  par une suite géométrique de raison  $(1 - \theta)$ , pour  $0 < \theta < 1$  puis on résout le système de Newton pour obtenir la direction de descente.

La condition de positivité d'une nouvelle itération est assurée avec le bon choix de la taille du pas qui est défini dans la partie suivante. Cette procédure est répétée jusqu'à ce qu'on trouve une nouvelle itération  $(x^+, y^+, s^+)$  qui est dans un  $\tau$ -voisinage de  $\mu^+$ -centre. Ce processus est répété jusqu'à ce que est suffisamment petit, i.e.,  $n\mu \leq \varepsilon$ .

Les paramètres  $\tau, \theta$  et le pas de déplacement  $\alpha$  doivent être choisie de telle sorte que

l'algorithme est optimisée dans le sens où le nombre d'itérations requises est le plus petit possible.

Le choix de paramètre  $\theta$  joue un rôle très important dans la théorie et la pratique des **IPMs**. Si  $\theta$  est une constante indépendante de la dimension  $n$  du problème, alors nous appelons l'algorithme de grand pas. Et si  $\theta$  dépend de la dimension du problème, tel que  $\theta = \frac{1}{\sqrt{n}}$ , alors l'algorithme est appelé algorithme de petit pas.

**Algorithme 3.2.1** *L'algorithme générique Primal-dual **IPMs** pour l'optimisation linéaire donnée comme suit :*

**Initialisation :**

Un paramètre de seuil  $0 < \tau < 1$  ;

Un paramètre de précision  $\varepsilon > 0$  ;

Un paramètre barrière fixe  $\theta, 0 < \theta < 1$  ;

Une fonction de proximité  $\Phi(v)$  ;

**Début :**

Soit  $(x^0, y^0, s^0)$  vérifie la **IPC** (une solution initiale réalisable) tel que :

$$k = 0; \mu^0 = 1; v^0 = \sqrt{\frac{x^0 s^0}{\mu^0}}; \Phi(x^0, y^0, s^0) \leq \tau ;$$

**Tant que :**  $(n\mu > \varepsilon)$  **faire :**

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

**Tant que :**  $(\Phi(v) > \tau)$  **faire :**

(Itération interne)

Résoudre le système (3.9) pour déterminer  $(d_x, \Delta y, d_s)$  puis  $(\Delta x, \Delta y, \Delta s)$  ;

Déterminer le pas de déplacement réalisable  $\alpha$  ;

**Mise à jour :**

$$x := x + \alpha \Delta x$$

$$y := y + \alpha \Delta y$$

$$s := s + \alpha \Delta s$$

$$v := \sqrt{\frac{xs}{\mu}}$$

Fin Tant que (itération interne)

*Fin Tant que (itération externe)*

*Fin Algorithme.*

### 3.3 Etude de la convergence de la méthode de Trajectoire Centrale via une fonction noyau

#### 3.3.1 Borne supérieure de $\Phi(v)$ pour chaque itération externe

Au début de chaque itération externe de l'algorithme, juste avant la mise à jour du paramètre  $\mu$ , on a :  $\Phi(v) < \tau$  . Après la réduction de  $\mu$  par le facteur  $(1 - \theta)$  il devient  $\mu^+$  tel que :

$$\mu^+ = (1 - \theta)\mu, \quad \text{avec } 0 < \theta < 1 \quad (3.3.1)$$

Le vecteur  $v$  est mise à jour par la relation suivante :

$$v^+ = \frac{v}{\sqrt{1 - \theta}} \quad (3.3.2)$$

Ceci conduit en général à une augmentation de la valeur de  $\Phi(v)$  dans les itérations externes, puis, au cours des itérations internes  $\Phi(v)$  diminue jusqu'à ce qu'elle atteigne la première valeur inférieure ou égale à  $\tau$  . Au cours de l'algorithme, les plus grandes valeurs de  $\Phi(v)$  se produisent juste après les mises à jour de  $\mu$ . Pour cela, nous avons besoin d'une bonne estimation de la borne supérieure de  $\Phi(v)$ .

Il deviendra clair que dans l'analyse de l'algorithme certaines fonctions inverse. Rapport avec les fonctions noyau et leurs dérivées premières jouent un rôle crucial. Nous introduisons ces fonctions inverses ici.

$\varrho : [0, \infty[ \rightarrow [1, +\infty[$  la fonction inverse de  $f$ .

$\rho : [0, \infty[ \rightarrow ]0, 1[$  la fonction inverse de  $-\frac{1}{2}f'$ .

Supposons dans toute la suite que  $f$  est une fonction noyau qualifiée. Nous avons le résultat suivant.

**Théorème 3.3.1 ([2])** *Pour tout vecteur positif  $v$  et tout  $\alpha > 1$ , on a :*

$$\Phi(\alpha v) \leq n f \left( \alpha \varrho \left( \frac{\Phi(v)}{n} \right) \right) \quad (3.3.3)$$

**Corollaire 3.3.1 ([2])** *Pour tout vecteur positif  $v$ , si  $\Phi(v) \leq \tau$  et  $\frac{1}{\sqrt{1-\theta}} > 1$ , on a :*

$$L_f(n, \theta, \tau) \leq (\Phi)_0 \quad (3.3.4)$$

$$(\Phi)_0 = \frac{n}{2} f''(1) \left( \frac{\varrho \left( \frac{\tau}{n} \right)}{\sqrt{1-\theta}} - 1 \right)^2 \quad (3.3.5)$$

### 3.4 Analyse de la décroissance de la fonction barrière de proximité $\Phi$

Dans cette partie, nous analysons la complexité de l'algorithme, nous commençons en premier à donner des conditions qui assurent la stricte faisabilité d'une itération de Newton d'où on fixe  $\mu$  et on calcule le pas de déplacement  $\alpha$  ensuite on prouve la décroissance de la fonction barrière de proximité  $\Phi$

$$x^+ = x + \alpha \Delta x, \quad s^+ = s + \alpha \Delta s \text{ et } y^+ = y + \alpha \Delta y \quad (3.4.1)$$

On utilise (3.1.6) dans ce système et on obtient

$$\begin{aligned} x^+ &= x + \left( e + \alpha \frac{\Delta x}{x} \right) = x \left( e + \alpha \frac{d_x}{v} \right) = \frac{x}{v} (v + \alpha d_x) \\ s^+ &= s + \left( e + \alpha \frac{\Delta s}{s} \right) = s \left( e + \alpha \frac{d_s}{v} \right) = \frac{s}{v} (v + \alpha d_s) \end{aligned} \quad (3.4.2)$$

Donc pour  $\mu$  fixé, on a

$$\begin{aligned} v_+ &= \sqrt{\frac{x^+ s^+}{\mu}} \\ &= \sqrt{(v + \alpha d_x)(v + \alpha d_s)} \end{aligned} \quad (3.4.3)$$

Pour tout  $\alpha > 0$  on pose :

$$u(\alpha) = \Phi(v_+) - \Phi(v) \quad (3.4.4)$$

Donc  $u(\alpha)$  est la différence de la proximité entre le nouveaux et ancien itérés, et on a :

$$\begin{aligned} \Phi(v_+) &= \Phi\left(\sqrt{(v + \alpha d_x)(v + \alpha d_s)}\right) \\ &\leq \frac{1}{2}(\Phi(v + \alpha d_x) + \Phi(v + \alpha d_s)) \end{aligned} \quad (3.4.5)$$

D'autre part

$$u(\alpha) \leq u_1(\alpha) \quad (3.4.6)$$

Où

$$u_1(\alpha) = \frac{1}{2}(\Phi(v + \alpha d_x) + \Phi(v + \alpha d_s)) - \Phi(v) \quad (3.4.7)$$

Avec

$$u(0) = u_1(0) = 0 \quad (3.4.8)$$

La première dérivée de  $u_1$  en point  $\alpha$  est :

$$u_1'(\alpha) = \frac{1}{2} \sum_{i=1}^n (f'(v_i + \alpha[d_x]_i)[d_x]_i + f'(v_i + \alpha[d_s]_i)[d_s]_i) \quad (3.4.9)$$

Où  $[d_x]_i$  et  $[d_s]_i$  désigne respectivement la  $i^{\text{ème}}$  composante des vecteurs  $d_x$  et  $d_s$ . On utilise (3.1.15), on obtient :

$$\begin{aligned} u_1'(0) &= \frac{1}{2} \nabla \Phi(v)^t (d_x + d_s) \\ &= -\frac{1}{2} \nabla \Phi(v)^t \nabla \Phi(v) \\ &= -2(\delta(v))^2. \end{aligned}$$

La deuxième dérivée de  $u_1$  en point  $\alpha$  est :

$$u_1''(\alpha) = \frac{1}{2} \sum_{i=1}^n (f''(v_i + \alpha[d_x]_i)[d_x]_i^2 + f''(v_i + \alpha[d_s]_i)[d_s]_i^2) \quad (3.4.10)$$

D'où  $u_1''(\alpha) > 0$ , si  $d_x \neq 0$ , ou  $d_s \neq 0$ , alors dans ce cas  $u_1$  est strictement convexe.

On note par  $\delta = \delta(v)$ ,  $\Phi = \Phi(v)$  et  $v_{\min} = \min_i(v_i)$ .

**Lemme 3.4.1 ([2])** *Soit  $u_1(\alpha)$  défini dans (3.4.7) et  $\delta$  défini dans (3.1.15), alors on a :*

$$u_1''(\alpha) = 2\delta^2 f''(v_{\min} - 2\alpha\delta) \quad (3.4.11)$$

**Lemme 3.4.2 ([2])** *Si le pas de déplacement  $\alpha$  vérifie l'inégalité suivante*

$$-f'(v_{\min} + 2\alpha\delta) + f'(v_{\min}) \leq 2\delta \quad (3.4.12)$$

*Alors*

$$u_1'(\alpha) \leq 0 \quad (3.4.13)$$

**Lemme 3.4.3 ([2])** *L'estimation de la plus grande valeur de  $\alpha$  vérifiant (3.4.12) est donnée par*

$$\bar{\alpha} = \frac{1}{2\delta} (\rho(\delta) - \rho(2\delta)) \quad (3.4.14)$$

**Lemme 3.4.4 ([2])** *Soient  $\rho$  et  $\alpha$  définis dans le lemme (2.3.1) alors on a :*

$$\bar{\alpha} \geq \frac{1}{f''(\rho(2\delta))} = \tilde{\alpha} \quad (3.4.15)$$

**Lemme 3.4.5 ([2])** *On suppose que  $h$  est une fonction convexe et deux fois différentiable avec*

$$h(0) = 0, h'(0) < 0 \quad (3.4.16)$$

*et  $h$  atteint son minimum global à  $t^* > 0$ , et  $h''$  est croissante pour tout  $t \in [0, t^*]$ , alors pour tout  $t \in [0, t^*]$ , on a*

$$h(t) \leq \frac{th'(0)}{2} \quad (3.4.17)$$

**Lemme 3.4.6 ([2])** *Si le pas de déplacement  $\alpha$  vérifie  $\alpha \leq \bar{\alpha}$  alors*

$$u(\alpha) \leq -\alpha\delta^2 \quad (3.4.18)$$

### 3.4.1 Analyse de la complexité algorithmique de l'algorithme

**Lemme 3.4.7** ([15]) *Si  $\alpha \in [0, 1]$  alors*

$$(1 + t)^\alpha \leq 1 + \alpha t, \forall t \geq -1 \quad (3.4.19)$$

**Démonstration :** On définit la fonction  $u$  par :

$$u(t) = (1 + t)^\alpha - 1 - \alpha t, \forall t \geq -1$$

Alors

$$\begin{aligned} u'(t) &= \alpha(1 + t)^{\alpha-1} - \alpha \\ u''(t) &= \alpha(\alpha - 1)(1 + t)^{\alpha-2} \\ u''(t) &\leq 0 \end{aligned}$$

Donc  $u$  est concave et  $u'(0) = 0$ , alors

$$u(t) \leq u(0) = 0$$

Ce qui termine la preuve. ■

**Lemme 3.4.8** ([15]) *Soit  $t_0, t_1, \dots, t_K$  une suite des nombres positifs qui vérifie :*

$$t_{k+1} \leq t_k - \beta t_k^{1-\gamma}, k = 0, 1, \dots, K - 1$$

*tels que  $\beta > 0$  et  $0 < \gamma \leq 1$ , alors :*

$$K \leq \left\lceil \frac{t_0^\gamma}{\beta_\gamma} \right\rceil \quad (3.4.20)$$

**Démonstration :** En utilisant de lemme (3.4.2), nous pouvons écrire

$$\begin{aligned}
t_{k+1}^\gamma &\leq (t_k - \beta t_k^{1-\gamma})^\gamma \\
&= t_k^\gamma (1 - \beta t_k^{-\gamma})^\gamma \\
&\leq t_k^\gamma (1 - \gamma \beta t_k^{-\gamma}) \\
&= t_k^\gamma - \gamma \beta
\end{aligned}$$

Donc  $t_k^\gamma \leq t_0^\gamma - \gamma \beta k$ , en remplaçant  $k$  par  $K$ , on obtient  $t_0^\gamma - \gamma \beta K > 0$ , alors

$$K \leq \left\lfloor \frac{t_0^\gamma}{\beta \gamma} \right\rfloor$$

Ce qui termine la preuve. ■

On note par  $(\Phi)_0$  la première  $\mu$ -mise à jour de  $\Phi(v)$  et  $\Phi_k$ ,  $k = 1, 2, \dots, K$  suite des valeurs de  $\Phi(v)$  dans les itérations internes. Alors d'après le corollaire (3.3.1) on a :

$$(\Phi)_0 = \frac{n}{2} f''(1) \left( \frac{\varrho\left(\frac{\tau}{n}\right)}{\sqrt{1-\theta}} - 1 \right)^2 \quad (3.4.21)$$

On pose qu'ils  $\exists \beta > 0$  et  $0 < \gamma \leq 1$ , tels que :

$$-\beta \Phi_k^{1-\gamma} \leq -\beta (\Phi_k - \tau)^{1-\gamma} \quad (3.4.22)$$

Alors

$$(\Phi_{k+1} - \tau) - (\Phi_k - \tau) \leq -\beta (\Phi_k - \tau)^{1-\gamma} \quad (3.4.23)$$

donc

$$(\Phi_{k+1} - \tau) \leq (\Phi_k - \tau) - \beta (\Phi_k - \tau)^{1-\gamma} \quad (3.4.24)$$

On utilise le lemme (3.4.3) pour  $t_k = \Phi_k - \tau > 0$ , on a  $K$  le nombre des itérations internes de chaque itération externe, écrit comme suit :

$$K \leq \frac{[(\Phi)_0 - \tau]^\gamma}{\beta\gamma} \leq \frac{[(\Phi)_0]^\gamma}{\beta\gamma} = \frac{\left(\frac{n}{2} f''(1) \left(\frac{\varrho(\frac{\tau}{n})}{\sqrt{1-\theta}} - 1\right)^2\right)^\gamma}{\beta\gamma} \quad (3.4.25)$$

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

**Théorème 3.4.1 ([7])** *Soit  $h$  le nombre d'itérations internes nécessaires pour trouver une solution optimale primale-duale approchée à une précision  $\epsilon > 0$ , alors on a :*

$$k \leq \frac{1}{\theta} \log \frac{n}{\epsilon} \quad (3.4.26)$$

**Démonstration :** Si le paramètre de la trajectoire centrale avec  $\mu^0 = 1$ , et  $\mu_k = (1 - \theta)^k \mu^0$ ,  $n\mu \leq \epsilon$ , on a :

$$\begin{aligned} (1 - \theta)^k \mu^0 n \leq \epsilon &\implies (1 - \theta)^k \leq \frac{\epsilon}{n} \\ &\implies k \log(1 - \theta) \leq \log \frac{\epsilon}{n} \end{aligned}$$

Puisque  $\log(1 - \theta) \geq -\theta$  on obtient

$$\begin{aligned} k\theta &\leq \log \frac{n}{\epsilon} \\ k &\leq \frac{1}{\theta} \log \frac{n}{\epsilon} \end{aligned}$$

Ce qui termine la preuve. ■

L'équation (3.4.27) représente le nombre total d'itérations nécessaires pour trouver une solution optimale primale-duale approchée avec des précisions  $\epsilon > 0$  et  $t > 0$  qui est

égale à le nombre d'itérations internes par le nombre d'itérations externes :

$$\frac{[(\Phi)_0]^\gamma}{\beta\gamma\theta} \log \frac{n}{\epsilon} = \frac{\left( \frac{n}{2} f''(1) \left( \frac{e(\frac{\tau}{n})}{\sqrt{1-\theta}} - 1 \right)^2 \right)^\gamma}{\beta\gamma\theta} \quad (3.4.27)$$

Dans (3.4.27), si on prend  $\tau = O(n)$  et  $\theta = \boldsymbol{\theta}(1)$ , le nombre d'itérations pour **IPMs** est trouvé à grande-pas. Par contre et si prond  $\tau = O(1)$  et  $\theta = \boldsymbol{\theta}(\frac{1}{\sqrt{n}})$ , le nombre d'itérations pour **IPMs** est trouvé à petite-pas.

## Conclusion

Les méthodes de la trajectoire centrale primal-dual sont les méthodes les plus attractives parmi les méthodes de points intérieurs pour résoudre un large éventail de problèmes d'optimisation en raison de leur complexité polynomiale et leur efficacité numérique. Elles sont connues par leur efficacité, rapidité de convergence, simplicité algorithmique et capacité de résoudre des problèmes de grandes tailles.

Dans ce mémoire, nous avons apporté des aménagements d'ordre théorique et algorithmique sur la méthode de trajectoire centrale via une fonction noyau. Cette approche a l'avantage de démarrer avec n'importe quel point  $(x^0, y^0, s^0)$  satisfaisant la condition **IPC**.

L'inconvénient majeur de ce type de méthodes réside dans sa complexité algorithmique. Cette dernière est égale à celle de la méthode de trajectoire centrale classique (c'est-à-dire le nombre d'itérations externes multiplié par le nombre d'itérations internes).

La meilleure complexité existante jusqu'à présent est  $\mathbf{O}(\sqrt{n} \log n \log \frac{n}{\epsilon})$ , pour la méthode à grand pas, où  $n$  désigne la taille du problème ( $P$ ).

En fin nous allons donner un tableau qui contient quelques fonctions noyau les plus connues sur la programmation linéaire et leurs grands et petits pas correspondant à chaque fonction.

$i$	Fonction noyau $f_i$	Grand-pas	Petite-pas	Ref
1	$\frac{t^2-1}{2} - \log t.$	$\mathbf{O}\left(n \log \frac{n}{\epsilon}\right)$	$\mathbf{O}\left(\sqrt{n} \log \frac{n}{\epsilon}\right)$	[18]
2	$\frac{t^{1+p}-1}{1+p} - \log t, \quad p \in [0, 1].$	$\mathbf{O}\left(n \log \frac{n}{\epsilon}\right)$	$\mathbf{O}\left(\sqrt{n} \log \frac{n}{\epsilon}\right)$	[11]
3	$\frac{t^2-1}{2} + \frac{6}{\pi} \tan\left(\pi \frac{1-t}{4t+2}\right).$	?	?	[2]
4	$\frac{t^2-1}{2} + \frac{6}{\pi} \tan\left(\pi \frac{1-t}{4t+2}\right).$	$\mathbf{O}\left(n^{\frac{3}{4}} \log \frac{n}{\epsilon}\right)$	$\mathbf{O}\left(\sqrt{n} \log \frac{n}{\epsilon}\right)$	[12]
5	$\frac{t^2-1}{2} - \log t + \frac{1}{8} \tan^2\left(\pi \frac{1-t}{4t+2}\right).$	$\mathbf{O}\left(n^{\frac{2}{3}} \log \frac{n}{\epsilon}\right)$	---	[16]
6	$\frac{t^2-1}{2} - \int_1^t e^{3\left(\left(\tan \frac{\pi}{2x+2}\right)^{-1}\right)} dx.$	$\mathbf{O}\left(\sqrt{n} (\log n)^2 \log \frac{n}{\epsilon}\right)$	---	[17]
7	$\frac{t^2-1}{2} - \log t + \lambda \tan^2 h(t),$ $h(t) = \frac{\pi(1-t)}{(3t+2)}, 0 < \lambda \leq \frac{8}{25}.$	$\mathbf{O}\left(n^{\frac{2}{3}} \log \frac{n}{\epsilon}\right)$	---	[9]
8	$t^2 - 2t + \frac{1}{\sin h(t)},$ $h(t) = \frac{\pi t}{1+t}.$	$\mathbf{O}\left(n^{\frac{3}{4}} \log \frac{n}{\epsilon}\right)$	---	[?]
9	$\frac{t^2-1-\log t}{2} + \frac{t^{1-q}-1}{2(q-1)}, q > 1.$	$\mathbf{O}\left(qn^{\frac{q+1}{2q}} \log \frac{n}{\epsilon}\right)$	$\mathbf{O}\left(q^2 \sqrt{n} \log \frac{n}{\epsilon}\right)$	[4]
10	$\frac{p}{2}t^2 + e^{p\left(\frac{1}{t}-1\right)} - \left(1 + \frac{p}{2}\right), \quad p > 0.$	$\mathbf{O}\left(\sqrt{n} (\log n)^2 \log \frac{n}{\epsilon}\right)$	$\mathbf{O}\left(\sqrt{n} \log \frac{n}{\epsilon}\right)$	[5]
11	$\frac{t^2-1}{2} + \frac{4}{\pi p} \left[\tan^p\left(\frac{\pi}{2t+2}\right) - 1\right], \quad p \geq 2.$	$\mathbf{O}\left(pn^{\frac{p+2}{2(p+1)}} \log \frac{n}{\epsilon}\right)$	$\mathbf{O}\left(p^2 \sqrt{n} \log \frac{n}{\epsilon}\right)$	[6]

**Tableau 1** : quelque fonctions noyau les plus connus sur la programmation linéaire

# Bibliographie

- [1] Y. Q. Bai, C. Roos, *A primal-dual interior point method based on a new kernel function with linear growth rate*, in : Proceedings of the 9<sup>th</sup> Australian Optimization Day, Perth, Australia, (2002).
- [2] Y. Q. Bai, M. El Ghami, C. Roos, *A comparative study of kernel functions for primal-dual interior point algorithms in linear optimization*, SIAM Journal on Optimization, 15, 101–128, (2004).
- [3] M. Bierlaire, *Introduction à l'optimisation différentiable*, presses polytechniques et universitaires romandes (PPUR), (2005).
- [4] M. Bouafia, D. Benterki, A. Yassine, *An efficient parameterized logarithmic kernel function for linear optimization*. Optimization Letters 12 (5), (2018) 1079-1097.
- [5] M. Bouafia, D. Benterki, A. Yassine, *Complexity analysis of interior point methods for linear programming based on a parameterized kernel function*, RAIRO-Operations Research, (50) 935–949, (2016).
- [6] M. Bouafia, D. Benterki, A. Yassine, *An efficient primal-dual Interior Point Method for linear programming problems based on a new kernel function with a trigonometric barrier term*, Optimization Theory and Applications, (170)528–545 (2016).
- [7] M. Bouafia, *étude asymptotique des méthodes de points intérieurs pour la programmation linéaire*, Thèse de doctrat, Université du havre, France, (2016).
- [8] G. Ciarlet, *Introduction à l'analyse numérique matricielle et à l'optimisation*, Masson, Paris, (1985).

- [9] X. Z. Cai, G. Q. Wang, M. El Ghami, and Y. J. Yue, *Complexity analysis of primal-dual interior-point methods for linear optimization based on a new parametric kernel function with a trigonometric barrier term*, Abstract and Applied Analysis, 11 pages, Art. ID 710158, (2014).
- [10] G. B. Dantzig, *Linear Programming and extensions*, Princeton University Press, Princeton, N. J, 1963.
- [11] M. El Ghami, I. D. Ivanov, C. Roos, T. Steihaug, *A polynomial-time algorithm for LO based on generalized logarithmic barrier functions*, International Journal of Applied Mathematics, 21, 99–115, (2008).
- [12] M. El Ghami, Z. A. Guennoun, S. Bouali, T. Steihaug, *Interior point methods for linear optimization based on a kernel function with a trigonometric barrier term*, Journal of Computational and applied Mathematics, 236, 3613–3623, (2012).
- [13] N. K. Karmarkar, *A new polynomial-time algorithm for linear programming*, in : Proceedings of the 16<sup>th</sup> Annual ACM Symposium on Theory of Computing, 4, 373–395, (1984).
- [14] B. Kheirfam, M. Moslem, *A polynomial-time algorithm for linear optimization based on a new kernel function with trigonometric barrier term*, Yugoslav Journal of Operations Research, 25(2), 233–250, (2015).
- [15] J. Peng, C. Roos, T. Terlaky, *Self-regular functions and new search directions for linear and semidefinite optimization*, Mathematical Programming, 93, 129–171,(2002).
- [16] M. R. Peyghami, S. Fathi Hafshejani, L. Shirvani, *Complexity of interior-point methods for linear optimization based on a new trigonometric kernel function*, Journal of Computational and Applied Mathematics, 255, 74–85, (2014).
- [17] M. R. Peyghami, S. Fathi Hafshejani, L. Shirvani, *Complexity of interior-point methods for linear optimization based on a new trigonometric kernel function*, Journal of Computational and Applied Mathematics, 255, 74–85, (2014).

- [18] C. Roos, T. Terlaky, J. Ph. Vial, *Theory and algorithms for linear optimization, An Interior-Point Approach*, John Wiley & Sons, Chichester, UK, (1997).
- [19] C. Roos, T. Terlaky, J. Vial, *Optimization Theory and Algorithm for Linear Programming Optimization*, Princeton University, (2001).
- [20] N. Z. Shor, Utilization of the operation of space dilatation in the minimization of convex functions, *Kibernetika*, 1(1970), pp :6–12. Traduction anglaise : *Cybernetics*,6, pp :7–15.