

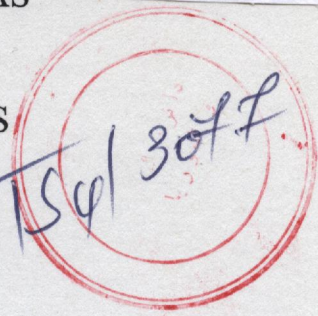


32-510-167-1

UNIVERSITE FERHAT ABBAS
SETIF

INSTITUT DE MATHÉMATIQUES

M E M O I R E



présenté par

KEBICHE Zakia

en vue de l'obtention du diplôme de

M A G I S T E R

Thème :

***MISE EN OEUVRE D'UNE METHODE
DE TRAJECTOIRE CENTRALE POUR LES PROBLEMES
COMPLEMENTAIRES LINEAIRES MONOTONES***



Jury

Dr . R . BENACEUR
Dr . A . KERAGHEL
Dr . M . ACHACHE
Dr . K . BELBAHRI

M .C univ. BATNA
C.C UFA. SETIF
C.C UFA. SETIF
C.C univ. BATNA

PRESIDENT
RAPPORTEUR
EXAMINATEUR
EXAMINATEUR

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE FERHAT ABBAS
SETIF

INSTITUT DE MATHEMATIQUES

M E M O I R E

présenté par

KEBICHE Zakia

en vue de l'obtention du diplôme de

M A G I S T E R

Thème :

***MISE EN OEUVRE D'UNE METHODE
DE TRAJECTOIRE CENTRALE POUR LES PROBLEMES
COMPLEMENTAIRES LINEAIRES MONOTONES***

Jury

Dr . R . BENACEUR

Dr . A . KERAGHEL

Dr . M . ACHACHE

Dr . K . BELBAHRI

M .C univ. BATNA

C.C UFA. SETIF

C.C UFA. SETIF

C.C univ. BATNA

PRESIDENT

RAPPORTEUR

EXAMINATEUR

EXAMINATEUR

RÉSUMÉ

Le problème complémentaire connu par ses différentes applications pratiques, constitue actuellement un modèle idéal pour le développement des algorithmes de point intérieur.

Dans ce type de méthodes, celles de trajectoire centrale sont particulièrement distinguées par leurs convergence polynomiale et asymptotiquement superlinéaire. De plus, comme procédures de résolution, elles sont de type Newton, ce qui devrait être en faveur de l'aspect numérique.

Malheureusement, les propriétés attractives de ces méthodes sont établies moyennant des hypothèses très restrictives pour l'implémentation numérique.

Nous présentons dans cette Thèse, un travail qui sort du cadre ordinaire des algorithmes prototypes (non programmables) au sens que l'on montre tout particulièrement comment on peut surmonter les difficultés cités ci - dessus.

A ce propos, nous avons pu mettre en oeuvre une procédure de trajectoire centrale dont le comportement numérique est mis en valeur à travers une comparaison à la méthode simpliciale de **Lemke**.

Le résultat de cette étude est un bon stimulant pour l'avenir du comportement numérique des méthodes en question.

MOTS CLÉS.

Méthodes de point intérieur, Méthodes de trajectoire centrale, Problème de complémentarité linéaire, Algorithme de pivotage de **Lemke**.

TABLE DES MATIÈRES

INTRODUCTION GÉNÉRALE

CHAPITRE I:

THÉORIE GÉNÉRALE ET RÉOLUTION CLASSIQUE DES PROBLÈMES COMPLÉMENTAIRES LINÉAIRES

I.0 - Introduction

I.1 - *PARTIE 1:*

THÉORIE GÉNÉRALE DE LA COMPLÉMENTARITÉ

I.1.1 - Définitions et notations

I.1.2 - Classifications des problèmes complémentaires linéaires (PCL)

I.1.3 - L'existence d'une solution de (PCL)

I.1.4 - Domaines d'application

I.2 - *PARTIE 2:*

RÉSOLUTION CLASSIQUE DU PROBLÈME COMPLÉMENTAIRE LINÉAIRE

I.2.1 - Présentation de la méthode de pivotage de **Lemke**

I.2.2 - Propriétés de la méthode

I.2.3 - Variantes de la méthode

CHAPITRE II:

MÉTHODES DE TRAJECTOIRE CENTRALE APPLIQUÉES AUX PROBLÈMES COMPLÉMENTAIRES LINÉAIRES

II.0 - Introduction

II.1 - Notations et hypothèses

II.2 - Présentation typique des méthodes de trajectoire centrale

II.3 - Méthode de trajectoire centrale avec poids

II.4 - Commentaires

CHAPITRE III:

SIMULATIONS NUMÉRIQUES

III.0 - Introduction

1 - Implémentation de l'algorithme basé sur une fonction barrière
logarithmique avec poids

III.2 - Comparaison avec l'algorithme de Lemke

III.3 - Exemples numérique

III.4 - Commentaires

CONCLUSION FINALE

ANNEXES:

*ANNEXE I - AUTRES MÉTHODES DE POINT INTÉRIEUR POUR
RÉSOLVRE (PCL)*

ANNEXE II - PRÉSENTATION DES PROGRAMMES

RÉFÉRENCES

INTRODUCTION GÉNÉRALE

Depuis son apparition, la complémentarité a attiré l'intérêt de plusieurs chercheurs. Son importance peut être mesurée par le rôle crucial qu'elle joue dans la résolution de plusieurs problèmes dans différents domaines (programmation linéaire (**PL**), programmation quadratique convexe (**PQC**), mécanique, ...). De plus, elle est considérée comme un modèle idéal pour le développement et l'analyse des algorithmes de point intérieur.

En effet, le problème complémentaire linéaire (**PCL**) généralise les deux créneaux de la programmation mathématique à savoir : la programmation linéaire et la programmation quadratique convexe.

On distingue deux types de méthodes pour résoudre le (**PCL**) : les méthodes simpliciales qui utilisent généralement la notion de pivots complémentaires où se pose le problème théorique concernant la complexité arithmétique exponentielle (la plus importante méthode est celle de pivotage de **Lemke**) et des méthodes de point intérieur qui font l'objet de la recherche dans le domaine de la programmation mathématique depuis (1984) (l'apparition de l'algorithme de **Karmarkar** pour la programmation linéaire).

Actuellement, on distingue trois classes importantes des méthodes de point intérieur à savoir : les méthodes affines, les méthodes de réduction du potentiel et celles de trajectoire centrale. Ces dernières sont les plus attractives, car d'une part, elles possèdent de bonnes propriétés théoriques, notamment la convergence à la fois polynomiale et superlinéaire, et d'autre part, les algorithmes sont essentiellement de type Newton ce qui devrait être en faveur des performances numériques. Néanmoins, les restrictions de ces méthodes ont conduit à un écart entre la théorie et la pratique qui reste jusqu'à présent difficile à réduire.

Notre étude (numérique) montre qu'avec ces restrictions, aucun algorithme de trajectoire centrale ne sera digne d'un comportement numérique acceptable, et que si l'on veut réussir une implémentation numérique, il est nécessaire d'éliminer

(de manière intelligente) ces restrictions, ou du moins, les plus gênantes sur le plan numérique tel que le critère de centralité.

A cet égard, nous avons implémenté au départ une version assez conforme à la théorie, au sens que l'on tient pas compte des restrictions en question. Nous avons constaté que l'algorithme ne converge pas à partir d'un point intérieur quelconque. Il faut donc absolument partir d'un point initial voisin de la trajectoire centrale. Pour cela, nous nous sommes inspirés des travaux de **J.Ding** et **T. Y. Li** pour remédier à cette difficulté. Ces derniers proposent une manière convenable pour la phase I moyennant une fonction barrière logarithmique avec poids.

Les résultats théoriques sont conservés, loin de là, le comportement numérique s'est merveilleusement amélioré surtout avec une recherche linéaire convenable et une certaine liberté au niveau de choix du paramètre barrière. Enfin, nous avons pris le soin de comparer ces résultats à ceux obtenus par l'algorithme simplicial de **Lemke**, histoire d'intégrer notre étude dans le cadre comparatif traditionnel entre les méthodes simpliciales et celles de point intérieur engagés depuis (1984).

Ceci nous a permis de conclure que les méthodes simpliciales restent encore les favorisées pour l'utilisateur, alors que les méthodes de point intérieur nécessitent davantage de développements susceptibles de conduire à un aspect numérique rivalisant : car ce n'est pas impossible.

PRÉSENTATION DE LA THÈSE:

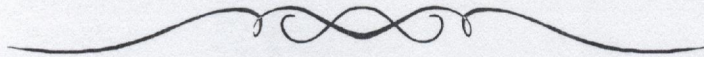


Cette Thèse est organisée comme suit :

Dans le premier chapitre, on présente la théorie générale et la résolution classique des problèmes complémentaires linéaires.

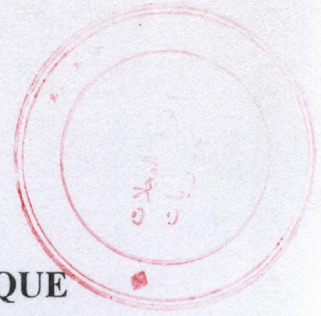
Le deuxième chapitre est consacré à faire une description générale des méthodes de trajectoire centrale appliquées aux problèmes complémentaires linéaires.

L'étude numérique de l'algorithme de pivotage de **Lemke** et d'un algorithme de trajectoire centrale basé sur une fonction barrière logarithmique avec poids est introduite dans le dernier chapitre.



CHAPITRE I

THÉORIE GÉNÉRALE ET RÉOLUTION CLASSIQUE DES PROBLÈMES COMPLÉMENTAIRES LINÉAIRES



I.0 - INTRODUCTION

I.1 - PARTIE 1:

THÉORIE GÉNÉRALE DE LA COMPLÉMENTARITÉ

I.1.1 - Définitions et notations

I.1.2 - Classification des problèmes complémentaires linéaires (PCL)

I.1.3 - L'existence d'une solution de (PCL)

I.1.4 - Domaines d'application

I.1.4.1 - Programmation linéaire (PL)

I.1.4.2 - Programmation quadratique convexe (PQC)

I.1.4.3 - Inéquations variationnelles

I.1.4.4 - Complémentarité et mécanique

I.1.4.5 - Autres domaines

I.2 - PARTIE 2:

RÉSOLUTION CLASSIQUE DU PROBLÈME COMPLÉMENTAIRE LINÉAIRE

I.2.1 - Présentation de la méthode de pivotage de Lemke

I.2.1.1 - Notions de base

I.2.1.2 - Description technique de la méthode

I.2.1.3 - Algorithme de pivotage de **Lemke**

I.2.1.4 - Convergence de la méthode

I.2.1.5 - Résultats concernant la programmation linéaire (**PL**) et la
programmation quadratique convexe (**PQC**)

I.2.2 - Propriétés de la méthode

I.2.3 - Variantes de la méthode

I.0 - INTRODUCTION

L'étude de la complémentarité a commencé par les travaux de **R . W . Cottle** publiés dans la période 1964 - 1966.

Ce chapitre contient essentiellement l'étude théorique générale des problèmes complémentaires (partie 1). De plus, on présente une méthode de résolution classique (partie 2).

I.1 - PARTIE 1:

THÉORIE GÉNÉRALE DE LA COMPLÉMENTARITÉ

I.1.1 - Définitions et notations.

Soit \mathbb{R}^n l'espace vectoriel réel .

$\mathbb{R}_+^n = \{x \in \mathbb{R}^n / x \geq 0\}$ l'orthant positif de \mathbb{R}^n .

$f : \mathbb{R}_+^n \rightarrow \mathbb{R}^n$ une fonction donnée.

Le problème complémentaire associé à f est défini par :

$$(PC) \begin{cases} \text{Trouver } x \in \mathbb{R}^n \text{ tel que :} \\ x \geq 0, f(x) \geq 0 \text{ et } \langle x, f(x) \rangle = 0 \end{cases}$$

Dans le cas où f est une fonction affine, le problème complémentaire devient linéaire et on le note par **(PCL)**. Il s'écrit sous plusieurs formes (générale, horizontale, verticale, standard, ...) suivant la forme de f .

En général, pour la forme la plus simple, le problème complémentaire linéaire consiste à chercher un couple (x, y) de n -vecteurs satisfaisant la contrainte linéaire $y = Mx + q$, les contraintes de non négativité $(x, y) \geq 0$ et les conditions dites de complémentarité $x_i y_i = 0$, pour $i = 1, \dots, n$.

Donc, le **(PCL)** est de la forme :

$$(PCL) \begin{cases} \text{Trouver } (x, y) \in \mathbb{R}^{2n} \text{ tel que :} \\ y = Mx + q \\ (x, y) \geq 0 \text{ et } x^t y = 0 \end{cases}$$

où : $M \in \mathbb{R}^{n \times n}$ et $q \in \mathbb{R}^n$.

Remarque.

Si f est une fonction quelconque (non affine), le problème **(PC)** est dit problème complémentaire non linéaire et on le note **(PCNL)**.

Définitions.

1/ Une solution du problème **(PC)** est dite non dégénérée si le vecteur $(x, f(x))$ possède au moins n composantes non nulles (le vecteur $(x, f(x))$ a $2n$ composantes).

Si x est une solution du problème **(PC)** non dégénérée et si on pose $y = f(x)$ alors : $A = \{ i / x_i > 0 \}$, $B = \{ i / y_i > 0 \}$ sont complémentaires et non vides.

D'où le nom de la complémentarité.

2/ Le problème **(PC)** est dit monotone si : $(f(x) - f(y))^t (x - y) \geq 0, \forall x, y \in \mathbb{R}^n$.

Dans le cas où f est une fonction affine : $f(x) = Mx + q$, le **(PCL)** est dit monotone si : $(x - y)^t M (x - y) \geq 0$.

C'est à dire si M est semi - définie positive alors le **(PCL)** est monotone.

I.1.2 - Classification des (PCL).

Les **(PCL)** sont typiquement classifiés conformément aux classes de la matrice M . A ce propos, on donne les définitions suivantes :

- Matrice définie positive (DP).

M est semi - définie positive **(SDP)** si : $x^t Mx \geq 0, \forall x \in \mathbb{R}^n$.

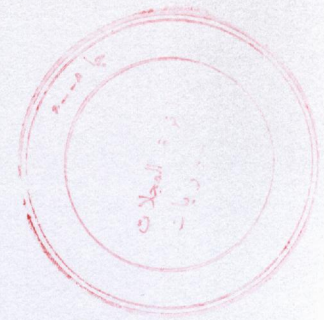
M est définie positive **(DP)** si : $x^t Mx > 0, \forall x \in \mathbb{R}^n, x \neq 0$.

- Matrice copositive (CP).

M est copositive **(CP)** si : $x^t Mx \geq 0, \forall x \geq 0$.

M est strictement copositive **(SCP)** si : $x^t Mx > 0, \forall x \geq 0$ et $x \neq 0$.

M est copositive plus **(CPP)** si :



- M est copositive (**CP**).

- Si : $x^t M x = 0, \forall x \geq 0$ et $x \neq 0 \Rightarrow (M + M^t) x = 0$.

- **P** - matrice.

M est **P**₀ - matrice si tous les mineurs principaux sont positifs.

M est **P** - matrice si tous les mineurs principaux sont strictement positifs.

-**P**_{*}- matrice.

M est **P**_{*}(**k**) - matrice si : $(1 + 4k) \sum_{i \in I_+(x)} x_i [Mx]_i + \sum_{i \in I_-(x)} x_i [Mx]_i \geq 0$

pour tout $x \in \mathbb{R}^n$ et $k \geq 0$, où :

$$I_+(x) = \{ i / x_i [Mx]_i > 0, i = 1, \dots, n \}$$

$$I_-(x) = \{ i / x_i [Mx]_i < 0, i = 1, \dots, n \}$$

$$P_* = \bigcup_{k \geq 0} P_*(k).$$

- **Matrice adéquate (A)**.

M est une matrice adéquate (**A**) si :

- M est **P**₀ - matrice.

- Si le déterminant d'une sous - matrice principale B est zéro alors les lignes et les colonnes de M qui contiennent B sont linéairement dépendantes.

- **Matrice semi - monotone (SM)**.

M est semi - monotone (**SM**) si : $\forall x \in \mathbb{R}^n, x \neq 0, \exists k$ indice tel que : $x_k > 0$ et $M_k x \geq 0$.

M est strictement semi - monotone (**SSM**) si : $\forall x \geq 0$ et $x \neq 0, \exists k$ indice tel que : $x_k > 0$ et $M_k x > 0$.

- **L** - matrice.

M est **L** - matrice si :

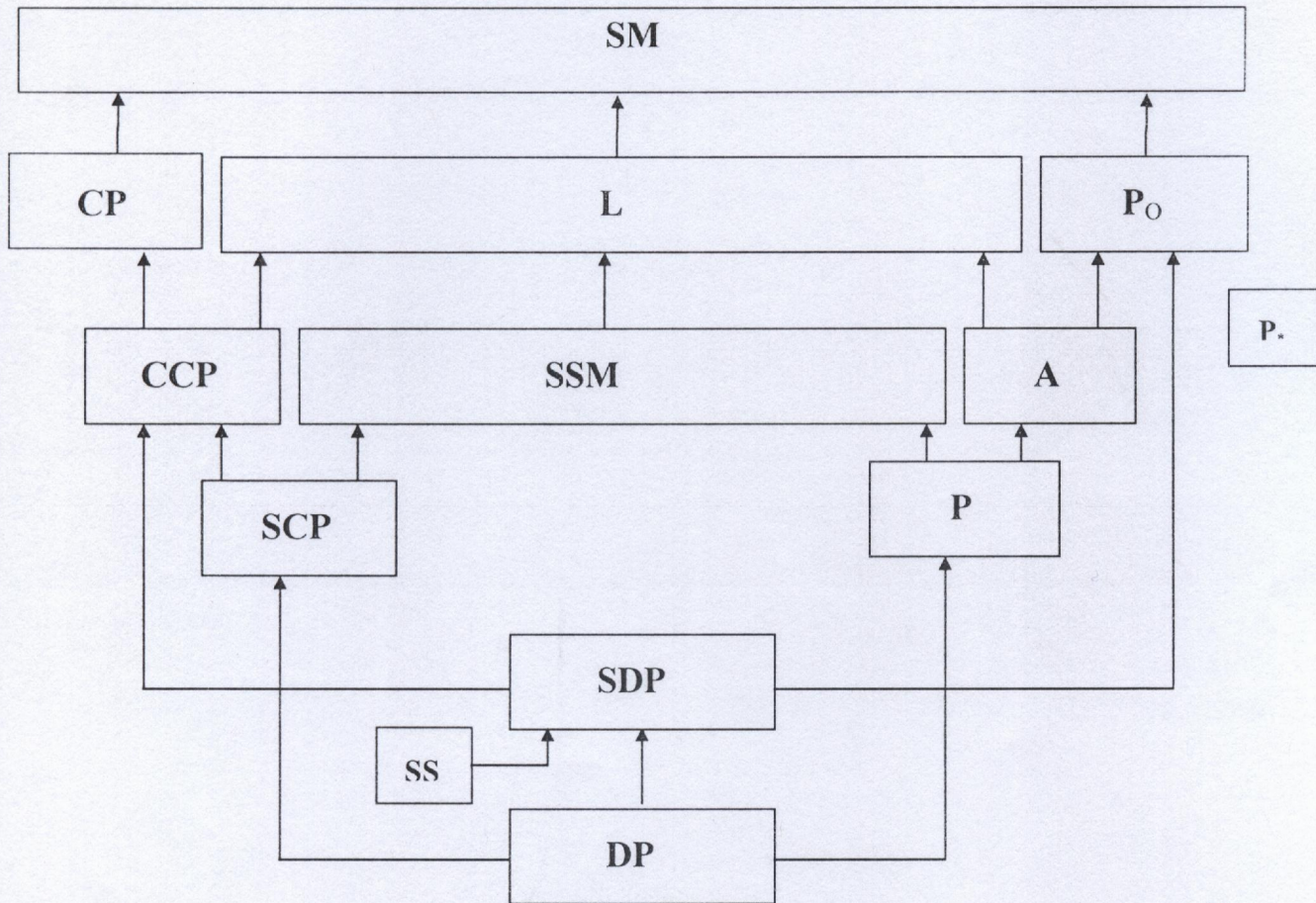
- M est semi - monotone (**SM**).

- Pour quelques $x \geq 0$ et $x \neq 0, Mx \geq 0$ et $x^t Mx = 0$, il existe deux matrices diagonales $\Lambda \geq 0$ et $\Omega \geq 0$ tels que : $\Omega x \neq 0$ et $(\Lambda M + M^t \Omega) x = 0$.

- *Matrice anti - symétrique (SS).*

M est skew - symétrique (SS) si : $x^t M x = 0, \forall x \in \mathbb{R}^n$.

La conclusion des relations de ces classes de matrices est résumée dans le tableau suivant :



Remarque.

La plus importante classe est celle des (PCL) monotones pour laquelle la matrice M est semi - définie positive. Elle couvre bien évidemment la programmation linéaire (PL) et la programmation quadratique convexe (PQC).

I.1.3 - Existence d'une solution pour le (PCL).

L'existence d'une solution pour le (PCL) est garantie seulement pour des classes spéciales de la matrice M .

Dantzig et **Cottle** ont montré que le (PCL) admet une solution si M est une P - matrice et un q vecteur arbitraire . Tandis que **Lemke** a montré que si M est (CCP), le (PCL) admet une solution pour un vecteur q arbitraire.

Récemment, plusieurs résultats concernant l'existence d'une solution sont obtenus en généralisant les classes obtenues par **Dantzig** , **Cottle** et **Lemke** : le (PCL) admet une solution si la matrice M appartient à l'une des classes suivantes où le vecteur q est arbitraire :

strictement semi - monotone (SSM), L - matrice , définie positive (DP),
strictement copositive (SCP),... .

L'unicité de la solution est vérifiée si M est une P - matrice.

I.1.4 - Domaines d'application.

I.1.4.1 - programmation linéaire.

Considérons les programmes linéaires duaux :

$$(PL) \quad \begin{cases} \min c x \\ Ax \geq b \\ x \geq 0 \end{cases} \quad (D) \quad \begin{cases} \max b y \\ A y \leq c \\ y \geq 0 \end{cases}$$

avec A : matrice de type (m, n) et de plein rang.

c : vecteur colonne de \mathbb{R}^n et b vecteur colonne de \mathbb{R}^m .

Comme le problème (PL) est convexe et les contraintes sont linéaires alors les conditions de **KKT** sont nécessaires et suffisantes.

Donc \bar{x} est une solution optimale de (PL) ssi : $\exists \bar{y} \in \mathbb{R}^m$, $\bar{z} \in \mathbb{R}^n$ tel que :

$$\begin{cases} c - A^t \bar{y} - \lambda = 0 \\ \bar{y}^t (b - A\bar{x}) = 0 \\ -\lambda^t \bar{x} = 0 \end{cases}$$

$$\Leftrightarrow \begin{cases} \lambda = c - A^t \bar{y} \\ v = A\bar{x} - b \\ -\bar{y}^t v = 0, \lambda^t \bar{x} = 0 \\ \bar{x} \geq 0, \bar{y} \geq 0, \lambda \geq 0, v \geq 0 \end{cases}$$

$$\Leftrightarrow \begin{cases} \begin{bmatrix} \lambda \\ v \end{bmatrix} = \begin{bmatrix} 0 & -A^t \\ A & 0 \end{bmatrix} \begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} + \begin{bmatrix} c \\ -b \end{bmatrix} \\ \left\langle \begin{pmatrix} \lambda \\ v \end{pmatrix}, \begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix} \right\rangle = 0 \\ \begin{pmatrix} \lambda \\ v \end{pmatrix} \geq 0, \begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix} \geq 0 \end{cases}$$

Si on pose :

$$w = \begin{pmatrix} \lambda \\ v \end{pmatrix}, \quad z = \begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix}, \quad q = \begin{pmatrix} c \\ -b \end{pmatrix} \quad \text{et} \quad M = \begin{bmatrix} 0 & -A^t \\ A & 0 \end{bmatrix}$$

Les conditions de **KKT** sont donc équivalentes au problème complémentaire linéaire suivant :

$$(PCL) \begin{cases} \text{Trouver } z \in \mathbb{R}^{n+m} \text{ tel que :} \\ z \geq 0, w = Mz + q \geq 0 \text{ et } w^t z = 0 \end{cases}$$

Remarques.

1/ La résolution de **(PCL)** associé à un programme linéaire **(PL)** nous fournit le couple de solutions optimales (\bar{x}, \bar{y}) de **(PL)** et **(D)** en même temps.

2/ Puisque $M = \begin{bmatrix} 0 & -A^t \\ A & 0 \end{bmatrix}$, on a toujours $z^t M z = 0, \forall z \in \mathbb{R}^n$.

C'est à dire M est **(SDP)**. Donc, le **(PCL)** associé à un **(PL)** est toujours monotone.

I.1.4.2 - programmation quadratique convexe .

Soit le programme quadratique convexe :

$$(PQC) \quad \begin{cases} \min (1/2) x^t H x + c^t x \\ A x \leq b \\ x \geq 0 \end{cases}$$

avec $c \in \mathbb{R}^n$ et $b \in \mathbb{R}^m$.

A : matrice de type (m, n) et de plein rang.

H : matrice carrée d'ordre n , symétrique et semi - définie positive.

Pour résoudre le problème **(PQC)**, on applique les conditions de **KKT** qui sont nécessaires et suffisantes dans ce cas :

$\bar{x} \in \mathbb{R}_+^n$ est une solution optimale de **(PQC)** ssi $\exists \bar{y} \in \mathbb{R}_+^m, \lambda \in \mathbb{R}_+^n$ tel que :

$$\begin{cases} c + H\bar{x} + A^t \bar{y} - \lambda = 0 \\ \bar{y}^t (A\bar{x} - b) = 0 \\ -\lambda^t \bar{x} = 0 \end{cases} \Leftrightarrow \begin{cases} \lambda = c + A^t \bar{y} + H\bar{x} \geq 0 \\ v = b - A\bar{x} \geq 0 \\ \bar{y}^t v = 0, \lambda^t \bar{x} = 0 \\ \bar{x} \geq 0, \bar{y} \geq 0, \lambda \geq 0, v \geq 0 \end{cases}$$

$$\text{Soient : } M = \begin{bmatrix} H & A^t \\ -A & 0 \end{bmatrix}, \quad q = \begin{bmatrix} c \\ b \end{bmatrix}, \quad w = \begin{bmatrix} \lambda \\ v \end{bmatrix}, \quad z = \begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix}$$

La résolution des équations de **KKT** est équivalente à celle de **(PCL)** suivant :

$$\left\{ \begin{array}{l} \text{Trouver } z \in \mathbb{R}^{n+m} \text{ tel que :} \\ z \geq 0, w = Mz + q \geq 0, w^t z = 0 \end{array} \right.$$

Remarques.

1/ On a : $M = \begin{bmatrix} H & A^t \\ -A & 0 \end{bmatrix}$, donc $z^t M z = z^t H z \geq 0, \forall z \in \mathbb{R}^n$ (H est **SDP**).

Le **(PCL)** associé à un **(PQC)** est monotone.

2/ Si M est une matrice **SDP** alors le **(PCL)** est équivalent à un programme quadratique convexe de la forme :

$$\left\{ \begin{array}{l} \min z^t (Mz + q) \\ Mz + q \geq 0 \\ z \geq 0 \end{array} \right.$$

dans le sens que (x, y) est une solution de **(PCL)** ssi elle est une solution minimale (optimale) du problème précédent avec la valeur zéro de l'objectif.

I.1.4.3 - les inéquations variationnelles.

Soit C un ensemble convexe, fermé et non vide de \mathbb{R}^n .

Le problème variationnel est de la forme :

$$(P) \left\{ \begin{array}{l} \text{Trouver } \bar{x} \in \mathbb{R}^n \text{ tel que :} \\ \bar{x} \in C, \langle x - \bar{x}, f(\bar{x}) \rangle \geq 0, \forall x \in C \end{array} \right.$$

Si de plus C est un cône, le problème variationnel est équivalent au problème :

$$(P') \left\{ \begin{array}{l} \text{Trouver } \bar{x} \in \mathbb{R}^n \text{ tel que :} \\ \bar{x} \in C, f(\bar{x}) \in C^+ \text{ et } \langle \bar{x}, f(\bar{x}) \rangle = 0 \end{array} \right.$$



où $C^+ = \{ y \in \mathbb{R}^n \mid \langle x, y \rangle \geq 0, \forall x \in C \}$ est le cône dual de C .

Dans le cas où $C = \mathbb{R}_+^n$, le problème (p') est équivalent au problème de complémentarité suivant :

$$\begin{cases} \text{Trouver } \bar{x} \in \mathbb{R}^n \text{ tel que :} \\ \bar{x} \geq 0, f(\bar{x}) \geq 0 \text{ et } \langle \bar{x}, f(\bar{x}) \rangle = 0 \end{cases}$$

I.1.4.4 - complémentarité et mécanique .

Plusieurs problèmes de mécanique peuvent être transformés en problèmes de complémentarité, citons par exemple :

- Mécanique des fluides et les problèmes à frontières libres.
- La théorie des plaques minces.

I.1.4.5 - Autres domaines.

Il existe d'autres domaines où s'applique la complémentarité tels que :

- La théorie des jeux.
- La théorie de min - max.

I.2 - PARTIE 2: RÉOLUTION DE (PCL).

Soit le problème complémentaire linéaire suivant :

$$(PCL) \quad \begin{cases} \text{trouver } z \in \mathbb{R}^n \text{ tel que:} \\ z \geq 0, Mz + q \geq 0 \text{ et } z^t(Mz + q) = 0 \end{cases}$$

où M est une matrice carrée d'ordre n et q un vecteur colonne de \mathbb{R}^n .

Parmi les méthodes de résolution de (PCL), on présente une méthode classique proposée par **C. E. Lemke** en 1969.

Il s'agit de la méthode de pivotage qui est une variante de la méthode simpliciale. Elle utilise la notion de pivots complémentaires, c'est-à-dire, la variable qui entre en base (x) et celle qui quitte la base (y) doivent vérifier la condition de complémentarité : $x \cdot y = 0$.

I.2.1 - Présentation de la méthode de pivotage de Lemke.

I.2.1.1 - Notions de base.

On met (PCL) sous la forme équivalente :

$$(P) \quad \begin{cases} \text{Trouver } w, z \in \mathbb{R}^n \text{ tel que :} \\ w - Mz = q & (1) \\ w_i \geq 0, z_i \geq 0 \text{ pour } i = 1, \dots, n & (2) \\ w_i z_i = 0 \text{ pour } i = 1, \dots, n & (3) \end{cases}$$

1/ (w_i, z_i) est un couple de composantes complémentaires.

2/ La composante w_i (resp. z_i) est dite variable de base si $w_i > 0$ (resp. $z_i > 0$).

3/ Si $w_i = 0$ (resp. $z_i = 0$) on dit que w_i (resp. z_i) est une variable hors base.

4/ Un couple (w, z) est dit solution de base réalisable complémentaire s'il vérifie les conditions suivantes :

- a) (w, z) est une solution de base réalisable de (1) et (2).
- b) une composante du couple (w_i, z_i) est une variable de base ($i = 1, \dots, n$).

Soit maintenant le problème suivant :

$$(P') \begin{cases} \text{trouver } z \in \mathbb{R}^n \text{ tel que:} \\ w - Mz - z_0 e_n = q & (4) \\ w_i \geq 0, z_i \geq 0, z_0 \geq 0 \text{ pour } i = 1, \dots, n & (5) \\ w_i \cdot z_i = 0 \text{ pour } i = 1, \dots, n & (6) \end{cases}$$

z_0 est une valeur positive de \mathbb{R} ($z_0 \geq 0$) et e_n un vecteur colonne de \mathbb{R}^n .

5/ Une solution (w, z, z_0) du problème (P') est dite solution de base réalisable presque complémentaire si elle vérifie :

- a) (w, z, z_0) est une solution de base réalisable (vérifie (4) et (5)).
- b) ni w_s ni z_s est une variable de base pour un certain $s \in \{1, \dots, n\}$.
- c) z_0 est une variable de base et on a soit w_i soit z_i une variable de base pour tout $i \in \{1, \dots, n\}$ tel que : $i \neq s$.

6/ L'ensemble des points $\{ \bar{x} + \lambda d \mid \lambda \geq 0 \}$ est une demie ligne non bornée obtenue en se déplaçant de \bar{x} dans la direction d .

Cet ensemble est appelé le rayon de d .

1.2.1.2 - Description technique de la méthode.

Etant donnée une matrice carrée M d'ordre n et un vecteur colonne q de \mathbb{R}^n . La méthode de pivotage de **Lemke** consiste à résoudre le problème suivant:

$$(I) \begin{cases} \text{Trouver } z \in \mathbb{R}^n \text{ tel que:} \\ w - Mz = q & (1) \\ w_i \geq 0, z_i \geq 0 \text{ pour } i = 1, \dots, n & (2) \\ w_i \cdot z_i = 0 \text{ pour } i = 1, \dots, n & (3) \end{cases}$$

Si $q \geq 0$: on obtient immédiatement une solution qui satisfait le système défini par (1), (2) et (3) en posant : $w = q, z = 0$.

Donc $(w, q) = (q, 0)$ est une solution de base réalisable complémentaire.

Dans le cas contraire, c'est-à-dire, s'il existe au moins un indice i tel que : $q_i < 0$, afin d'obtenir une base réalisable du système (I), on introduit une variable artificielle z_0 associée au vecteur colonne $(-e_n)$ tel que toutes ses composantes soient égales à (-1).

Alors on a le système suivant :

$$(II) \begin{cases} \text{trouver } z \in \mathbb{R}^n \text{ tel que :} \\ w - Mz - z_0 e_n = q & (4) \\ w_i \geq 0, z_i \geq 0, z_0 \geq 0 \text{ pour } i = 1, \dots, n & (5) \\ w_i z_i = 0 \text{ pour } i = 1, \dots, n & (6) \end{cases}$$

où : $z_0 = \max \{ -q_i \mid i = 1, \dots, n \}$.

Le choix de z_0 nous fournit alors la solution du système (II) donnée par :

$$z = 0, w = q + z_0 e_n \geq 0.$$

Remarque.

Pour obtenir une solution de base réalisable complémentaire du système (I) lorsqu'elle existe, la logique suivie consiste à ramener la variable artificielle z_0 à zéro par une série de pivots en résolvant le système (II).

I.2.1.3 - L'algorithme de pivotage de Lemke.

Description.

L'algorithme démarre de la solution initiale : $z = 0, w = q + z_0 e_n$, à travers une série de pivots, il produit une suite de solutions de base réalisables presque complémentaires (w, z, z_0) conformément au tableau :

w	z	z_0	Le second membre
I	-M	$-e_n$	q

$$w \geq 0, z \geq 0, z_0 \geq 0$$

Dans chaque itération k, la base se change en faisant entrer dans le vecteur de base exactement une variable hors base dite variable entrée.

Supposons que (y_1, \dots, y_n) sont les variables de base à l'itération k et soit x_s la variable qui va entrer en base.

On a le tableau suivant :

variables de base	y_1, \dots, y_{n-1}, y_n	x_s	les autres variables	le second membre
y_1	1 0 ... 0 0 0	\bar{a}_{1s}	.	\bar{q}_1
.	0 1 ... 0 0 0	\bar{a}_{2s}	.	\bar{q}_2
.
.	0 0 ... 0 1 0	.	.	.
y_n	0 0 ... 0 0 1	\bar{a}_{ns}	.	\bar{q}_n

Puisque la variable qui va entrer en base est x_s , alors on garde toutes les variables hors base différentes de x_s égales à zéro et on donne la valeur λ à x_s , on a :

$$(*) \begin{cases} x_s = \lambda \\ y_i = q_i - \lambda \bar{a}_{is} & \text{pour } i = 1, \dots, n \\ \text{toutes les autres variables sont nulles} \end{cases}$$

On a les deux alternatives suivantes :

a/ Si $\bar{a}_{is} \leq 0, \forall i = 1, \dots, n$ alors $y_i \geq 0, \forall i = 1, \dots, n, \forall \lambda \geq 0$.

comme λ varie de 0 à $+\infty$, cette solution trace une demie ligne non bornée (un rayon) de l'ensemble des solutions réalisables de (*).

b/ S'il existe un indice i tel que $\bar{a}_{is} > 0$ et si on veut que la solution dans (*) reste non négative (réalisable) alors la valeur que λ doit prendre est :

$$\lambda = \min \left\{ \frac{\bar{q}_i}{\bar{a}_{is}} \mid \bar{a}_{is} > 0 \right\}$$

Remarques.

1/ L'algorithme de pivotage se termine toujours soit avec une solution de base réalisable complémentaire soit avec une demie ligne non bornée (rayon de termination).

2/ Lorsqu'un pivot est obtenu avec le vecteur colonne associé à z_0 comme colonne de pivotage et la r -ième ligne comme ligne de pivotage, alors le second membre devient non négatif ($q_r = \min \{ q_i, i = 1, \dots, n \}$) et le vecteur $(w_1, \dots, w_{r-1}, z_0, w_{r+1}, \dots, w_n)$ sera le vecteur de base initial.

3/ Tous les vecteurs de base obtenus dans l'algorithme sauf peut être le dernier vecteur de base sont des vecteurs de base réalisables presque complémentaires.

Algorithme.**Début d'algorithme:**

(Données : M : matrice carrée d'ordre n et semi - définie positive .

q : vecteur colonne de \mathbb{R}^n).

Initialisation:

Si : $q \geq 0$ Stop .

$(w, z) = (q, 0)$ est une solution de base réalisable complémentaire.

Si non : Trouver s tel que : $q_s = \min \{ q_i, i = 1, \dots, n \}$.

On fait le pivotage entre la ligne s comme ligne de

pivotage et la colonne de z_0 comme colonne de pivotage.

Donc w_s quitte la base et z_0 le remplace.

Ainsi z_0 et $(w_i, i = 1, \dots, n$ avec $i \neq s)$ sont non négatifs.

Posons : $y_s = z_s$

Itération:**Etape1:**

Soit d_s le vecteur colonne correspondant à la variable y_s

Si : $d_s \leq 0$, faire 4.

Si non : on détermine l'ensemble I tel que : $I = \{i / d_s [i] > 0\}$, ensuite,

on choisit un indice r tel que :
$$\frac{\bar{q}_r}{d_s[r]} = \min_{i \in I} \left\{ \frac{\bar{q}_i}{d_s[i]} \right\}$$

où \bar{q} : est le vecteur colonne du second membre.

Si : la variable de base dans la ligne r est z_0 , faire 3.

Si non : faire 2.

Etape2:

La variable de base dans la ligne r est soit w_k ou z_k pour un certain $k \neq s$.

Donc on fait le pivotage entre la ligne r et la colonne de y_s .

Si la variable qui quitte la base est w_k (resp. z_k) on pose $y_s = z_k$ (resp. $y_s = w_k$) et on retourne à l'étape 1.

Etape3:

On fait le pivotage entre la ligne de z_0 et la colonne de y_s . Ainsi la variable z_0 quitte le vecteur de base. On obtient une solution de base réalisable complémentaire.

Etape4:

Puisque $d_s \leq 0$, on s'arrête avec un rayon de termination :

$$\mathfrak{R} = \{(w, z, z_0) + \lambda d \mid \lambda \geq 0\}.$$

Ce rayon est déterminé tel que tous ses points satisfassent le système (II) défini par (4), (5) et (6).

(w, z, z_0) est une solution de base réalisable presque complémentaire finale.

d : est une direction extrémale dont toutes les composantes sont définis par :

$$\begin{cases} 1 & \text{: pour la ligne correspondante à } y_s \\ -d_s & \text{: pour les lignes correspondantes aux variables de base} \\ 0 & \text{: ailleurs} \end{cases}$$

Fin de l'algorithme.

Remarque.

L'algorithme se termine dans les deux cas suivants :

- a/ La variable z_0 quitte le vecteur de base ou bien devient nulle dans la solution.
- b/ z_0 peut être strictement positive dans la solution et la colonne de pivotage peut être encore non positive, dans ce cas l'algorithme se termine avec un rayon de termination \mathfrak{R} .

I.2.1.4 - Convergence de la méthode.

Sous les conditions de non dégénérescence, l'algorithme s'arrête au bout d'un nombre fini d'étapes avec ou bien une solution de base réalisable complémentaire (SBRC) ou un rayon de termination \mathfrak{R} .

Dans le cas général, si un rayon de termination est introduit dans l'algorithme, il est possible que le (PCL) possède une solution mais l'algorithme soit incapable de la trouver.

Cependant, si M vérifie la copositivité plus, le rayon de termination apparaît seulement dans le cas où le (PCL) n'a pas de solution.

Théorème de convergence.

Supposons que chaque solution de base réalisable presque complémentaire du système (II) définie par (4), (5) et (6) est non dégénérée, alors l'algorithme s'arrête après un nombre fini d'étapes.

De plus si M est copositive plus et le système défini par (1) et (2) est consistant (il existe au moins un point qui vérifie (1) et (2)) alors l'algorithme s'arrête avec une (SBRC).

Réciproquement, lorsque M est copositive plus et si l'algorithme se termine avec un rayon de termination alors le (PCL) n'a pas de solution (le système défini par (1) et (2) est non réalisable).

Remarque :

On peut dire que l'algorithme de pivotage résout le (PCL) toutes les fois que M est copositive plus, et dans ce cas, l'algorithme s'arrête avec une (SBRC) ou se termine avec un rayon de termination \mathfrak{R} qui implique que le (PCL) n'a pas de solution.



1.2.1.5 - Résultats concernant la (PL) et la (PQC).

a - Programmation linéaire.

Soit le programme linéaire :

$$\begin{cases} \min c^t x \\ Ax \geq b \\ x \geq 0 \end{cases}$$

A : matrice de type (m, n) et de plein rang . $b \in \mathbb{R}^m$ et $c \in \mathbb{R}^n$.

On lui associe le (PCL) suivant :

$$\begin{cases} \text{trouver } z \in \mathbb{R}^{n+m} \text{ tel que :} \\ z \geq 0, w = Mz + q \geq 0 \text{ et } w^t z = 0 \end{cases}$$

où

$$M = \begin{bmatrix} 0 & -A^t \\ A & 0 \end{bmatrix}, q = \begin{bmatrix} c \\ -b \end{bmatrix}, z = \begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix}$$

Résultats.

1/ Dans les (PCL) associés aux programmes linéaires (PL), la matrice M est toujours copositive plus puisque :

- $\forall z \in \mathbb{R}^n$ on a : $z^t M z = 0$, donc : M est copositive ($\forall z \geq 0, z^t M z \geq 0$).

- De plus si : $z^t M z = 0, \forall z \geq 0$, alors $(M + M^t)z = 0$.

2/ Si l'algorithme de pivotage appliqué à un (PCL) (associé au (PL)) se termine avec un rayon de termination , alors le (PL) doit être ou bien non réalisable ou s'il est réalisable, la fonction objective est non bornée dans l'ensemble des solutions réalisables.

b - programmation quadratique convexe.

Considérons le programme quadratique convexe :

$$\begin{cases} \min c^t x + \frac{1}{2} x^t H x \\ Ax \leq b \\ x \geq 0 \end{cases}$$

A : matrice de type (m, n) et de plein rang.

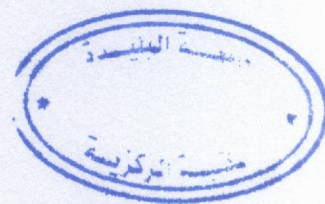
$b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ et H matrice carrée d'ordre n, symétrique et semi - définie positive.

Le (PCL) associé à (PQC) est de la forme :

$$\begin{cases} \text{trouver } z \in \mathbb{R}^{n+m} \text{ tel que :} \\ z \geq 0, w = Mz + q \geq 0 \text{ et } w^t z = 0 \end{cases}$$

où

$$M = \begin{bmatrix} H & A^t \\ -A & 0 \end{bmatrix}, q = \begin{bmatrix} c \\ b \end{bmatrix}, z = \begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix}$$



Résultats.

1/ La matrice M est toujours copositive plus pour un programme quadratique convexe puisque :

- M est SDP (du fait que H l'est aussi), donc elle est copositive.
- De plus si : $z^t M z = 0, \forall z \geq 0$ alors $(M + M^t) z = 0$.

2/ Si l'algorithme de pivotage appliqué à un (PCL) (associé à (PQC)) se termine avec un rayon de termination, alors le (PQC) doit être ou bien non réalisable ou s'il est réalisable, la fonction objective est non bornée dans l'ensemble des solutions réalisables.

I.2.2 - Propriétés de la méthode.

a. l'inefficacité théorique de cet algorithme est traduite par sa complexité arithmétique exponentielle (2^n) donnée par le nombre total de pivots que nécessite l'algorithme au pire des cas.

En pratique, le nombre d'itérations enregistré lors de la mise en oeuvre de l'algorithme est nettement inférieur à celui donné en théorie.

b. Cet algorithme est capable de résoudre les problèmes de la programmation linéaire (PL) et la programmation quadratique convexe (PQC). Il nous fournit les solutions primales et duales en même temps si elles existent (dans ce cas, la matrice M est toujours copositive plus).

Malheureusement, on est obligé de travailler dans \mathbb{R}^{n+m} au lieu de \mathbb{R}^n , donc le coût de l'algorithme de pivotage de Lemke augmente avec les dimensions des problèmes.

c. La dégénérescence pose un problème de convergence à l'algorithme de pivotage. L'exemple suivant (dû à M. M. Kostreva) le prouve.

$$M = \begin{pmatrix} 1 & 2 & 0 \\ 0 & 1 & 2 \\ 2 & 0 & 1 \end{pmatrix}, \quad q = \begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix}$$

on appliquant l'algorithme de pivotage, on obtient les vecteurs de base réalisables presque complémentaires suivants:

le vecteur de base initial est (z_0, w_2, w_3) suivi par (z_0, z_1, w_3) , (z_0, z_2, w_3) ,

(z_0, z_2, w_1) , (z_0, z_3, w_1) , (z_0, z_3, w_2) , (z_0, z_1, w_2) et (z_0, z_1, w_3) dans cet ordre.

Puis on retrouve le vecteur initial (z_0, z_1, w_3) , et l'algorithme cycle indéfiniment.

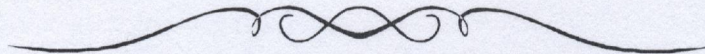
I.2.3 - Variantes de la méthode.

Dans l'algorithme de pivotage, on choisit toujours le vecteur colonne original associé à z_0 est $(-e_n)$.

Afin d'éviter les problèmes rencontrés dans l'algorithme de pivotage, deux variantes se présentent .

L'idée utilisée dans la première variante est de remplacer le vecteur (e_n) par un autre vecteur $d > 0$. Dans la deuxième variante et sans introduire la variable artificielle z_0 , l'idée est d'utiliser une colonne de M mais sous la condition que toutes ses composantes soient strictement positives.

Malheureusement, on rencontre les mêmes problèmes qui figurent dans l'algorithme de pivotage.



CHAPITRE II

MÉTHODES DE TRAJECTOIRE CENTRALE APPLIQUÉES AUX PROBLÈMES COMPLÉMENTAIRES LINÉAIRES

II.0 - Introduction

II.1 - Notations et hypothèses

II.2 - Présentation typique des méthodes de trajectoire centrale

II.2.1 - Principe général

II.2.2 - Description technique

II.2.3 - Description algorithmique

II.2.4 - Résultats de convergence

II.2.5 - Difficultés numériques

II.3 - Méthode de trajectoire centrale avec poids

II.4 - Commentaires

II.0 - INTRODUCTION

Le succès observé dans les méthodes de point intérieur pour la **(PL)** a permis aux chercheurs du monde d'adapter ces dernières aux problèmes de complémentarité linéaire, en tenant compte des raisons suivantes :

- Les méthodes existantes pour résoudre le **(PCL)** sont des variantes simpliciales qui utilisent généralement la notion de pivots (chapitre **(I)**) où se pose le problème théorique concernant la complexité arithmétique « exponentielle », exprimée par le nombre total de pivots éventuels.
- Considérer le **(PCL)** comme un modèle général pour le développement et l'analyse des algorithmes de point intérieur (puisque le **(PCL)** couvre les problèmes de **(PL)** et de **(PQC)**).

Dans ce chapitre, on présente une classe de méthodes de point intérieur dites de trajectoire centrale. C'est la plus attractive des méthodes de point car d'une part, elles possèdent de bonnes propriétés théoriques, notamment la convergence à la fois polynomiale et superlinéaire et d'autre part, les algorithmes sont essentiellement de type Newton ce qui devrait être en faveur des performances numériques.

Ces résultats sont établis moyennant des hypothèses très gênantes pour l'implémentation numérique. Nous précisons à cet égard à quel niveau les difficultés se présentent, puis nous proposons à la fin un bon remède inspiré de l'approche barrière logarithmique avec poids.



II.1 - Notations et hypothèses.

II.1.1 - Notations.

- Notons par :

$\mathbb{R}_+^n = \{ x \in \mathbb{R}^n \mid x \geq 0 \}$ l'orthant positif de \mathbb{R}^n .

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

$S = \{ (x, y) \in \mathbb{R}_+^{2n} \mid y = Mx + q \}$ l'ensemble des solutions réalisables de **(PCL)**.

$S_{\text{int}} = \text{int}(S) = \{ (x, y) \in \mathbb{R}_{++}^{2n} \mid y = Mx + q \}$ l'ensemble des solutions strictement réalisables de **(PCL)**.

$S_{\text{pc}} = \{ (x, y) \in S \mid x_i y_i = 0 \text{ pour } i = 1, \dots, n \}$ l'ensemble des solutions optimales de **(PCL)**.

- $e = (1, \dots, 1)^t \in \mathbb{R}^n$.

- $X = \text{diag} \{x\} \mid x > 0$, $X e = x$ et $X^{-1} e = (1/x_i)$ pour $i = 1, \dots, n$.

- $Y = \text{diag} \{y\} \mid y > 0$, $Y e = y$ et $Y^{-1} e = (1/y_i)$ pour $i = 1, \dots, n$.

- X, X^{-1}, Y, Y^{-1} : sont des matrices diagonales symétriques et définies positives

II.1.2 - Hypothèses.

Sans perte de généralité on suppose :

(H1). $S_{\text{int}} \neq \emptyset$, i.e., un point strictement réalisable existe (hypothèse standard dans n'importe quel algorithme de point intérieur).

(H2). M est semi - définie positive. Dans ce cas, le **(PCL)** est dit monotone.

Les hypothèses **(H1)** et **(H2)** permettent de simplifier l'étude des algorithmes en question.

II.2 - Présentation typique des méthodes de trajectoire centrale.

Les méthodes de trajectoire centrale sont introduites à l'origine à travers la méthode de fonction barrière logarithmique (86 - 88) puis peu à peu elles prennent une forme assez indépendante (dites convenables).

Dans ce paragraphe, nous donnons une présentation qui prend en compte et l'origine de la méthode et les progrès qu'elle a connus.

II.2.1 - Principe général.

L'idée générale d'une méthode de trajectoire centrale consiste à suivre un chemin particulier (dit des centres), en prenant comme direction de déplacement celle de Newton.

Autrement dit, elle génère une suite $\{(x^k, y^k)\}$ strictement réalisable et une suite $\{\mu^k = (x^k)^t y^k\}$ qui exprime la condition de complémentarité. Cette dernière est vérifiée lorsque la suite $\{\mu^k\}$ tend vers zéro. Auquel cas, on obtient une solution optimale de (PCL).

II.2.2 - Description technique.

Le (PCL) monotone est équivalent au programme quadratique convexe (PQC) suivant :

$$\begin{cases} \min x^t y \\ y = M x + q \\ (x, y) \geq 0 \end{cases}$$

au sens que (x, y) est une solution de (PCL) ssi (x, y) est une solution optimale de (PQC) avec une valeur optimale de l'objectif nulle (Chapitre I).

Donc la résolution de (PCL) est équivalente à celle de (PQC).

La notion de la trajectoire centrale peut être introduite moyennant une fonction barrière logarithmique, il suffit donc d'associer au (PQC) le problème perturbé suivant :

$$(PCL)_\mu \begin{cases} \min \left[x^t y - \mu \sum_{i=1}^n \ln(x_i y_i) \right], \quad \mu > 0 \\ y = Mx + q \\ (x, y) > 0 \end{cases}$$

Théorème d'existence.

Soient M une matrice semi - définie positive et $S_{int} \neq \emptyset$, alors le problème $(PCL)_\mu$ possède une solution optimale unique pour tout $\mu > 0$.

Preuve: Soit $\mu > 0$ fixé.

L'unicité:

La fonction objective :

$$f(x) = x^t (Mx + q) - \mu \left[\sum_{i=1}^n \ln(x_i) + \sum_{i=1}^n \ln(Mx + q)_i \right] \text{ est strictement convexe}$$

puisque'elle se compose de deux fonctions l'une est convexe et l'autre est strictement convexe (on obtient le même résultat si on dérive f deux fois).

Donc, une solution optimale est unique si elle existe .

L'existence:

On démontre que l'ensemble :

$$S_{int}(w) = \{ x \in \mathbb{R}_{++}^n \mid Mx + q > 0 \text{ et } f(x) \leq w \}$$

est : non vide, fermé et borné moyennant les résultats de l'analyse convexe.

Théorème.

Soient $\mu > 0$ et M une matrice semi - définie positive .

(x, y) est une solution optimale de $(\mathbf{PCL})_\mu$ ssi (x, y) satisfait le système suivant :

$$(I) \begin{cases} XYe - \mu e = 0 \\ y = Mx + q \\ (x, y) > 0 \end{cases}$$

où $X = \text{diag}\{x\}$, $Y = \text{diag}\{y\}$ et $e = (1, \dots, 1)^t \in \mathbb{R}^n$.

Preuve:

\Rightarrow) Soit (x, y) une solution optimale de $(\mathbf{PCL})_\mu$, alors il existe un vecteur $v \in \mathbb{R}^n$

(KKT) tel que :

$$\begin{cases} y - \mu X^{-1}e + M^t v = 0 \\ x - \mu Y^{-1}e - v = 0 \\ y = Mx + q \\ (x, y) > 0 \end{cases}$$

$$\Leftrightarrow \begin{cases} XYe - \mu e + XM^t v = 0 \\ XYe - \mu e - Yv = 0 \\ y = Mx + q \\ (x, y) > 0 \end{cases}$$

$$\Leftrightarrow \begin{cases} (XM^t + Y)v = 0 \\ y = Mx + q \\ (x, y) > 0 \end{cases}$$

Puisque M est semi - définie positive et $X > 0$, on peut prouver simplement que $v = 0$. Donc, (x, y) satisfait le système **(I)**.

⇔) Soit maintenant (x, y) un couple satisfaisant le système **(I)**, chaque terme $[x_i y_i - \mu \log(x_i y_i)]$ où $(x_i, y_i) > 0$ atteint son minimum ssi $x_i y_i = \mu$ (cette condition est toujours vérifiée par le système **(I)**). D'où le résultat. □

Remarques.

- 1/ la résolution de $(PCL)_\mu$ est équivalente à résoudre le système **(I)**.
- 2/ L'ensemble de toutes les solutions du système **(I)** est appelé trajectoire centrale, on le note par :

$$S_{cen} = \{ (x(\mu), y(\mu)) / \mu > 0 \}.$$

Résolution du système (I).

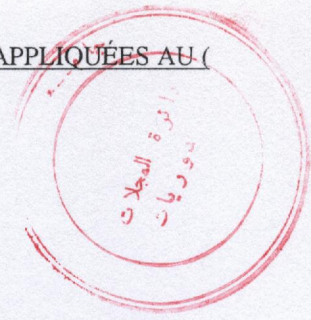
Puisque la première équation du système (I) est non linéaire, une résolution directe est généralement impossible. Alors, on se limite à chercher des solutions approximatives suivant la trajectoire centrale en formant une suite décroissante $\{\mu^k = (x^k)^t y^k\}$ de la condition de complémentarité qui tend vers zéro à la limite. La méthode de Newton est considérée comme l'une des meilleures de résolution. On considère donc la fonction suivante :

$$F_\mu(x, y) = \begin{bmatrix} XYe - \mu e \\ y - Mx - q \end{bmatrix}$$

La solution (x^*, y^*) par l'équation $F_\mu(x, y) = 0$ est une solution optimale du problème perturbé $(PCL)_\mu$.

La direction de Newton est calculée l'équation suivante :

$$F_\mu + \nabla F_\mu \cdot \Delta w = 0, \text{ où } \Delta w = (\Delta x, \Delta y).$$



On obtient le système d'équations linéaires suivant :

$$\begin{cases} Y\Delta x + X\Delta y = XYe - \mu e \\ \Delta y = M\Delta x \end{cases}$$

dont la solution est : $\Delta x = (M + X^{-1} Y)^{-1} (Ye - \mu X^{-1} e)$ et $\Delta y = M \Delta x$

Le nouveau itéré est calculé comme suit : $(x, y) = (x, y) - (\Delta x, \Delta y)$

Facteur de centralité.

Puisqu'on ne peut pas tracer le chemin S_{cen} de manière exacte, il est nécessaire de mesurer la qualité de chaque solution trouvée. Pour cela, on introduit la notion du facteur de centralité : c'est le scalaire défini par :

$$\delta(x, y, \mu) = \min_{\mu \in \mathbb{R}_+} \left\| XYe - \mu e \right\| = \left\| XYe - \left(\frac{x^t y}{n} \right) e \right\|$$

On dit qu'un point est voisin de la trajectoire centrale s'il appartient à l'ensemble suivant :

$$S_{cen}(\alpha) = \left\{ (x, y) \in S_{int} \mid \left\| XYe - \left(\frac{x^t y}{n} \right) e \right\| \leq \left(\frac{x^t y}{n} \right) \alpha : \alpha > 0 \right\}$$

II.2.3 - Description algorithmique.

Les algorithmes de résolution possèdent en général les mêmes idées de base : identifier une bonne solution du système (I) (c'est - à - dire, $x(\mu) > 0$, $y(\mu) > 0$) pour $\mu = \mu_0$, puis diminuer progressivement μ (prenons $\mu_1 < \mu_0$) en résolvant le système (I) au voisinage de $(x(\mu_0), y(\mu_0))$ et ainsi de suite.

Donc, on obtient une suite $\{x(\mu_k), y(\mu_k)\}$ qui tend vers zéro à la limite :

$$\lim_{k \rightarrow \infty} (x^k)^t y^k = 0, \text{ où } x^k = x(\mu_k) \text{ et } y^k = y(\mu_k)$$

Plusieurs auteurs ont développé des algorithmes de trajectoire centrale pour le (PCL) monotone (Kojima, Mizuno, Yoshize, Megiddo,...) dont le schéma général est le suivant:

Algorithme de base.

Début d'algorithme.

Données : Soient $\varepsilon > 0$ un paramètre de précision et $0 < \alpha \leq 0.1$ une constante donnée).

$(x^0, y^0) \in S_{cen}(\alpha)$, c'est - à - dire, (x^0, y^0) est un point strictement réalisable voisin de la trajectoire centrale.

Itération: Calculer :

- Les paramètres : $\delta = \frac{\alpha}{1 - \alpha}$ et $\mu = (1 - \frac{\delta}{\sqrt{n}}) \frac{x^t y}{n}$

- la direction de Newton $(\Delta x, \Delta y)$ solution du système linéaire :

$$\begin{cases} Y\Delta x + X\Delta y = X Y e - \mu e \\ \Delta y = M\Delta x \end{cases}$$

- le nouveau itéré : $(x, y) = (x, y) - (\Delta x, \Delta y)$

Test d'arrêt: L'algorithme s'arrête dès que : $x^t y < \varepsilon$.

Fin d'algorithme.

II.2.4 - Résultats de convergence.

L'algorithme de base converge au bout de $O(\sqrt{n}L)$ itérations, de plus la convergence asymptotique est au moins linéaire si le point initial est voisin de la trajectoire centrale.

Théorème de convergence.

Soient : $0 < \alpha \leq 0.1$ et $\delta = \frac{\alpha}{1-\alpha}$, Supposons que :

$(x, y) \in S_{\text{cen}}(\alpha)$ et $\mu = (1 - \frac{\delta}{\sqrt{n}}) \frac{x^t y}{n}$, alors le point (x^*, y^*) défini par :

$(x^*, y^*) = (x, y) - (\Delta x, \Delta y)$ satisfait :

$$(x^*, y^*) \in S_{\text{cen}}(\alpha) \quad \text{et} \quad (x^*)^t y^* \leq (1 - \frac{\delta}{6\sqrt{n}}) x^t y \quad . \square$$

II.2.5 - Difficultés numériques.

L'algorithme de base possède de bonnes propriétés théoriques telles que la polynomialité et la convergence asymptotique au moins linéaire.

Malheureusement, en pratique on rencontre des problèmes au niveau de l'initialisation puisque si on démarre d'un point initial intérieur quelconque la convergence théorique n'est pas garantie, qu'arrivera t'il alors en pratique ?

En fait, nous avons quand même pris le soin de programmer ce type d'algorithme. Le résultat est que l'on s'arrive toujours à la fin de la phase 1, le point obtenu est toujours loin de la trajectoire centrale et alors la phase 2 ne marche pas.

C'est la condition de centralité qui constitue la difficulté principale. Pour remédier à cette difficulté, nous nous sommes inspirés des travaux de *J.Ding et T.Y.Li* [2] qui proposent un algorithme de même type guidé par une fonction barrière logarithmique avec poids.

II.3 - Méthode de trajectoire centrale avec poids.

On considère le problème perturbé suivant :

$$(\text{PCL})_{\mu r} \begin{cases} \min \left[x^t y - \mu \sum_{i=1}^n r_i \ln(x_i y_i) \right], & \mu > 0 \\ y = Mx + q \\ (x, y) > 0 \end{cases}$$

où $r = (r_1, r_2, \dots, r_n)^t \in \mathbb{R}_{++}^n$ est le poids associé à la fonction barrière logarithmique.

Notons que $(\text{PCL})_{\mu r}$ généralise $(\text{PCL})_{\mu}$ au sens que si $r_i = 1, \forall i$ on retrouve ce dernier. Donc, comme dans le cas de $(\text{PCL})_{\mu}$, le $(\text{PCL})_{\mu r}$ est équivalent au système non linéaire suivant :

$$(\text{II}) \begin{cases} XYe - \mu r = 0 \\ y = Mx + q \\ (x, y) > 0 \end{cases}$$

On applique la méthode de Newton au système (II), on obtient le système linéaire suivant :

$$\begin{cases} Y\Delta x + X\Delta y = XYe - \mu r \\ \Delta y = M\Delta x \end{cases}$$

Le nouveau itéré est défini comme : $(x, y) = (x, y) - (\Delta x, \Delta y)$

où : $\Delta x = (M + X^{-1} Y)^{-1} (Ye - \mu X^{-1} r)$ et $\Delta y = M \Delta x$.

L'algorithme basé sur cette méthode se présente comme suit :

Algorithme .

D'un point initial connu $(x^0, y^0) \in S_{\text{int}}$, on choisit le poids r de telle façon que le point (x^0, y^0, μ^0) sera sur la trajectoire centrale (c'est - dire, il vérifie le système (II)).

Début d'algorithme.

(*Données* : $\varepsilon > 0$ est un paramètre de précision) .

Initialisation : $(x^0, y^0) \in S_{\text{int}}$, $\mu^0 = \frac{\|X^0 Y^0 e\|}{\sqrt{n}}$, $r = \frac{X^0 Y^0 e}{\mu^0}$ et $\alpha = (1 - \frac{\sqrt{2}}{2})\eta$.

Où $X^0 = \text{diag}(x_i^0)$, $Y^0 = \text{diag}(y_i^0)$, $\eta = \min_{i=1}^n r_i$ et $R = \text{diag}(r_i)$ pour $i = 1, \dots, n$

Itération : calculer :

- Les paramètres :

$$\mu = (1 - \frac{\delta}{\sqrt{n}}) \frac{(x^k)^t R y^k}{n}$$

- La direction de Newton $(\Delta x, \Delta y)$ solution du système linéaire :

$$\begin{cases} Y\Delta x + X\Delta y = X Y e - \mu r \\ \Delta y = M\Delta x \end{cases}$$

- Le nouveau itéré :

$$(x, y) = (x, y) - (\Delta x, \Delta y)$$

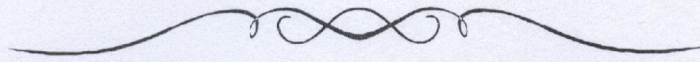
Test d'arrêt : $(x^k)^t R y^k < \varepsilon$

Fin d'algorithme.

II.3 - Commentaires:

Notons en particulier que l'introduction du poids entraîne une grande flexibilité dans le choix du point initial sans détériorer les résultats théoriques.

Sur le plan numérique, c'est merveilleusement amélioré. Ces derniers propos seront conformément dans le chapitre suivant.



III.0 - INTRODUCTION

Notre étude comparative est organisée comme suit :

Au départ, on a implémenté l'algorithme basé sur une fonction barrière logarithmique avec poids en trois étapes :

Etape1: conforme à la théorie (la convergence est garantie mais excessivement lente).

Etape2: Introduction d'une procédure pratique pour calculer le pas de déplacement (le nombre d'itérations et le temps d'exécution sont considérablement réduits).

Etape3: En plus de la procédure présentée à l'étape 2, on prend une certaine liberté au niveau du choix du paramètre barrière μ . les résultats obtenus sont meilleurs que précédemment, par conséquent, c'est cette dernière version qui sera comparée à la méthode de pivotage de **Lemke**.

L'ensemble de ces tests préliminaires montre que l'algorithme de pivotage de Lemke reste encore le favori pour l'utilisateur. Néanmoins, pour des exemples spéciaux le résultat est en faveur de l'algorithme de trajectoire centrale.

III.1 - Implémentation de l'algorithme barrière avec poids.

L'algorithme est programmé en turbo - pascal, il est appliqué à des (PCL) monotones proviennent essentiellement de la (PL) et de la (PQC). Les tests sont effectués sur (IBM 386) avec le test d'arrêt ($\epsilon = 10^{-6}$).

III.1.1 - Cas linéaire.

On considère le (PL) suivant :

$$\begin{cases} \min c^t x \\ Ax \leq b \\ x \geq 0 \end{cases}$$

On lui associe le (PCL) suivant :

$$\begin{cases} \text{trouver } z \in \mathbb{R}^{n+m} \text{ tel que :} \\ z \geq 0, w = Mz + q \geq 0 \text{ et } w^t z = 0 \end{cases}$$

où

$$M = \begin{bmatrix} 0 & A^t \\ -A & 0 \end{bmatrix}, \quad q = \begin{bmatrix} c \\ b \end{bmatrix}, \quad z = \begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix}$$

A est une matrice de type (m, n) et de plein rang . $b \in \mathbb{R}^m$ et $c \in \mathbb{R}^n$.

Dans les exemples ci - dessous :

(w^0, z^0) désigne la solution réalisable initiale de (PCL) et (w^*, z^*) sa solution optimale.

On a seulement les valeurs de z^0 et z^* puisque : $w^0 = M z^0 + q$ et $w^* = M z^* + q$.

Exemple1: Les éléments de la matrice A sont présentés dans le tableau ci - dessous :

2	1
1	2
0	1

$$b = (8, 7, 3)^t \text{ et } c = (-4, -5)^t$$

$$z^0 = (2.212543, 1.559423, 1.959192, 1.652742, 1.611887)^t$$

$$z^* = (3, 2, 1, 2, 0)^t .$$

Exemple2: La matrice A est donnée par :

2	1	0	-1	0	0
0	0	1	0	1	-1
1	1	1	1	1	1

$$b = (0, 0, 1)^t \text{ et } c = (3, -1, 1, 0, 0, 0)^t$$

$$z^0 = (0.013566, 0.141865, 0.110026, 0.268386, 0.110026, 0.233305, 0.474328, 0.762517, 2.175028)^t$$

$$z^* = (0, 0.5, 0, 0.5, 0, 0, 0.5, 0, 0.5)^t$$

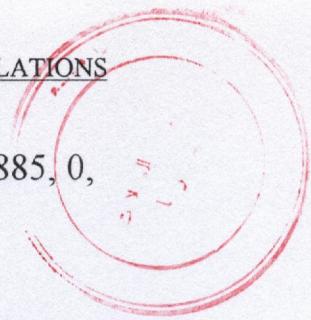
Exemple3: La matrice A est donnée par :

1	2	3	4	5	5	4	3	2	1
6	7	8	9	10	5	2	8	3	1
11	12	13	14	15	6	7	80	90	10
1	10	20	30	40	50	60	80	90	10
3	9	27	60	45	60	75	8	9	46

$$b_i = 10000 \text{ pour } i = 1, \dots, n \text{ et } c_i = -1 \text{ pour } i = 1, \dots, 10 .$$

$$z^0 = (263.115358, 141.578025, 58.537897, 16.249702, 2.046456, 21.550602, 15.528872, 17.532554, 6.575938, 22.583593, 315.184982, 177.790899, 23.186013, 11.846703, 21.972158)^t$$

841.121495, 0, 0, 0, 0, 124.610592, 0, 0, 0, 0, 0, 0, 0.0887885, 0,
0.007788)^t



Notre implémentation de l'algorithme passe par les trois étapes :

Etape1:

Conforme à la théorie, c'est - à - dire :

- On prend la valeur du paramètre μ donnée en théorie : $\mu = (1 - \frac{\delta}{\sqrt{n}}) \frac{x^t R y}{n}$

- Le nouveau point : $(x, y) = (x, y) - (\Delta x, \Delta y)$.

Notons ici que le pas de déplacement est constant ($\alpha = 1$).

Les résultats obtenus pour l'implémentation de cette étape sont donnés dans le tableau suivant :

Algorithme	Taille	Nombre d'itérations	Temps (s)
<i>Exemple 1</i>	5	141	3.02
<i>Exemple 2</i>	9	3628	239.81
<i>Exemple 3</i>	15	27946	3169.37

- Tableau 1 -

Les solutions obtenues sont :

pour l'exemple 1:

$z^* = (2.999999, 2, 1.000001, 1.999999, 0.000001)^t$

pour l'exemple 2:

0, 0.5, 0.000001, 0.499998, 0.000002, 0.000003, 0.5, 0.030738,
0.500002)^t

pour l'exemple 3:

(841.112659, 0.000002, 0, 0, 0, 124.608341, 0, 0, 0, 0, 0, 0, 0.088781,
0.007790)^t

Remarque:

Les résultats obtenus sont précis, cependant le temps et nombre d'itérations augmentent avec la dimension du problème.

On note que la convergence est garantie, mais très lente.

Etape2:

On introduit une procédure pratique équivalente à la recherche linéaire au niveau des résultats, mais beaucoup moins coûteuse pour calculer le pas de déplacement α moyennant les conditions suivantes :

$$x = x - \alpha_x \Delta x > 0 \text{ et } y = y - \alpha_y \Delta y > 0.$$

On prend donc :

- Le pas de déplacement : $\alpha = \min(\alpha_x, \alpha_y)$.
- La valeur de μ comme précédemment.
- Le nouveau point est : $(x, y) = (x, y) - \alpha(\Delta x, \Delta y)$.

L'implémentation de cette étape nous donne les résultats suivants :

Algorithme	Taille	Nombre d'itérations	Temps (s)
<i>Exemple 1</i>	5	66	1.41
<i>Exemple 2</i>	9	1214	82.23
<i>Exemple 3</i>	15	9444	1092.69

- Tableau 2 -

Les solutions obtenues sont :

pour l'exemple 1:

$$z^* = (2.999999, 1.999999, 1.000001, 1.999999, 0.000001)^t$$

pour l'exemple 2:

$$z^* = (0, 0.5, 0.000001, 0.499998, 0, 0.000005, 0.5, 0.436075, 0.500002)^t$$

pour l'exemple 3:

$$841.112669, 0.000002, 0, 0, 0, 124.608341, 0.000001, 0, 0, 0, 0, 0, 0.088781, 0, 0.007790)^t$$

Remarque:

Comme dans le tableau 1, le nombre d'itérations et le temps d'exécution augmentent toujours avec la dimension du problème.

Par rapport aux résultats de l'étape 1, on voit que le nombre d'itérations et le temps d'exécution sont considérablement réduits. Donc cette dernière est plus efficace : elle est pratiquement 3 fois plus rapide que l'étape 1.

Etape3:

En plus de la procédure pratique qui assure la positivité des itérés, on prend une certaine liberté au niveau du choix du paramètre barrière μ (i.e., on ne tient pas compte des valeurs théoriques) puisqu'à l'exception de préserver la polynomialité et la positivité des itérés, il n'y a pas une bonne raison pour se limiter aux valeurs théoriques du paramètre barrière μ .

En effet, des valeurs de μ indépendantes des valeurs théoriques et l'introduction d'une procédure pour calculer le pas de déplacement peuvent diminuer sensiblement le nombre d'itérations et le temps d'exécution.

Dans cette étape, on démarre par exemple d'une valeur initial du paramètre μ donnée par : $\mu^0 = 0.1$ et on réitère par : $\mu^k = \mu^{k-1} / 10$ pour $k = 1, 2, \dots$

Les résultats obtenus sont présentés dans le tableau suivant : Ils sont plus raisonnables que les précédents.

Algorithme	Taille	Nombre d'itérations	Temps (s)
<i>Exemple 1</i>	5	6	0.11
<i>Exemple 2</i>	9	7	0.44
<i>Exemple 3</i>	15	11	2.63

- Tableau 3 -

Les solutions obtenues sont :

pour l'exemple 1:

$$z^* = (2.999999, 2, 1, 2, 0)^t$$

pour l'exemple 2:

0, 0.5, 0.000001, 0.499999, 0.000002, 0.000001, 0.499999, 0.067092,
 0.500001)^t

pour l'exemple 3:

841.112672, 0, 0, 0, 0, 124.608343, 0, 0, 0, 0, 0, 0, 0.088781, 0,
 0.007790)^t

Remarque:

On voit que les résultats obtenus sont meilleurs que précédemment en nombre d'itérations et en temps de calcul.

Donc, cette version s'avère la plus performante, elle est pratiquement 13 à 31 fois plus rapide que la version 2. Par conséquent, c'est cette dernière qui sera comparée à l'algorithme de pivotage de Lemke.

III.1.2 - Cas quadratique.

On considère le (PQC) suivant :

$$\begin{cases} \min c^t x + \frac{1}{2} x^t H x \\ Ax \leq b \\ x \geq 0 \end{cases}$$

A est une matrice de type (m, n) et de plein rang. $b \in \mathbb{R}^m$ et $c \in \mathbb{R}^n$.

H est une matrice carrée d'ordre n, symétrique et semi-définie positive.

Le (PCL) associé à (PQC) est de la forme :

$$\begin{cases} \text{trouver } z \in \mathbb{R}^{n+m} \text{ tel que :} \\ z \geq 0, w = Mz + q \geq 0 \text{ et } w^t z = 0 \end{cases}$$

où

$$M = \begin{bmatrix} H & A^t \\ -A & 0 \end{bmatrix}, \quad q = \begin{bmatrix} c \\ b \end{bmatrix}, \quad z = \begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix}$$

On considère les exemples ci - dessous :

Exemple1: Les matrices A et H sont présentées dans les tableaux ci - dessous respectivement :

1	1
-1	2

2	-2
-2	4

$$b = (2, 2)^t \text{ et } c = (-2, -6)^t$$

$$z^0 = (1.264315, 0.438134, 3.935084, 2.316594)^t$$

$$z^* = (0.8, 1.2, 2.8, 0)^t$$

Exemple2: Les matrices A et H sont présentées dans les tableaux ci - dessous respectivement :

1	2
-1	2

8	6
6	8

$$b = (6, 4)^t \text{ et } c = (-2, 3)^t$$

$$z^0 = (0.713997, 0.034870, 1.363515, 1.083066)^t$$

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

Les résultats obtenus par l'implémentation des trois étapes sont présentés respectivement dans les suivants :

Etape1	Taille	Nombre d'itérations	Temps (s)
<i>Exemple 1</i>	4	355	2.60
<i>Exemple 2</i>	4	1040	8.24

- Tableau 1 -

Les solutions obtenues sont :

pour l'exemple 1:

$$z^* = (0.800001, 1.199999, 2.800001, 0.000002)^t$$

pour l'exemple 2 :

$$z^* = (0.250001, 0, 0.000001, 0.000001)^t$$

Etape2	Taille	Nombre d'itérations	Temps (s)
<i>Exemple 1</i>	4	178	1.48
<i>Exemple 2</i>	4	745	6.40

- Tableau 2 -

Les solutions obtenues sont :

pour l'exemple 1:

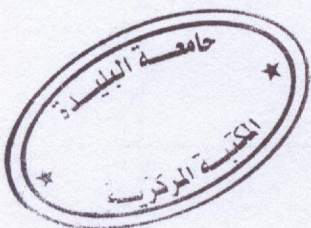
$$z^* = (0.800001, 1.199999, 2.800001, 0.000002)^t$$

pour l'exemple 2 :

$$z^* = (0.250001, 0, 0.000001, 0)^t$$

Etape3	Taille	Nombre d'itérations	Temps (s)
<i>Exemple 1</i>	4	6	0.01
<i>Exemple 2</i>	4	6	0.01

- Tableau 3 -



Les solutions obtenues sont :

pour l'exemple 1 :

$$z^* = (0.8, 1.199999, 2.800004, 0)^t$$

pour l'exemple 2 :

$$z^* = (0.250004, 0, 0.000001, 0)^t$$

Remarque:

Comme dans le cas de la (PL), le nombre d'itérations et le temps sont considérablement réduits pour les exemples de la (PQC).

Donc, la dernière étape est toujours la plus efficace.

III.2 - Comparaison avec l'algorithme de Lemke.

III.2.1 - Cas linéaire. Considérons les exemples suivant :

Exemple1: A est donné par le tableau :

2	1
-2	4

$$b = (2, 3)^t \text{ et } c = (1, 1)^t$$

$$z^0 = (0.335970, 0.673700, 0.152487, 0.049431)^t$$

$$z^* = (0, 0, 0, 0)^t$$

Exemple 2: C'est l'exemple 1 de (III.1.1)

Exemple3: A est donnée par :

-1	2	0
1	-3	-1
1	1	0
-1	-0.5	1
0	-2	0

$$b = (4, -3, 9, -2, -5)^t \text{ et } c = (-1, -1, -1)^t$$

$$^0 = (4.117006, 3.007412, 1.558284, 2.202755, 0.846034, 5.816187, 2.641060, 1.557681)^t$$

$$z^* = (6.5, 2.5, 5.75, 0, 0, 2, 1, 0.25)^t.$$

Exemple4: L'exemple 2 de (III.1.1).

Exemple5: A est donnée par :

3	0	0	0	0
0.8	1	0	0	0
0.32	0.8	1	0	0
0.128	0.32	0.8	1	0
0.0512	0.128	0.32	0.8	1

$$b = (1, 1, 1, 1, 1)^t \text{ et } c = (-0.0256, -0.064, -0.16, -0.4, -1)^t$$

$$^0 = (0.155923, 0.319231, 0.229558, 0.229484, 0.347055, 0.122576, 0.351330, 0.330032, 0.332522, 1.060878)^t$$

$$z^* = (0, 0, 0, 0, 1, 0, 0, 0, 0, 1)^t.$$

Exemple6: La matrice A est présentée dans le tableau suivant :

2	-6	2	7	3	8
-3	-1	4	-3	1	2
8	-3	5	-2	0	2
4	0	8	7	-1	3
5	2	-3	6	-2	-1

$$b = (1, -2, 4, 1, 5)^t \text{ et } c = (-18, 7, -12, -5, 0, -8)^t$$

$$^0 = (1.163220, 2.631928, 0.010540, 0.013106, 4.055956, 0.005665, 0.558612, 1.098286, 1.606507, 0.400404, 1.166049)^t$$

$$z^* = (2, 4, 0, 0, 7, 0, 0.333333, 0, 1.666667, 1, 0)^t$$

Exemple7: La matrice A est présentée dans le tableau suivant :

1	0	-4	3	1	1
5	3	1	0	-1	3
4	5	-3	3	-4	1
0	-1	0	2	1	-5
-2	1	1	1	2	2
2	-3	2	-1	4	5

$$b = (1, 4, 4, 5, 7, 5)^t \quad \text{et} \quad c = (-4, -5, -1, -3, 5, -8)^t$$

$$^0 = (0.068136, 0.629179, 1.423777, 1.424181, 0.710708, 0.064582, 0.016925, 1.253623, 0.948488, 0.479816, 2.001110, 0.724658)^t$$

$$z^* = (0, 0, 2.5, 3.5, 0, 0.5, 0.5, 1.5, 0, 0, 1.5, 0)^t .$$

Exemple8: L'exemple que 3 de (III.1.3).

Exemple9: Le tableau ci - dessous présente les éléments de la matrice A :

1	2	3	4	5	5	4	3	2	1
6	7	8	9	10	5	2	8	3	1
11	12	13	14	15	6	7	80	90	10
1	10	20	30	40	50	60	80	90	10
3	9	27	60	45	60	75	8	9	46
90	100	100	20	30	1000	900	25	1	1
3	30	300	2	20	200	1	10	100	150
5	10	15	20	25	30	35	40	45	50
1	11	111	2	22	222	3	33	333	4
7	70	8	80	9	90	10	100	15	155

$$b_i = 10000 \quad \text{pour} \quad i = 1, \dots, 10 \quad \text{et} \quad c_i = -1 .$$

$$^0 = (1.494503, 11.139990, 7.208822, 31.868903, 73.568456, 0.589676, 2.068013, 17.66657, 12.253799, 11.054999, 327.654094, 44.456750, 21.354428, 11.170232, 18.716298, 0.341455, 5.299552, 19.550481, 3.567429, 12.707693)^t$$

50.147133, 0, 0, 18.009937, 155.909303, 0, 0, 15.979992, 17.503173,
 31.898538, 0, 0, 0, 0.006333, 0.008089, 0.010597, 0.002297, 0, 0.000300,
 0.001348)^t.

Exemple10: La matrice A est donnée par :

518	31	39	59	97	49	22	86	16	23
96	64	14	11	36	12	49	84	55	753
50	28	14	31	59	72	99	40	51	393
76	165	22	20	22	43	61	80	27	393
89	80	78	46	75	31	954	69	33	733
34	49	85	69	49	75	328	40	16	293
30	14	12	46	68	96	62	188	49	823
45	30	74	73	70	61	31	36	35	833
88	74	51	36	60	32	52	20	51	223
73	39	71	74	474	61	28	12	20	133

$b_i = 10000$ et $c_i = -1$ pour $i = 1, \dots, 10$.

$^0 = (7.270952, 22.887774, 24.734995, 4.340420, 4.408340, 31.446521,$
 $0.335163, 3.33514, 78.994220, 0.861111, 1.725278, 0.172366,$
 $0.760991, 0.343705, 0.231104, 1.131966, 0.060395, 0.384417, 2.148054,$
 $1.745868)^t$
 $0, 0, 7.555147, 58.860535, 0, 0, 0, 0, 146.974477, 0, 0, 0, 0, 0, 0,$
 $0.005148, 0.004876, 0.011318, 0)^t$.

Les résultats obtenus par **Lemke** et la version 3 sont présentés dans le tableaux
 ci - dessous :

version3	Lemke	Taille	Nombre d'itérations		Temps (s)	
<i>Exemple 1</i>		4	6	1	0.06	0.01
<i>Exemple 2</i>		5	6	5	0.11	0.06
<i>Exemple 3</i>		8	7	11	0.37	0.11
<i>Exemple 4</i>		9	7	5	0.44	0.11
<i>Exemple 5</i>		10	7	3	0.60	0.05
<i>Exemple 6</i>		11	9	15	1.04	0.39
<i>Exemple 7</i>		12	10	11	1.48	0.28
<i>Exemple 8</i>		15	11	5	2.63	0.22
<i>Exemple 9</i>		20	13	17	7.75	1.21
<i>Exemple 10</i>		20	15	43	9.12	2.96

- Tableau comparatif -

III.2.2 - Cas quadratique. On considère les exemples de (III.1.2), on obtient donc le tableau comparatif suivant :

version3	Lemke	Taille	Nombre d'itérations		Temps (s)	
<i>Exemple 1</i>		4	6	4	0.01	0.01
<i>Exemple 2</i>		4	6	2	0.01	0.01

- Tableau comparatif -

III.2.3 - cas complémentaire.

on considère le (PCL) suivant :

$$\begin{cases} \text{trouver } z \in \mathbb{R}^n \text{ tel que :} \\ z \geq 0, w = Mz + q \geq 0 \text{ et } w^t z = 0 \end{cases}$$

M est une matrice carrée d'ordre n et q un vecteur de \mathbb{R}^n .

Exemple 1: La matrice M est donnée par :

0.5	0.5
-2	1

0.5	0.5
-2	1

$$q = (-1, 2)^t$$

$$z^0 = (1.365943, 1.914737)^t$$

$$z^* = (4/3, 2/3)^t$$

Exemple2: La matrice M est donnée par :

2	1	1
1	2	1
1	1	2

$$q = (-4, -5, -1)^t$$

$$z^0 = (1.884461, 2.384610, 0.406029)^t \text{ et } z^* = (1, 2, 0)^t$$

Exemple3: La matrice M est donnée par :

1	1	3	4
5	3	1	1
2	1	2	2
1	4	1	1

$$q = (-1, 2, 1, -3)^t$$

$$z^0 = (0.028508, 1.227882, 0.942195, 0.793278)^t$$

$$z^* = (0, 0, 0, 3)^t$$

version3	Lemke	Taille	Nombre d'itérations		Temps (s)	
<i>Exemple 1</i>		2	7	3	0.01	0.01
<i>Exemple 2</i>		3	6	3	0.01	0.05
<i>Exemple 3</i>		4	7	2	0.05	0.01

- Tableau comparatif -

Remarque:

L'algorithme de pivotage de Lemke est de 2 à 12 fois plus rapide que la version 3 pour le cas linéaire. Le même résultat est obtenu pour le cas quadratique et complémentaire.

D'une façon générale, c'est la méthode de pivotage de **Lemke** qui est la plus performante. Malheureusement, pour quelques exemples spéciaux, le résultat est en faveur de l'algorithme de trajectoire centrale.

III.3 - Exemples spéciaux.

Dans cette partie, on montre que la méthode de pivotage de **Lemke** malgré son succès, reste à souffrir des problèmes de dégénérescence. De plus, la convergence n'est pas garantie si la matrice M n'appartient pas aux classes des P - matrices.

Problème 1 (dégénéré): on considère l'exemple suivant (donné dans le chapitre I)

La matrice M est représentée dans le tableau ci - dessous :

1	2	0
0	1	2
2	0	1

Le vecteur q est donné par : $q = (-1, -1, -1)^t$

Si on applique la méthode de pivotage, on voit que l'algorithme cycle à partir de l'itération 9 indéfiniment.

Tandis que si on applique la version 3 à partir du point initial

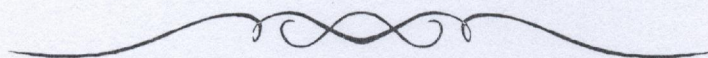
$z^0 = (0.984952, 0.984952, 0.984952)^t$ on obtient la solution :

$z^* = (0.626563, 0.626563, 0.626563)^t$ après 2 itérations.

III.4 - Commentaires.

D'une façon générale, c'est la méthode de pivotage de Lemke qui est la plus performante surtout par rapport au temps.

Ceci n'a rien de surprenant puisque les méthodes de point intérieur visent les grandes dimensions, cependant il ne faut pas oublier que les exemples de taille réduite permettent de saisir les véritables difficultés des algorithmes.



CONCLUSION FINALE

L'étude préliminaire que nous avons effectuée dans cette Thèse constitue un bon stimulant pour développer le comportement numérique des méthodes de trajectoire centrale pour la (PCL) et ce, après une absence quasi - totale de tests numériques due aux difficultés rencontrés dans la phase 1(facteur de centralité).

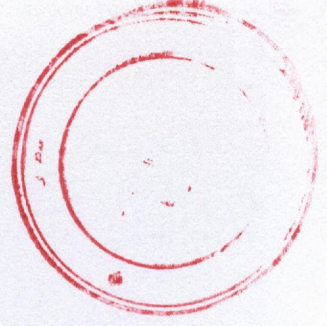
A présent, nous avons réussi l'implémentation d'une version guidée par une fonction barrière logarithmique avec poids. Pour mesurer la qualité des résultats obtenus, nous avons pris le soin de comparer cet algorithme à celui de pivotage de **Lemke**.

On a montré à partir de plusieurs exemples que l'algorithme de pivotage est le plus performant. cependant, pour des exemples spéciaux la méthode de **Lemke** présente certaines difficultés, alors que l'algorithme barrière avec poids fournit les solutions optimales désirées.

Ceci donne une valeur particulière à l'algorithme de trajectoire centrale en particulier et aux méthodes de point intérieur en général.

Bien évidemment, l'algorithme (T.C) peut subir d'autres améliorations, par exemple en tirant profit des subtilités numériques des méthodes Newtoniennes, mais il faudra d'abord le faire théoriquement car ce n'est pas évident. Dans toutes les circonstances, les résultats seront certainement plus appréciables pour les grandes dimensions.





ANNEXE I

AUTRES MÉTHODES DE POINT INTÉRIEUR POUR RÉSOUDRE (PCL)

I.1 - Méthodes de trajectoire centrale non réalisables.

L'obtention d'un point initial strictement réalisable en théorie ou en pratique n'est pas toujours facile.

En théorie, cette difficulté peut être surmonter par l'introduction des variables artificielles.

En pratique, l'idée proposée est de prendre des points initiaux non nécessairement réalisables et on essaye à réaliser la faisabilité et l'optimalité en même temps. D'où, le non des méthodes non réalisables.

I.1.1 - Principe.

A chaque itération, on essaye à réaliser la faisabilité et la complémentarité en même temps en appliquant l'algorithme de Newton au système (I), à partir d'un point initial intérieur non nécessairement réalisable (x^0, y^0) pour le (PCL), c'est-à-dire, $y^0 \neq Mx^0 + q$ et $(x^0, y^0) > 0$.

I.1.2 - Description technique.

Notre objectif est de réaliser la faisabilité et la complémentarité en même temps, donc pour mesurer la complémentarité on utilise la quantité $x^t y$.

Pour la faisabilité, il est naturel d'utiliser la norme $\|y - Mx - q\|$ comme une mesure.

Ceci nous conduit à définir la fonction :

$$\Phi(x, y) = x^t y + \|y - Mx - q\|$$

Donc, l'idée naturelle est de réduire la valeur de cette fonction dans chaque itération (c'est-à-dire, on forme une suite $\{\Phi^k\}$ qui tend progressivement vers zéro).

I.1.3 - Description algorithmique.

L'algorithme de résolution possède les mêmes idées que l'algorithme de base de trajectoire centrale réalisable, à l'exception que les itérés ici ne sont pas forcément réalisables à chaque itération. Il prend donc la forme suivante :

Algorithme.

(**Données:** soient ε un paramètre de termination et α un paramètre choisi).

Initialisation: Soit $(x^0, y^0) > 0$ non nécessairement réalisable .

Itération: Calculer :

- la direction de Newton $(\Delta x, \Delta y)$ solution du système :

$$\begin{cases} Y\Delta x + X\Delta y = -(XYe - \mu e) \\ \Delta y - M\Delta x = -y + Mx + q \end{cases}$$

où (x, y) est un itéré quelconque .

- Le nouveau itéré : $\bar{x} = x + \alpha \Delta x$ et $\bar{y} = y + \alpha \Delta y$

Test d'arrêt: L'algorithme s'arrête dès que : $\Phi \leq \varepsilon$

Fin d'algorithme.

I.1.4 - Propriétés.

1/ L'étude théorique complète de ces algorithmes n'est pas encore établie

(en cours d'étude) puisque plusieurs questions restent ouvertes (complexité et convergence)

2/ On manque des tests numériques.

I.2 - Méthodes affines.

I.2.1 - Principe.

Ces méthodes généralisent les méthodes affines appliquées aux problèmes de la (PL). L'idée principale est d'appliquer l'approche originale de

(PCL)

Dikin dans l'espace primal-dual .

Suivant l'idée de Dikin (1967), on remplace les conditions de non négativité par un ellipsoïde contient le nouveau itéré .

I.2.2 - Description technique.

Soit (x^0, y^0) une paire strictement réalisable de (PCL). On construit un ellipsoïde qui approxime la région réalisable, puis on construit un problème d'optimisation pour déterminer une direction et avec un pas de déplacement convenable, on définit le nouveau itéré.

L'algorithme prend une forme simple à programmer .

Algorithme.

(*Données* : soient : ε un paramètre de précision et θ le pas de déplacement).

Initialisation: (x^0, y^0) : paire initiale de solutions réalisables intérieures .

Itération: Calculer :

- la direction $(\Delta x, \Delta y)$ solution d'un problème d'optimisation.
- le nouveau itéré : $x = x + \theta \cdot \Delta x$ et $y = y + \theta \cdot \Delta y$.

Test d'arrêt: L'algorithme arrête si : $x^t \cdot y < \varepsilon$.

Fin d'algorithme.

I.2.3 - Convergence.

Il n'y a pas de démonstration de complémentarité polynomiale en générale, mais on peut la prouver dans des cas très spéciaux.

I.2.4 - Propriétés.

1/ La méthode est simple : ne nécessite ni transformation projective, ni fonction potentiel, ni préparation du problème .

2/ C'est une méthode primale-duale performante en pratique et plus rapide que toutes les autres versions connues d'après les chercheurs .

(PCL)

I.3 - Méthodes de réduction du potentiel.**I.3.1 - Principe.**

L'idée de base est de ramener la fonction potentiel à $(-\infty)$ c'est à dire conduire la condition de complémentarité $x^t \cdot y$ vers zéro et alors l'obtention d'une solution approximée dans S_{int} .

I.3.2 - Description technique.

Puisque le (PCL) monotone est équivalent à un (PQC), donc il est plus commode d'écrire le (PCL) sous la forme :

$$\begin{cases} \min x^t \cdot y \\ y = Mx + q \\ (x, y) \geq 0 \end{cases}$$

Notre but est de trouver un point réalisable (x, y) tel que : $x^t \cdot y < \varepsilon$, $\varepsilon > 0$.

On introduit donc la fonction : $f(x, y) = p \log(x^t \cdot y) - \sum_{i=1}^n \log(x_i \cdot y_i)$, ($p > n$)

dite fonction potentiel.

Les méthodes de réduction du potentiel sont appliquées au problème de la forme :

$$\begin{cases} \min f(x, y) = p \log(x^t \cdot y) - \sum_{i=1}^n \log(x_i \cdot y_i) \\ y = Mx + q \\ (x, y) > 0 \end{cases}$$

qui peut être résolu par n'importe quelle méthode itérative.

L'algorithme associée à ces méthodes se déroule généralement comme suit :

Chaque itéré est obtenu en faisant une recherche linéaire du pas de déplacement dans la direction où le potentiel diminue .

Algorithme.

(Données : ε est une précision donnée et δ est un montant de diminution) .

Initialisation: (x^0, y^0) est une paire strictement réalisable donnée .

Itération: Calculer :

- La direction $(\Delta x, \Delta y)$.
- Le pas de déplacement $\bar{\theta}$ par une recherche linéaire tels que :
 $(x, y) - \bar{\theta}(\Delta x, \Delta y) \in S_{\text{int}}$ et $f[(x, y) - \bar{\theta}(\Delta x, \Delta y)] \leq f(x, y) - \delta$.
- Le nouveau itéré : $(x, y) = (x, y) - \bar{\theta}(\Delta x, \Delta y)$.

Test d'arrêt: On arrête si : $x^t \cdot y \leq \varepsilon$.

Fin d'algorithme.

I.3.3 - Convergence:

La polynomialité ($O(\sqrt{n} L)$ itérations) et la convergence globale sont établies grâce à l'utilisation de la fonction potentiel .

I.3.4 - Propriétés de la méthode.

1/ C'est une extension d'une méthode primale-duale de la (PL).

2/ L'ingrédient principal est l'utilisation de La fonction potentiel qui assiste dans le choix du pas de déplacement .

3/ La méthode est très efficace en faisant une recherche linéaire du pas de déplacement dans la direction où le potentiel diminue .

4/ $x^t \cdot y$ se réduit dans chaque itération suivant la réduction de la fonction potentiel f .

5/ Elle n'a pas la simplicité de la méthode affine, on rencontre des difficultés en pratique .

RÉFÉRENCES

[1] **Y . Adnan**, " Etudes adaptatives et comparatives de certains algorithmes en optimisation. Implémentations effectives et applications ", Thèse de Doctorat, Mathématiques appliquées de l'université scientifique et médicale de Grenoble I. Juillet (1989).

[2] **S . Bazaraa, H . D . Sherali, C . M . Shetty**, " Non linear programming, theory and algorithms ", Second édition (1993).

[3] **J . F . Bonnans, J . C . Gilbert, C . Lemareshal et C . Sagastizal**, " Méthodes numériques d'optimisation ", (I.N.R.I.A), 78153, Le Chesnay, France Septembre (1995).

[4] **M . Boughazi**, " Contributions à l'étude des algorithmes d'optimisation en analyse des données ", Thèse de Doctorat (Avril 1987), Grenoble.

[5] **J . C . Culioli**, " Introduction à l'optimisation ", Edition Marketing (1994).

[6] **J . Ding , T . Y . Li**, " An algorithm based on weighted logarithmic barrier fonction for linear complementarity problems ", The Arabian Journal for Science and Engineering, Volume 15, Number 4B (1990).

[7] **A . Fischer, C . Kanzow**, " On finite termination of an iterative method for linear complementarity problems ", Mathematical Programming 74 (1996) 279 - 292.

[8] **R . M . Freund and S . Mizuno**, " Intérieur point methods : current status and futur directions ", Mathematical Programming, NO.51, Optima (Octobre 1996).

[9] **C . Gonzaga**, " Path - following methods for linear programming ", Siam Review Vol.34, No.2, P.P. 167 - 224, February 1992.

[10] **O . Guler**, "Generalized linear complementarity problems ", Mathematics of operations research Vol.20, NO.2, May (1995), U.S.A.

[11] **B . Jansen, C . Roos and T . Terlaky**, " A polynomial primal - dual Dikin - type algorithm for linear programming ", Mathematics of operations research Vol.21, NO.2, May (1995), U.S.A.

[12] **A . Keraghel**, " Etude adaptative et comparative des principales variantes dans l'algorithme de Karmarkar ", Thèse de Doctorat (nouvelle Thèse), Mathématiques appliquées de l'université scientifique et médicale de Grenoble I. Juillet (1989).

[13] **M . Kojima, S . Mizuno, and A . Yoshise**, " A polynomial time algorithm for a class of linear complementarity problems ", Mathematical programming 44 (1989), P.1.

[14] **M . Kojima, N . Megiddo, T . Noma and A . Yoshise**, " A unified approach to interior point algorithms for linear complementarity problems : A summary ", Operations Research letters 10 (1991) 247 - 254, North - Holland.

[15] **M . Kojima, S . Mizuno, and A . Yoshise**, " An $O(\sqrt{n} L)$ iteration potentiel reduction algorithm for linear complementarity problems ", Mathematical programming 50 (1991), 331 - 324, North - Holland.

[16] **M . Kojima, N . Megiddo and Y . Ye**, " An interior point potentiel reduction algorithm for linear complementarity problem ", Mathematical programming 54 (1992), 267 - 279, North - Holland.

- [17] **O . L . Mangasarian**, " Simple computable bounds for solutions of linear complementarity problems and linear programs ", *Mathematical programming* 25 (1985), 1- 12, North - Holland.
- [18] **N . Megiddo**, " Pathways to the optimal set in linear programming ", in N . Megiddo, Ed., *Progress in Mathematical programming, Interior - Point and Related Methods* (Springer, New York, 1989).
- [19] **S . Mizuno**, " A new polynomial time method for a linear complementarity problem ", *Mathematical programming* 56 (1992), 31- 43, North - Holland.
- [20] **S . R . Mohan, S . K . Neogy and R . Sridhar**, " The generalized linear complementarity problem revisited ", *Mathematical Programming* 74 (1996), 279 - 292.
- [21] **Monteiro and Adler**, " Interior path - following primal - dual algorithms part I : Linear programming ", *Mathematical programming* 44 (1989), 27 - 41.
- [22] **K . G . Murty**, " Linear complementarity, linear and nonlinear programming ", Berlin : Helderman, (1988).
- [23] **S . Wright**, " An infeasible - interior - point algorithm for linear complementarity problems ", *Mathematical programming* 67 (1994), 29 - 51.
- [24] **S . Wright and Y . Zang**, " A superquadratic infeasible interior - point method for linear complementarity problems ", *Mathematical Programming* 73 (1996) 269 - 289.
- [25] **S . J . Wright**, " Primal - dual interior - point methods ", Copyright (1997) by Siam.

[26] Y . Y e and K . Anstreicher, " On quadratic and $O(\sqrt{n} L)$ convergence of a predictor - corrector algorithm for LCP ", Mathematical Programming 62 (1993), 537 - 551, North - Holland

[27] Y . Zang, " On the convergence of a class of infeasible interior - point methods for the horizontal linear complementarity problems ", Siam Optimization, Vol.4, NO.1, P.P.208 - 227, (February 1994).

[Faint handwritten text, possibly bleed-through from the reverse side of the page]

