

REPUBLIQUE ALGERIENNE DEMOCRATIQUE & POPULAIRE  
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR & DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE MENTOURI – CONSTANTINE  
FACULTE DES SCIENCES EXACTES  
DEPARTEMENT DE MATHEMATIQUES  
ECOLE DOCTORALE DE MATHEMATIQUES- POLE DE CONSTANTINE  
N° d'ordre : 058 / Mag / 2010  
N° de série : 004 / Mat / 2010



## MEMOIRE

Présenté pour l'Obtention du Diplôme de *Magister*

FILIERE : MATHEMATIQUES  
OPTION : MATHEMATIQUES APPLIQUEES

## PAR

HAFSI Narimen

## THEME

*Une recherche linéaire avec des  
fonctions majorantes dans certains  
problèmes d'optimisation*

SOUTENU PUBLIQUEMENT LE : 14 / 02 / 2010.

DEVANT LE JURY COMPOSE DE :

PRESIDENT:	M. Dakhmouche	M.C.	U. M. CONSTANTINE
RAPPORTEUR:	B. Merikhi	M. C.	U. F. A. SETIF
EXAMINATEURS:	Dj. Benterki	M. C.	U. F. A. SETIF
	R. Zitouni	M. C.	U. F. A. SETIF

## *Remerciements*

*Je tiens à exprimer ma profonde reconnaissance à Monsieur B. MERIKHI Maître de conférences à l'université de Sétif, qui a dirigé ce travail, pour ses conseils, ses encouragements, sa disponibilité, qui ont contribués à l'élaboration de ce travail.*

*Je remercie Mr. M. DAKHMOUCHE, Maître de conférences à l'université de Constantine, qui m'a honoré en acceptant de présider ce jury, je lui exprime mon respect et toute ma reconnaissance.*

*Mes vifs remerciements vont également aux docteurs : D.BENTERKI, Maître de conférences à l'université de Sétif et R. ZITOUNI, Maître de conférences à l'université de Sétif, qui ont accepté de juger ce travail et d'être les membres du jury.*

*Je remercie aussi tous les membres de l'équipe d'optimisation, du Laboratoire LMFN de Sétif.*

*Enfin, mes vifs remerciements à tous les enseignants du département de mathématiques de l'université de Sétif.*

# Table des matières

<b>Introduction</b>	<b>2</b>
<b>1 Préliminaires</b>	<b>5</b>
1.1 Analyse convexe . . . . .	5
1.1.1 Ensembles et Applications Affines . . . . .	5
1.1.2 Ensembles convexes . . . . .	6
1.1.3 Fonctions convexes . . . . .	8
1.1.4 Rappel sur les dérivées . . . . .	10
1.2 Programmation linéaire (PL) . . . . .	10
1.2.1 Fondements théoriques . . . . .	10
1.2.2 Dualité en Programmation Linéaire . . . . .	13
1.2.3 Résolution d'un programme linéaire . . . . .	16
1.2.4 Recherche linéaire mono-dimensionnelle . . . . .	18
<b>2 Programmation semi-définie linéaire</b>	<b>28</b>
2.1 Préliminaires matriciels . . . . .	28
2.1.1 Matrices semi-définies positives . . . . .	28
2.2 Formulation du problème . . . . .	31
2.2.1 Dualité en Programmation Semi-Définie . . . . .	33
2.2.2 Exemples de problèmes convertibles en SDP . . . . .	38
2.2.3 Résolution de SDP . . . . .	41

<b>3</b>	<b>Les fonctions majorantes</b>	<b>43</b>
3.1	Introduction . . . . .	43
3.2	Problème ( $D_r$ ) : Aspects théoriques . . . . .	45
3.2.1	Propriétés fondamentales de $f_r$ . . . . .	46
3.2.2	Le problème $D_r$ possède une solution unique . . . . .	46
3.3	Comportement de la solution lorsque $r \rightarrow 0$ . . . . .	47
3.4	Direction de descente de Newton et recherche linéaire . . . . .	49
3.5	Quelques inégalités utiles . . . . .	51
3.6	Calcul des fonctions majorantes . . . . .	52
3.6.1	La première fonction majorante . . . . .	53
3.6.2	La deuxième fonction majorante . . . . .	56
3.6.3	La troisième fonction majorante . . . . .	58
3.7	Description de l'algorithme . . . . .	60
3.8	Tests numériques . . . . .	61
3.8.1	Exemple 1 "Exemple cube" . . . . .	62
3.8.2	Exemple 2 "Exemple artificiel" . . . . .	73
	<b>Conclusion</b>	<b>76</b>

# Introduction

Le sujet traité dans ce mémoire concerne le problème de programmation dit d'optimisation sous contraintes semi-définie, encore appelé problème SDP. Cette appellation est une conséquence de la terminologie anglaise SemiDefinite Programming. Ce problème est une extension naturelle de la programmation linéaire dans le sens où les variables sont des matrices symétriques semi-définies positives au lieu d'être des vecteurs. Ce qui explique d'ailleurs le transport du savoir faire de la programmation linéaire à la programmation semi-définie. L'étude de ce problème a commencé dès le début des années 60 en tant que problème mathématique non linéaire, il existe alors très peu de résultats publiés et ils sont principalement d'ordre théorique.

Depuis les années 90, grâce notamment aux travaux fondateurs de ALIZADEH, NE-MIROVSKI, NESTEROV [94] entre autres, les méthodes de points intérieurs ont pu être étendues à la résolution de problèmes SDP tout en gardant la plupart des bonnes propriétés qui avaient été observées pour les programmes linéaires. En fait, de nombreux résultats sur les programmes linéaires, notamment en termes de dualité et d'optimalité, ont été étendus aux problèmes SDP.

Il a résulté de tout cela un grand nombre de domaines dans lesquels les problèmes SDP ont trouvé des applications. On cite : Optimisation combinatoire, Optimisation non linéaire, Théorie des graphes, commande robuste, etc,...

Actuellement, le problème (SDP) constitue l'un des sujets de recherche le plus convoité dans le domaine de l'optimisation numérique. Le but étant de développer une méthodologie adéquate. A ce propos, il est impératif de traiter des questions ouvertes, déjà étudiées

au niveau de la programmation linéaire : Initialisation, pas de déplacement, coût excessif de l'itération, qui sont ici plus complexes en raison de la structure complexe du problème.

Nous proposons dans ce mémoire une approche, barrière logarithmique dans laquelle, on introduit une procédure originale pour le calcul du pas de déplacement basée sur les fonctions majorantes : On obtient une approximation explicite entraînant une décroissance significative de l'objectif, de plus elle est économique et robuste, contrairement aux méthodes classiques de recherche linéaire.

Présentation du mémoire :

Ce mémoire est réparti en trois chapitres, le premier chapitre présente un rappel des notions fondamentales d'usage fréquent pour la suite, à savoir : L'analyse convexe, et quelques résultats d'existence et d'unicité d'un programme linéaire. Le chapitre 2, est consacré à la programmation semi-définie, on commence par un bref aperçu historique résumant les principaux résultats durant la dernière décennie, suivi par l'exposé de l'approche barrière logarithmique. Enfin, nous terminons le troisième chapitre par l'introduction des fonctions majorantes qui constituent l'originalité de notre étude, ce chapitre est étayé par des expérimentations numériques et des dessins des graphes de fonctions majorantes concluantes. Notre travail ouvre des perspectives pour d'autres classes de problèmes avec autres types de méthodes.

# Chapitre 1

## Préliminaires

### 1.1 Analyse convexe

#### 1.1.1 Ensembles et Applications Affines

**Définition 1** *Un sous-ensemble  $F$  de  $\mathfrak{R}^n$  est dit affine si :*

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

Autrement dit, un sous-ensemble affine contient toujours la “droite” passant par deux de ses points  $x$  et  $y$ .

**Définition 2** *On appelle hyperplan de  $\mathfrak{R}^n$  toute partie affine de dimension  $(n - 1)$*

**Définition 3** *Une application*

$$T : \mathfrak{R}^n \rightarrow \mathfrak{R}$$

$$x \longmapsto Tx$$

*est dite affine si :*

$$T[(1 - \lambda)x + \lambda y] = (1 - \lambda)Tx + \lambda Ty, \forall x, y \in \mathfrak{R}^n, \forall \lambda \in \mathfrak{R}.$$

## 1.1.2 Ensembles convexes

### Les sous-ensembles convexes

**Définition 4** Un sous-ensemble  $C$  de  $\mathfrak{R}^n$  est dit convexe si :

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

Autrement dit, un ensemble convexe contient toujours le “segment” joignant deux de ses points  $x$  et  $y$ .

Sachant que :

$$[x, y] = \{(1 - \lambda)x + \lambda y / 0 \leq \lambda \leq 1\}.$$

**Remarque 1** Tout ensemble affine est convexe. La réciproque est fausse.

### Cônes convexes

**Définition 5** Un sous-ensemble  $K$  de  $\mathfrak{R}^n$  est un cône ssi  $\forall x \in K, \forall \lambda \geq 0, \lambda x \in K$ .

Un cône est donc une union de demi-droites fermées positives issues de l'origine, ce dernier peut (ou ne peut pas) appartenir à  $K$ .

**Définition 6** Un cône est dit pointé ou saillant ssi  $K \cap (-K) = \{0\}$ .

**Propriété 1** 1. Tout ensemble de la forme :  $K = \{x \in \mathfrak{R}^n / b_i^t x \leq 0, i \in I \subseteq \mathbb{N}, b_i \in \mathfrak{R}^n\}$  est un cône convexe.

2. De même l'ensemble des solutions du système d'inégalités homogènes :

$$\{x \in \mathfrak{R}^n / Ax \leq 0, A \text{ est une } m \times n - \text{matrice}\}$$

est un cône convexe.

## Cône de récession

Soit  $C$  un convexe non vide de  $\mathfrak{R}^n$  et  $a \in C$ , on pose

$$C_\infty(a) = \{d \in \mathfrak{R}^n / a + \lambda d \in C, \forall \lambda > 0\}.$$

Alors  $C_\infty(a)$  est un cône convexe non vide. En général,  $C_\infty(a)$  dépend de  $a$ .

**Définition 7** On appelle cône de récession (ou asymptote) de  $C$  l'ensemble

$$C_\infty = \bigcap_{a \in C} C_\infty(a).$$

Un élément  $d \in C_\infty$ , est appelée **direction de récession**.

**Théorème 1** Si  $C \neq \emptyset$ , convexe fermé  $\Rightarrow C_\infty(a) = C_\infty(b) \forall a, b \in C$ ; (c-à-d. :  $C_\infty(a)$  est indépendant de  $a$ . De plus  $C_\infty$  est fermé.

**Proposition 2** Soit  $C$  un convexe fermé non vide de  $\mathfrak{R}^n$ , alors

$$C \text{ est borné ssi } C_\infty = \{0\}.$$

**Remarque 2** Le cône de récession est introduit en 1913 par Steinitz pour  $C$  non vide non nécessairement convexe. C'est le cône fermé défini par

$$C_\infty = \{d \in \mathfrak{R}^n / \exists x_n \in C, t_n \longrightarrow +\infty : \frac{x_n}{t_n} \longrightarrow d\}.$$

La raison principale de son importance est la caractérisation des ensembles compacts de  $\mathfrak{R}^n$  traduite par la proposition précédente pour  $C$  fermé de  $\mathfrak{R}^n$ .

## Ensembles strictement / fortement convexes

**Définition 8** *Un sous-ensemble  $C \subseteq \mathbb{R}^n$  est dit **strictement convexe** s'il vérifie :*

$$\forall x, y \in C, (x \neq y), ]x, y[ = \{(1-t)x + ty \mid t \in ]0, 1[ \} \subset \text{int}(C).$$

**Définition 9** *Une partie  $C \subseteq \mathbb{R}^n$  est dite **fortement convexe** si :*

$$\forall x, y \in C, (x \neq y), \forall t \in ]0, 1[ \exists \gamma > 0 \text{ tel que } : B[(1-t)x + ty, \gamma] \subset C.$$

**Remarque 3** *Nous avons les implications suivantes :  $C$  fortement convexe  $\Rightarrow C$  strictement convexe  $\Rightarrow C$  convexe.*

### 1.1.3 Fonctions convexes

Pour introduire les fonctions convexes, il est plus commode de considérer des fonctions définies sur  $\mathbb{R}^n$  tout entier, susceptibles de prendre des valeurs infinies, i.e : des fonctionnelles avec la convention suivante :

$$\begin{aligned} \infty + \alpha &= \alpha + \infty = +\infty & \alpha - \infty &= -\infty + \alpha = -\infty, \quad \forall \alpha \in \mathbb{R} \\ \infty + \infty &= +\infty & \infty - \infty &= (-\infty) + \infty = +\infty \end{aligned}$$

Cela nous dispense d'avoir à préciser chaque fois le domaine de définition. Si toutefois  $f$  est une fonction définie sur un sous-ensemble  $C \subset \mathbb{R}^n$  et  $C \neq \emptyset$ , on la remplace par l'extension suivante :

$$\tilde{f}(x) = \begin{cases} f(x), & \text{si } x \in C \\ +\infty, & \text{sinon.} \end{cases}.$$

Soit alors

$$f : \mathbb{R}^n \rightarrow [-\infty, +\infty] = \overline{\mathbb{R}} = \mathbb{R} \cup \{\pm\infty\}.$$

**Remarque 4**  $f : C \subset \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$  est convexe sur  $C$  ssi son extension  $\tilde{f}$  (définie précédemment) est convexe sur  $\mathbb{R}^n$ .

**Proposition 3**  $f : \mathbb{R}^n \rightarrow ]-\infty, +\infty]$  est convexe si l'une des deux propriétés équivalentes suivantes est vérifiée :

1.  $f[(1-\lambda)x + \lambda y] \leq (1-\lambda)f(x) + \lambda f(y), \forall \lambda \in [0, 1], \forall x, y \in \mathbb{R}^n.$

2.  $f(\sum_{i=1}^m \lambda_i x_i) \leq \sum_{i=1}^m \lambda_i f(x_i), \forall m \in \mathbb{N}, \forall \lambda_i \geq 0 / \sum_{i=1}^m \lambda_i = 1, \forall x_i \in \mathbb{R}^n.$

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

L'inégalité (2) est appelée **inégalité de Jensen**.

**Lemme 1** Soit  $f : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$  alors,  $f$  est convexe sur  $\mathbb{R}^n$  ssi  $\forall x, d \in \mathbb{R}^n$  la fonction à une variable réelle  $\varphi_{x,d}(t) = f(x + td)$  est convexe sur  $\mathbb{R}$ .

### Semi-continuité (inférieure)

**Définition 10** Une fonction  $f : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$  est dite **semi-continue inférieurement (s.c.i)** en  $x^0 \in \mathbb{R}^n$  si :

$$\forall \varepsilon > 0, \exists \delta > 0 / f(x) \geq f(x^0) - \varepsilon, \text{ dès que } \|x - x^0\| \leq \delta.$$

$f$  est dite **semi-continue supérieurement (s.c.s.)** en  $x^0$  si  $-f$  est (s.c.i) en  $x^0$ .

Evidemment,  $f$  est continue en  $x^0$  ssi elle est à la fois **(s.c.i.)** et **(s.c.s.)** en  $x^0$ . Enfin, on dit que  $f$  est **(s.c.i.)**, si elle est **(s.c.i.)** en tout point  $x \in \mathbb{R}^n$ .

**Remarque 5** 1- On peut remplacer  $\mathbb{R}^n$  dans la définition précédente par un sous-ensemble quelconque  $C \subset \mathbb{R}^n$ .

2- La semi-continuité inférieure de  $f$  en  $x^0$  peut être exprimée par :  $f(x^0) \leq \lim_{k \rightarrow \infty} f(x^k)$ , pour chaque suite  $\{x^k\}$  convergeant vers  $x^0$  et telle que :  $\lim_{k \rightarrow \infty} f(x^k) \in \overline{\mathfrak{R}}$ . Autrement dit :  $f$  est (s.c.i.) en  $x^0$  ssi  $f(x^0) = \lim_{x \rightarrow x^0} \inf f(x^k) \stackrel{\text{déf}}{=} \lim_{\delta \downarrow 0} \left( \inf_{\|x-x^0\| \rightarrow \delta} f(x) \right)$ .

3- On peut également exprimer la (s.c.i.) en termes de voisinages comme c'est le cas pour la continuité. A ce propos, on dit que :

$f$  est (s.c.i.) en  $x^0$  si  $\forall \lambda \in \mathfrak{R}$  vérifiant  $\lambda < f(x^0), \exists V$  (voisinage de  $x$ ) tel que  $\forall x \in V$  on ait  $\lambda < f(x)$ .

### 1.1.4 Rappel sur les dérivées

**Définition 11** Le gradient d'une fonction  $f : \mathfrak{R}^n \rightarrow \mathfrak{R}^n$  continuellement différentiable. évalué au point  $x \in \mathfrak{R}^n$  s'écrit :

$$\nabla f(x) = \left( \frac{\partial f(x)}{\partial x_1} \quad \frac{\partial f(x)}{\partial x_2} \quad \frac{\partial f(x)}{\partial x_3} \quad \dots \quad \frac{\partial f(x)}{\partial x_n} \right)^t.$$

et le coefficient de la  $i^e$  ligne et la  $j^e$  colonne de la matrice Hessienne s'écrit :

$$[\nabla^2 f(x)]_{i j} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}.$$

## 1.2 Programmation linéaire (PL)

### 1.2.1 Fondements théoriques

#### Définition d'un programme linéaire

La programmation linéaire est une technique d'optimisation utilisée aujourd'hui dans beaucoup de domaines et diverses applications, notamment dans l'industrie pétrolière, l'énergie, le transport, les réseaux...

En termes d'optimisation, un programme linéaire (PL) consiste à minimiser (ou maximiser) une forme linéaire ou fonction coût (appelée objectif ou économique) soumise à

des contraintes linéaires (ou plutôt affines).

### Formes usuelles d'un programme linéaire

Un (PL) peut être écrit sous l'une des trois formulations suivantes :

a) **Forme canonique :**

$$\left\{ \begin{array}{l} \min c^t x, \\ Ax \geq b, \\ x \geq 0, \end{array} \right. \quad (a)$$

où  $A$  est une  $m \times n$  - matrice,  $b \in \mathfrak{R}^m$  (constant),  $0 \neq c \in \mathfrak{R}^n$  (vecteur constant) et  $x$  est un vecteur inconnu de  $\mathfrak{R}^n$  (à chercher).

De nombreux modèles d'application se ramènent de manière naturelle à cette forme qui se prête mieux à l'analyse théorique.

b) **Forme standard :**

$$\left\{ \begin{array}{l} \min c^t x \\ Ax = b \\ x \geq 0 \end{array} \right. \quad (b)$$

C'est la forme à laquelle on se ramène le plus souvent à cause de sa convenance pour l'analyse théorique et le traitement numérique.

c) **Forme canonique :**

$$\left\{ \begin{array}{l} \min c^t x \\ Ax \leq b \\ Bx \geq b', B \text{ } p \times n \text{ - matrice, } b' \in \mathfrak{R}^n \end{array} \right. \quad (c)$$

Cette forme qui généralise les formes (a) et (b), est parfois commode pour le traitement des programmes linéaires à variables non astreintes en signe. On parle alors de programmation linéaire généralisée.

**Remarque 6** *Les trois formes précédentes sont équivalentes au sens suivant :*

1. On peut convertir un (PL) d'une forme à l'autre moyennant des transformations algébriques élémentaires :

Une inégalité se transforme en égalité en ajoutant (ou retranchant), selon le sens de l'inégalité, une variable supplémentaire **non négative** appelée **variable d'écart** :

$$\begin{aligned} a_i^t x &\leq b_i \rightarrow a_i^t x + y_i = b_i, y_i \geq 0, \\ a_j^t x &\geq b_j \rightarrow a_j^t x - y_j = b_j, y_j \geq 0. \end{aligned}$$

De même, une égalité est équivalente à deux inégalités :

$$a_k^t x = b_k \iff \begin{cases} a_k^t x \leq b_k \\ \text{et} \\ -a_k^t x \leq -b_k \end{cases} .$$

Une variable  $x$  non astreinte en signe, peut être soit éliminée à partir d'une contrainte d'égalité (s'il en a une), soit remplacée par deux variables astreintes :

$$x = x^+ - x^- \quad \text{et} \quad x^- \geq 0, x^+ \geq 0,$$

où rappelons le :

$$\forall i, x_i^+ = \max(0, x_i), \quad x_i^- = \max(0, -x_i).$$

2. Toute méthode de résolution valable pour une forme, permet également de résoudre les autres, en ce sens que si l'on connaît la solution de l'un on peut en déduire celle de l'autre. Cependant, sur le plan numérique, on préfère souvent la forme standard.
3. Le vecteur  $c$  est appelé **vecteur coût**, et  $A$  est une  $m \times n$ -matrice de plein rang.

**Définition 12** Un ensemble convexe de la forme :  $P = \{x \in \mathbb{R}^n / Ax = b, x \geq 0\}$  est appelé **polyèdre convexe**.

**Définition 13** Un polyèdre convexe borné est appelé **polytope convexe**.

**Définition 14** Soit  $C$  un convexe non vide de  $\mathbb{R}^n$ . Un point  $x \in C$  est dit **extrémal** ( ou **sommet** de  $C$  ) si l'on a :

$$y, z \in C \quad \text{et} \quad x = (1 - \lambda)y + \lambda z \Rightarrow x = y = z, \quad \text{où } 0 < \lambda < 1.$$

**Définition 15** Tout polytope est l'enveloppe convexe de ses points extrémaux.

**Définition 16** Un polytope convexe de  $\mathbb{R}^n$  possédant  $(n+1)$  sommets est appelé :  $n$ -simplexe.

Par exemple dans  $\mathbb{R}^2$  le triangle  $T = \{e_1, e_2, 0\}$  avec  $e_1 = (1, 0), e_2 = (0, 1)$  est un 2-simplexe.

**Définition 17** Ainsi l'ensemble des solutions de  $(PL)$  est le polyèdre convexe

$$P = \{x \in \mathbb{R}^n / Ax = b, x \geq 0\}$$

(éventuellement vide, éventuellement non borné).

A ce propos, il est important de signaler qu'un programme linéaire peut ne pas avoir de solution : par exemple s'il n'y a aucun point réalisable, on dit que le problème est **non réalisable**, ou encore si  $f(x) \rightarrow -\infty$  pour  $x$  réalisable (le problème est non borné), il n'y a cependant aucune difficulté pour détecter ces situations.

C'est pourquoi nous ne considérons que le cas où le problème possède une solution (non nécessairement unique).

## 1.2.2 Dualité en Programmation Linéaire

La notion de dualité est un concept fondamental en programmation linéaire qui conduit à un résultat de portée théorique en pratique : le théorème de dualité.

Aussi, étant donné un programme linéaire (désigné comme primal) on peut toujours lui associer un autre programme linéaire appelé problème dual.

Les deux sont construits à partir des mêmes contraintes, à la différence que si l'un est un problème de minimisation, l'autre est un problème de maximisation.

Deux problèmes duaux ne constituent donc pas deux problèmes différents, mais deux aspects du même problème.

### Définition formelle d'un programme linéaire dual

Soit le programme linéaire écrit sous forme standard :

$$\left\{ \begin{array}{l} \min c^t x \\ Ax = b \\ x \geq 0 \end{array} \right. \quad (P)$$

On appelle dual de  $(P)$ , le programme linéaire :

$$\left\{ \begin{array}{l} \max b^t y \\ A^t y \leq c \\ y \in \mathfrak{R}^m \end{array} \right. \quad (D)$$

Qui est équivalent au problème suivant :

$$\left\{ \begin{array}{l} \max b^t y \\ A^t y + s = c \\ s \geq 0 \\ y \in \mathfrak{R}^m \end{array} \right. \quad (D)$$

On note que les variables du dual sont en bijection avec les contraintes du primal, tandis que les contraintes du dual sont en bijection avec les variables du primal.

**Exemple 1** Soit le programme linéaire primal

$$\left\{ \begin{array}{l} \min (2x_1 - 3x_2) \\ x_1 - x_2 \leq 1 \\ 2x_1 + 3x_2 \geq 4 \\ x_1 + x_2 = 3 \\ x_1 \geq 0, x_2 \in \mathfrak{R}. \end{array} \right. \quad (P)$$

Le dual de ce programme s'écrira :

$$\left\{ \begin{array}{l} \max (y_1 + 4y_2 + 3y_3) \\ y_1 + 2y_2 + y_3 \leq 2 \\ -y_1 + 3y_2 + y_3 = -3 \\ y_1 \leq 0, y_2 \geq 0, y_3 \in \mathfrak{R} \end{array} \right. \quad (D)$$

**Remarque 7 Exemple 2 Remarque 8 1.** Le dual du dual est le primal.

2. Il est utile d'écrire directement le dual d'un programme linéaire mais sous forme standard ( le transformer en forme canonique). Cette définition de dualité s'étend donc à travers ces transformations à tout programme linéaire primal écrit ou non sous forme canonique.

## Résultats fondamentaux

**Théorème 2** Si  $x$  et  $y$  sont réalisables pour (P) et (D) respectivement, alors :  $c^t x \geq b^t y$

**corollaire 1** Si  $x^*$  et  $y^*$  sont respectivement réalisables pour (P) et (D) et si :

$$c^t x^* = b^t y^*$$

alors  $x^*$  et  $y^*$  sont optimales pour leurs problèmes respectifs .

**Remarque 9** *Il est évident que si  $x^*$  et  $y^*$  sont optimales pour  $(P)$  et  $(D)$  respectivement alors :  $c^t x^* = b^t y^*$ .*

### **Théorème de dualité**

- i .** Si l'un ou l'autre des problèmes  $(P)$  et  $(D)$  admet une solution optimale finie, il est de même pour l'autre, et les valeurs correspondantes des objectifs sont égales.
- ii .** Si l'objectif d'un problème n'est pas borné, l'autre problème n'admet pas de solution réalisable.

### **Application de la dualité**

En plus de l'éclairage qu'apporte la dualité sur certains points d'ordre théorique permettant de mieux caractériser les propriétés du problème donné, les résultats précédents montrent qu'en résolvant un programme linéaire  $(P)$ , on résout en même temps son dual.

### **1.2.3 Résolution d'un programme linéaire**

La solution d'un programme linéaire est toujours atteinte en un sommet (point extrémal) de la région admissible, c-à-d : un point réalisable satisfaisant exactement  $(m)$  contraintes linéairement indépendantes.

Le nombre de sommets associés à un problème pouvant être exponentiellement croissant en fonction de la taille du problème ( $n$  est le nombre de variable et  $m$  est celui des contraintes), il est de l'ordre  $C(n, m) = \frac{n!}{m!(n-m)!}$ . Une méthode pratique ne devrait donc pas examiner autant de points pour arriver à l'optimum.

### **La méthode du simplexe**

Cette méthode est une procédure qui examine les points extrémaux (réalisables) d'une façon convenable, le passage d'un sommet à un autre entraîne souvent une diminution de l'objectif si le problème est un problème de minimisation.

La résolution d'un programme linéaire par cette méthode est traditionnellement divisée en deux phases :

- **La phase 1** : Est l'étape d'initialisation, elle consiste à chercher un premier sommet réalisable, et peut être posée (de plusieurs manières) sous forme d'un programme linéaire possédant une solution initiale triviale.
- **La phase 2** : Construit une suite de sommets adjacents réalisables diminuant la valeur de l'objectif d'une itération à l'autre. Le nombre de sommets étant fini, la convergence de la méthode du simplexe vers l'optimum est garantie en un nombre fini d'opérations à condition que le problème ne soit pas dégénéré i.e : aucune composante de la solution réalisable de base n'est nulle.

### Méthodes de points intérieurs

Les méthodes de points intérieurs comme leur nom l'indique éviteront soigneusement la frontière de l'ensemble admissible et ne souffriront donc pas de l'aspect combinatoire inhérent à la méthode du simplexe. Khachiyan (1979) fut le premier à proposer un algorithme dit polynomial, c'est-à-dire dont le nombre d'itérations grandit d'une manière polynomiale avec la taille du problème. Malheureusement, cet algorithme s'est avéré inefficace en pratique, il a fallu donc attendre les travaux de Karmarkar (1984) pour susciter un engouement pour les méthodes de points intérieurs.

Actuellement, l'importance de ces méthodes a dépassé le cadre de l'optimisation linéaire et elles sont beaucoup plus utilisées dans le cadre de l'optimisation convexe, grâce notamment aux travaux de Nesterov et Nemirovsky (1994).

Actuellement, les méthodes de points intérieurs sont devenues compétitives à la méthode du simplexe surtout pour les problèmes de grande taille ( $> 1000$  variables ou contraintes).

Il y a, principalement, quatre grandes catégories des méthodes de points intérieurs :

- Les méthodes affines.
- Les méthodes projectives.

- Les méthodes de réduction du potentielle.
- Les méthodes de trajectoire centrale.

Les méthodes de points intérieurs utilisent les fonctions dites fonctions barrières afin d'obliger les itérés à rester dans l'ensemble réalisable.

**Définition 18 " Fonction barrière "**

Soit  $C \subset \mathbb{R}^n$ . On appelle fonction barrière associée à  $C$  toute fonction  $B$  définie sur  $int(C)$  telle que :

$$\left\{ \begin{array}{l} B \text{ est continue} \\ B(x) \geq 0, \forall x \in int(C) \\ \lim B(x) = +\infty, \text{ quand } x \rightarrow x^* \in Fr(C). \end{array} \right.$$

**Exemple 3** Pour  $C = \{x \in \mathbb{R}^n / g_i(x) \leq 0, i = 1 : m\}$

avec  $g_i \in C^1(\mathbb{R}^n)$

et  $int(C) = \{x \in \mathbb{R}^n / g_i(x) < 0, i = 1 : m\}$ ,

on peut prendre  $B_1(x) = -\sum_{i=1}^m \frac{1}{g_i(x)}$  ou encore  $B_2(x) = -\sum_{i=1}^m \log(-g_i(x))$ ,

le problème pénalisé correspondant est alors :

$$\left\{ \begin{array}{l} \min_{x \in int(C)} (f(x) + \frac{1}{\mu} B(x)), \\ \mu > 0 \end{array} \right. , \tag{P}$$

avec  $\mu > 0$  paramètre de pénalisation.

### 1.2.4 Recherche linéaire mono-dimensionnelle

#### Le problème et ses motivations

La résolution d'un programme mathématique avec ou sans contraintes

$$\left\{ \begin{array}{l} \min f(x) \\ x \in F \subseteq \mathbb{R}^n \end{array} \right. , \tag{D}$$

se fait généralement par des méthodes de descente, dans lesquelles, on passe à chaque itération, par la résolution du sous-problème :

$$\begin{cases} \min \varphi(\alpha) = f(x + \alpha d_k) \\ \alpha \in I = \mathfrak{R}_+ (\text{ou } [a, b], a > 0) \end{cases}, \quad (R-L)$$

pour déterminer le pas de déplacement  $\alpha_k$ .

(R-L) est appelé problème de recherche linéaire, et  $\varphi$  la fonction de recherche linéaire.

### Hypothèses de travail

- $\varphi'(\alpha) < 0$ , condition de descente, équivalente à :

$$\varphi(\alpha_k) = f(x^k + \alpha_k d_k) < f(x^k) = \varphi(0).$$

- L'optimum  $\alpha^*$  de (R-L) existe. Dans le cas contraire, nous avons l'une des deux alternatives suivantes :

1.  $\inf_I \varphi(\alpha) = -\infty \Rightarrow (c)$  non borné :  $\inf_F f(x) = -\infty$ .
2.  $\inf_I \varphi(\alpha)$  fini mais non atteint : on se limite à  $\bar{\alpha}$  tel que  $\varphi(\bar{\alpha}) \cong \inf_I \varphi(\alpha)$ .

Si l'on détermine la valeur exacte  $\alpha^*$ , la R-L est dite exacte (R-L-E), chose impossible en pratique pour (au moins) les raisons suivantes :

1. Opération très coûteuse : nombre d'évaluations excessif.
2.  $\alpha^*$  risque de ne pas exister.

### Attention !

En sacrifiant la précision, on risque d'affecter la convergence des algorithmes de descente.

Il faut donc adopter une R-L qui garantie un degré de précision convenable ou une descente suffisante de  $f(x)$ , susceptible d'induire les algorithmes en question à converger. Une telle R-L est dite **inexacte** (R-L-I).

**Remarque 10** 1. *Les méthodes classiques : Newton, Dichotomie, interpolation sont*

considérées comme des méthodes de R-L-E : on cherche une solution plus ou moins précise (on minimise vraiment  $\varphi(\alpha)$  ).

2. Les méthodes de R-L-I que nous allons envisager sont **économiques** au sens que l'on cherche une solution approchée de R-L avec un nombre d'évaluations minimale...

## Méthodes de R-L économiques

### Orientation

On utilise principalement la propriété de descente :

$$\varphi(\alpha_k) = f(x^k + \alpha_k d_k) < f(x^k) = \varphi(0) \text{ où } \varphi'(0) < 0.$$

### Avantages

- C'est simple et souvent plus efficace.
- On réduit  $f(x)$  d'une itération à l'autre.

### Inconvénients

- La convergence globale de la suite des itérés  $\{x^k\}$  vers un point stationnaire n'est pas garantie à cause du **phénomène d'effort négligeable**. En effet, si  $\bar{\alpha}_k > 0$  est le plus petit  $\alpha > 0$  tel que :  $f(x^k + \alpha_k d_k) = f(x^k)$ , (c-à-d :  $\varphi(\alpha) = \varphi(0)$ ),

**alors** : les décroissances négligeables se présentent si :

$$\alpha_k \downarrow 0 \text{ (trop petit) ou si } \alpha_k \uparrow 0 \text{ (trop grand).}$$

### Remède

Il faut développer des conditions (règles de sélection) pour terminer la R-L avec un pas convenable (c-à-d : loin des extrémités de  $[0, \bar{\alpha}_k]$  ) et qui donne une réduction significative à  $f(x)$  forçant des algorithmes de descente à converger globalement.

## Schéma algorithmique général de R-L

Etant donné  $\alpha > 0$  et une information convenable sur  $\varphi$ , un algorithme de R-L doit posséder trois réponses possibles :

- (O)  $\alpha$  est convenable , on arrête la R-L.
- (G)  $\alpha$  est non convenable (car trop petit).
- (D)  $\alpha$  est non convenable (car trop grand).

Ainsi, n'importe quel  $\alpha$  satisfaisant (G) (resp. (D)) peut servir comme une borne inférieure(resp. supérieure) de l'intervalle d'incertitude de  $\alpha$ , d'où l'algorithme général suivant :

### Algorithme général de R-L

On choisit  $\alpha > 0$  initial (conformément aux informations fournies par (c)).

On pose  $\alpha_g = 0$  et  $\alpha_d = +\infty$  (machine au besoin).

- (1) Si (O) stop on passe à  $x^{k+1}$
- (2) Si (G) on prend  $\alpha_g = \alpha$  , si (D) on prend  $\alpha_d = \alpha$  et aller en (3)
- (3) Choisir un nouveau  $\alpha \in [\alpha_g, \alpha_d]$  et revenir en (1).

L'idée principale des algorithmes de R-L consiste ainsi à réduire l'intervalle d'incertitude  $[\alpha_g, \alpha_d]$  jusqu'à ce que  $|\alpha_d - \alpha_g| \leq \epsilon$  (suffisamment petit).

L'objectif étant bien évidemment de trouver un pas  $\alpha$  ni trop petit ni trop grand vérifiant  $\varphi(\alpha) < \varphi(0)$  , afin de rendre le déplacement de  $x^k$  à  $x^{k+1}$  raisonnable.

### Questions importantes

Avant de définir les règles de sélection, il convient d'avoir des explications rigoureuses concernant les points suivants :

- [Q1] – Pourquoi  $\alpha$  non trop grand et non trop petit ?
- [Q2] – Comment assurer ça (critères) ?
- [Q3] – De tels  $\alpha$  existent-ils ?
- [Q4] – Comment calculer effectivement un  $\alpha$  convenable (algorithmes précis) ?

## Réponses et explications

Pour [Q1] :

On peut donner les explications suivantes :

- Un pas non trop grand est nécessaire pour assurer (forcer) la décroissance  $\varphi(\alpha) < \varphi(0)$ , car avec un pas trop grand, on risque de quitter le voisinage de  $x^k$  et perdre ainsi la propriété de descente qui n'est que locale...
- Un pas non trop grand est nécessaire pour éviter à la suite  $\{x^k\}$  le phénomène d'oscillations :

$$\begin{array}{cccccccc} x^k & - & & - & & - & & - & x^{k+1} \\ x^{k+2} & - & & - & & - & & - & x^{k+3} \end{array} .$$

- De même, le pas ne doit pas être trop petit pour éviter les progrès négligeables de  $x^k$  vers  $x^*$  (solution de (c)) (qui risque d'être loin si  $x^0$  est une initialisation grossière).

Pour [Q2] :

- Il existe un consensus général pour définir le terme **trop grand** : on dit que  $\alpha > 0$  n'est pas trop grand si :

$$\varphi(\alpha) \leq \varphi(0) + m\alpha\varphi'(0). \quad (1.1)$$

où  $m$  est un coefficient choisi dans  $]0, 1[$ , sa valeur précise n'est pas cruciale,  $m = 0.1$  est une valeur raisonnable (en pratique).

- $\alpha > 0$  ne satisfaisant pas (1.1) est considéré trop grand.

**Géométriquement**, cela correspond au cas où  $f(x)$  est non bornée inférieurement.

**Algébriquement** :

Si  $\alpha$  vérifiant (1.1) est trop grand ( $\alpha \rightarrow +\infty$ ) alors :

$$\lim_{\alpha \rightarrow +\infty} \varphi(\alpha) = \varphi(0) + m\varphi'(0), \quad \lim_{\alpha \rightarrow +\infty} \alpha = -\infty \text{ (du fait que } \varphi'(0) < 0 \text{)}.$$

donc  $\varphi$  n'est pas bornée inférieurement et identiquement pour  $f(x)$ .

Conclusion : S'il n'y a pas de  $\alpha > 0$  vérifiant (1.1), c'est que  $f(x)$  n'est pas bornée inférieurement, auquel cas, le problème (c) n'admet pas de solution.

**Remarque 11** La condition (1.1) garantie automatiquement la descente  $\varphi(\alpha) < \varphi(0)$  puisque  $\varphi'(0) < 0$ .

En revanche, un pas non trop grand peut être trop petit, d'ailleurs la propriété (1.1) est vraie pour tout  $\alpha > 0$  suffisamment proche de zéro, puisque dans ce cas on a :

$$\varphi(\alpha) \cong \varphi(0) + \alpha\varphi'(0) \leq \varphi(0) + m\alpha\varphi'(0), (m < 1).$$

Il faut donc empêcher ça en introduisant un critère complémentaire à (1.1). Or, d'après ce qui précède, il suffit de considérer l'inégalité dans l'autre sens avec un coefficient  $m' > m$ . C'est justement le critère de **Goldstein** (1965) :

$$\varphi(\alpha) \geq \varphi(0) + m'\alpha\varphi'(0); \quad \text{avec } 0 < m < m' < 1. \quad (1.2)$$

**Remarque 12** La règle de **Goldstein** s'applique lorsque le gradient de la fonction ne peut être évalué (ou est trop coûteux à obtenir).

On peut vérifier que pour  $\alpha > 0$  suffisamment proche de zéro le critère (1.2) n'est pas vérifié (continuité de  $\varphi$  + la condition  $\varphi'(0) < 0$ ).

En effet, soit  $\alpha \sim 0$  vérifiant (1.2) alors :

$$\lim_{\alpha \rightarrow 0} \frac{\varphi(\alpha) - \varphi(0)}{\alpha} = \varphi'(0) \geq m'\varphi'(0),$$

d'où  $m' \geq 1$  ce qui est absurde.

Si  $f(x)$  est  $C^1$  et si le calcul de  $\varphi'(\alpha)$  ne pose pas de problèmes (numériquement), un autre critère (dû à **Wolfe** (1968) et **Powell** (1976)) peut être envisagé pour exclure les pas trop petits :

$$\varphi'(\alpha) \geq m''\varphi'(0); \quad 0 < m < m'' < 1. \quad (1.3)$$

**Remarque 13** Le critère de **Wolfe** nécessite l'évaluation du gradient à chaque fois que

la fonction est calculée.

Dans la règle de **Wolfe** aussi, c'est la relation (1.1) qui permet d'assurer que  $\alpha$  n'est pas choisi trop grand. Mais comme on suppose que le calcul du gradient de la fonction ne nécessite pas beaucoup plus de calcul que l'évaluation de la fonction elle-même, on utilise la relation (1.3) au lieu de la relation (1.2). Et donc à chaque point  $\alpha$ ,  $\varphi'(\alpha)$  peut être calculée par :

$$\varphi'(\alpha) = d^t \nabla f(x + \alpha d).$$

On a dit que le critère de **Wolfe** est envisagé pour exclure les pas trop petits.

**Remarque 14** En effet,  $\alpha > 0$  proche de zéro ne vérifie pas (1.3) car si oui on aura :

$$\lim_{\alpha \rightarrow 0} \varphi'(\alpha) = \varphi'(0) \geq m'' \varphi'(0) \Rightarrow m'' \geq 1 \text{ (on utilise la continuité de } \varphi').$$

**Remarque 15** "Sur le choix de  $m, m'$  et  $m''$  "

**Définition 19** Il convient de choisir le coefficient de descente  $m < \frac{1}{2}$  et  $m' > \frac{1}{2}$  dans la règle de **Goldstein**.

Le choix de  $m$  et  $m''$  dans la règle de **Wolfe-Powell**, n'est pas très critique. Des valeurs courantes sont :  $m = 0.1$  et  $m'' = 0.7$ .

**Pour [Q3] :**

On démontre (de manières assez différentes) le résultat suivant :

**Lemme 2** Supposons  $f(x)$  bornée inférieurement,  $0 < m < \frac{1}{2}$ ,  $m < m' < 1$  et  $\varphi'(0) < 0$ , **alors** : Il existe un intervalle  $[\alpha_1, \alpha_2]$  ( $0 < \alpha_1 < \alpha_2$ ) tel que pour tout  $\alpha \in [\alpha_1, \alpha_2]$  vérifie les critères de sélection (1.1) et (1.2) ou (1.3).

Le résultat est également valable pour d'autres critères (**Armijo ...**).

**Remarque 16** Si  $m' < m$  Fletcher et El Baali construisent une fonction  $f$  de classe  $C^1$  pour laquelle il n'existe pas de  $\alpha$  convenable.

**Pour [Q4] :**

Il suffit d'écrire correctement les algorithmes de R-L correspondant aux descriptions précédentes, en ajoutant les précisions qu'il faut pour l'aspect numérique.

Pour la convergence, on démontre en général que la suite  $\{x^k\}$  pour laquelle on sélectionne  $\{\alpha_k\}$  par les règles précédentes, converge vers un point d'accumulation  $x^*$  qui vérifie les conditions nécessaires d'optimalité.

La vitesse de convergence devra en principe être meilleure qu'avec une R-L exacte, c'est d'ailleurs la motivation principale de ces méthodes. . .

### **Procédure type de Goldstein (G) , Wolfe (W)**

(1) On dispose de  $\alpha_g = 0$ ,  $\alpha_d =$ une valeur (maximale) quelconque  $m = 0.1$ ,  $m' = 0.7$

$$\varphi'(0) = \nabla f(x^k)^t d_k; \quad \alpha = \alpha_0 \text{ (valeur initiale choisie)}$$

(2)  $\varphi(\alpha) = f(x^k + \alpha d_k)$

**si**  $\varphi(\alpha) \leq \varphi(0) + m\alpha\varphi'(0)$  aller en (3)

**sinon** prendre  $\alpha_d = \alpha$  et aller en (5)

(3) (G) **si**  $\varphi(\alpha) \geq \varphi(0) + m'\alpha\varphi'(0)$  **stop**

**sinon** aller en (4)

(W) calculer  $\varphi'(\alpha) = \nabla f(x^k + \alpha d_k)^t d_k;$

**si**  $\varphi'(\alpha) \geq m'\varphi'(0)$  **stop**

**sinon** aller en (4)

(4)  $\alpha_g = \alpha$

(5) Rechercher un nouveau  $\alpha \in ]\alpha_g, \alpha_d[$  (par dichotomie) et retourner en (2).

**N.B :**

- Il faut toujours commencer par la condition de descente (1.1).

- Le choix de  $\alpha_0$  est très important pour l'efficacité des algorithmes, il faut, à chaque fois qu'il est possible, utiliser les informations fournies par l'algorithme d'optimisation dans lequel est insérée la procédure de R-L. Cela a pour effet de réduire considérablement le nombre d'évaluations de  $\varphi$  (ou  $\varphi'$ ).

### Procédure d'Armijo

On se donne  $\epsilon \in ]0, 1[$  et  $\lambda > 1$ ; valeurs typiques  $\epsilon = 0.2$ ,  $\lambda = 2$

$\alpha$  est acceptable s'il vérifie à la fois :

$$\varphi(\alpha) \leq \varphi(0) + \epsilon\alpha\varphi'(0) \quad (\text{non trop grand}), \quad (1.4)$$

$$\varphi(\lambda\alpha) > \varphi(0) + \lambda\alpha\varphi'(0) \quad (\text{non trop petit}). \quad (1.5)$$

On choisit  $\alpha = \alpha^* > 0$  vérifiant (1.4), si (1.5) est également vérifiée stop **si** sinon on agrandit  $\alpha$  (le doubler) jusqu'à trouver un entier  $t > 0$  tel que  $\alpha = 2^t\alpha^*$  vérifie (1.5).

**Si**  $\alpha^*$  vérifie (1.5), alors il est divisé par 2 jusqu'à trouver le plus petit entier  $t > 0$  tel que  $\alpha^*/2^t$  vérifie (1.4).

### Performances des algorithmes de R-L (Retour à l'exactitude)

Il ne suffit pas de considérer la performance des algorithmes de R-L seulement en terme de nombre d'évaluations par itération (de  $\varphi$  ou  $\varphi'$ ), mais aussi en terme de la réduction de  $f(x)$  obtenue à chaque itération. Ce point important est parfois négligé en théorie (convergence).

Signalons à ce propos, l'existence de procédures de R-L avec lesquelles une seule évaluation par itération suffit pour trouver  $\alpha_k$  convenable, alors que n'importe quelle méthode de descente incorporant ce type de R-L s'avère extrêmement inefficace.

Le meilleur pas que l'on puisse trouver est donc celui qui entraîne une réduction maximale pour  $f(x)$ , ce qui suggère que  $\alpha_k$  doit être une bonne approximation du minimiseur de  $\varphi(\alpha)$  c-à-d voir la R-L comme un problème de minimisation mono-dimensionnelle. De

tels  $\alpha_k$  produisent sans doute une importante décroissance au niveau de  $f(x)$  et jouent un rôle considérable dans les résultats de convergence...

Les critères pratiques (et c'est là toute la différence avec la R-L-E) correspondant à cette idée nécessitent éventuellement que la quantité  $|\varphi'(\alpha_k)|$  soit suffisamment réduite, par exemple :

$$|\varphi'(\alpha_k)| \leq -\eta\varphi'(0), \quad (1.6)$$

où le coefficient  $\eta \in ]0, 1[$ , indique la précision avec laquelle  $\alpha$  approxime le point stationnaire de  $\varphi$ .

La condition (1.6) assure que  $\alpha_k$  n'est pas trop petit et peut donc être associée à la condition de descente (1.1). De plus, elle présente l'avantage que son interprétation en terme de minimum local suggère des méthodes très efficaces pour calculer  $\alpha_k$  ; par exemple, la méthode d'interpolation avec sauvegarde converge très rapidement.

# Chapitre 2

## Programmation semi-définie linéaire

### 2.1 Préliminaires matriciels

#### 2.1.1 Matrices semi-définies positives

**Notation 4** *Soit :*

$M_n(\mathfrak{R}) = M_n$  l'ensemble des matrices carrées de taille  $n$  à coefficients dans  $\mathfrak{R}$ .

$S_n(\mathfrak{R}) = S_n$  l'ensemble des matrices symétriques réelles de taille  $n$  c-à-d : les matrices  $A \in M_n$  vérifiant  $A = A^t$ .

#### Produit scalaire, normes

Le produit scalaire de deux matrices  $A$  et  $B$  de  $M_n$  noté :  $A \cdot B$  ( ou  $\langle A, B \rangle$  ) est défini par :

$$A \cdot B = \langle A, B \rangle = \text{Tr}(A^t B) = \sum_{i=1}^n \sum_{j=1}^n a_{ij} b_{ij} = B \cdot A$$

Où  $\text{Tr}(A) = \text{Trace}(A) = \sum_{i=1}^n a_{ii}$  (la somme des éléments diagonaux de  $A$ ).

On a la propriété suivante :

$$\text{Tr}(AB) = \text{Tr}(BA),$$

à ce produit scalaire, on associe une norme, dite de Frobenius

$$\| A \|_F = \sqrt{\langle A, A \rangle} = \sqrt{\text{Tr}(A^t A)},$$

on utilise également la norme spectrale

$$\| A \|_2 = \sqrt{|\lambda_{\max}(AA^t)|}.$$

**Remarque 17** Si  $A \in S_n$ , on a les résultats suivants :

$$\begin{aligned} \| A \|_F &= \left( \sum_i^n \lambda_i^2(A) \right)^{\frac{1}{2}} = \left( \sum_{i,j} a_{ij}^2 \right)^{\frac{1}{2}}, \\ \| A \|_2 &= \max_i |\lambda_i(A)| = \varphi(A) = \text{le rayon spectral de } A, \end{aligned}$$

où les  $\lambda_i$ ,  $i = 1..n$  désignent les valeurs propres de  $A$  (qui sont toutes réelles).

### Matrices semi-définies positives

**Définition 20** Soit  $A \in S_n$

$\cdot A$  est dite semi-définie positive ou  $A \succeq 0$  si  $\forall x \in \mathfrak{R}^n; x^t A x \geq 0$ .

$\cdot A$  est dite définie positive ou  $A \succ 0$  si  $\forall x \in \mathfrak{R}^n, x \neq 0$  et  $x^t A x > 0$ .

**Notation 5**  $S_n^+(\mathfrak{R}) = S_n^+ = \{A \in S_n / A \succeq 0\}$ .

$S_n^{++}(\mathfrak{R}) = S_n^{++} = \{A \in S_n / A \succ 0\}$ .

**Proposition 6** Soit  $B \in M_n$  inversible, alors :

$$A \in S_n^+ \Leftrightarrow B^t A B \in S_n^+$$

et

$$A \in S_n^{++} \Leftrightarrow B^t A B \in S_n^{++}$$

### Le cône $S_n$

Nous étudions quelques propriétés spécifiques à  $S_n$  et sa structure conique.

**Proposition 7** 1.  $S_n^+$  est un cône fermé pointé dans  $S_n$ .

(car  $S_n^+ \cap (-S_n^+) = \{0_n\}$ ;  $0_n$  est la matrice nulle).

2.  $S_n^{++}$  n'est pas un cône (car  $0_n \notin S_n^{++}$ ).

3.  $S_n^{++}$  est l'intérieur de  $S_n^+$  ( c'est l'intérieur relatif ),

(à rappeler que :  $C \subset \mathfrak{R}^n$ ,  $C \neq \emptyset$ , l'intérieur relatif de  $C$  est :

$$ri(C) = \{aff(C) / \exists \varepsilon > 0; (x + \varepsilon B) \cap aff(C) \subseteq C\},$$

et tel que :

$$aff(C) = \left\{ x = \sum_i^n \lambda_i x_i, x_i \in C, \sum_i^n \lambda_i = 1 \right\},$$

$B$  est la boule unité euclidienne de  $\mathfrak{R}^n$ ;

$$B = \{x \in \mathfrak{R}^n / \|x\| \leq 1\}.$$

et si

$$aff(C) = \mathfrak{R}^n \Rightarrow ri(C) = int(C) \quad / \quad int(C) = \{x \in C \quad / \quad \exists \varepsilon > 0; x + \varepsilon B \subseteq C\},$$

et  $C$  un convexe de  $\mathfrak{R}^n$ .

4. La structure du cône de  $S_n^+$  induit un ordre partiel sur  $S_n$  dit : "Ordre de Löwner" :

$$A \succeq B \Leftrightarrow A - B \in S_n^+(\mathfrak{R}),$$

$$A \succ B \Leftrightarrow A - B \in S_n^{++}(\mathfrak{R}).$$

( $\succ, \succeq$  : relation d'ordre partiel ).

**Proposition 8** Soient  $A, B \in S_n^+$ , alors :

1.  $A + B \succeq B$

$$2. A^{\frac{1}{2}}BA^{\frac{1}{2}} \succeq 0,$$

(à noter que si  $A \in S_n$  alors il existe une unique matrice  $M \in S_n^+$  telle que  $A = M^2$  de plus  $M$  commute avec  $A$  et vérifie  $\text{rg}(A) = \text{rg}(M)$  et on note souvent  $M = A^{\frac{1}{2}}$ ).

$$3. \text{Tr}(AB) \leq \text{Tr}(A)\text{Tr}(B).$$

4. Les valeurs propres de la matrice  $AB$  sont positives.

**Remarque 18** Le produit de deux matrices semi-définies positives n'est pas nécessairement une matrice semi-définie positive.

Effectivement : soient les matrices  $A$  et  $B$  suivantes :

$$A = \begin{pmatrix} 2 & 1 \\ 3 & 4 \end{pmatrix} \quad \text{et} \quad B = \begin{pmatrix} 5 & 4 \\ 7 & 6 \end{pmatrix},$$

on a :  $A \succeq 0$  et  $B \succeq 0$  mais  $AB \not\succeq 0$ .

## 2.2 Formulation du problème

Un programme semi-défini linéaire de forme standard est défini par :

$$\begin{cases} \min C \cdot X = \text{Tr}(C^t X) \\ A_i \cdot X = b_i, i = 1..m \\ X \in S_n^+(\mathfrak{R}) \end{cases} \quad (SDP)$$

où :  $C, A_i \in M_n, b = (b_1, b_2, \dots, b_m)^t \in \mathfrak{R}^m$ .

**Remarque 19** 1. On peut supposer ( sans perte de généralité) que les matrices  $C$  et  $A_i$  sont symétriques.

Dans le cas ; où  $C$  n'est pas symétrique, on la symétrise en remplaçant  $C$  par  $\frac{1}{2}(C + C^t)$  de même pour les  $A_i$  où on prend  $\frac{1}{2}(A_i + A_i^t)$  à la place de  $A_i$ .

2. On peut supposer également (et sans perte de généralité) que les matrices  $A_i, i = 1..m$ , sont linéairement indépendantes pour des besoins théoriques.
3. La programmation semi-définie est un cas particulier de la programmation cônica, et une généralité de la programmation linéaire, ici on manipule des matrices au lieu des vecteurs.

Effectivement on peut ramener un problème de programmation linéaire en un problème de programmation semi-définie. Pour ce fait, soit le programme linéaire suivant :

$$\begin{cases} \min c^t x \\ Ax = b \\ x \geq 0, x \in \mathbb{R}^n \end{cases}, \quad (PL)$$

posons

$$C = \begin{pmatrix} c_1 & 0 \dots & 0 \\ 0 & c_2 \dots & 0 \\ 0 & 0 \dots 0 & c_n \end{pmatrix}, \quad X = \begin{pmatrix} x_1 & 0 \dots & 0 \\ 0 & x_2 \dots & 0 \\ 0 & 0 \dots 0 & x_n \end{pmatrix},$$

$$A_i = \begin{pmatrix} a_{i1} & 0 \dots & 0 \\ 0 & a_{i2} \dots & 0 \\ 0 & 0 \dots & a_{in} \end{pmatrix} \quad \text{et } b_i = b_i,$$

on a donc :

$$\begin{cases} c^t x = \langle C, X \rangle = Tr(C^t X), \\ Ax = b \Leftrightarrow A_i \cdot X = b_i, i = 1..m, \\ x \geq 0 \Leftrightarrow x_i \geq 0, \forall i \Rightarrow X \succeq 0, \end{cases}$$

d'où :

$$(PL) \Leftrightarrow \begin{cases} \min c^t x \\ Ax = b \\ x \geq 0, x \in \mathbb{R}^n \end{cases} \Rightarrow \begin{cases} \min \langle C, X \rangle \\ A_i \cdot X = b_i, i = 1..m, \\ X \succeq 0 \end{cases} \Leftrightarrow (SDP).$$

De plus l'ensemble des contraintes vérifiant  $X \succeq 0$  est infini contrairement à la programmation linéaire où il est fini ( $x \geq 0$ ).

**Définition 21** Une matrice  $X \in S_n$  est dite réalisable (resp. strictement réalisable) pour SDP si :

$$\langle A_i, X \rangle = b_i, i = 1..m \text{ et } X \in S_n^+,$$

(resp.  $\langle A_i, X \rangle = b_i, i = 1..m$  et  $X \in S_n^{++}$ ).

On note  $Y$  ( resp.  $\hat{Y}$  ) l'ensemble des solutions réalisables (resp. l'ensemble des solutions strictement réalisables) pour SDP.

**Définition 22** La valeur optimale (primale) de SDP est définie par :

$$p^* = \inf \{ \langle C, X \rangle ; \langle A_i, X \rangle = b_i, i = 1..m, X \in S_n^+ \},$$

$$X^* \in Y \text{ et } \langle C, X^* \rangle = p^*.$$

## 2.2.1 Dualité en Programmation Semi-Définie

La dualité en SDP est très similaire à la dualité classique en Programmation Linéaire à quelques différences près.

Soit le problème SDP linéaire primal sous la forme standard :

$$(SDP) \quad m_p = \min_x [ \langle C, X \rangle , \langle A_i, X \rangle = b_i, i = 1..m, X \in S_n^+ ].$$

Pour obtenir le problème dual de SDP, on considère la fonction Lagrangienne :

$$q(y) = \min_{x \in S_n^+} \left[ \langle C, X \rangle + \sum_{i=1}^m (b_i - \langle A_i, X \rangle) y_i, y \in \mathfrak{R}^m \right],$$

d'où

$$\max_{y \in \mathfrak{R}^m} q(y) = \max_{y \in \mathfrak{R}^m} \min_{x \in S_n^+} \left[ \left\langle \left( C - \sum_{i=1}^m y_i A_i \right), X \right\rangle + \sum_{i=1}^m b_i y_i \right],$$

donc

$$\max_{y \in \mathfrak{R}^m} q(y) = \begin{cases} \max_{y \in \mathfrak{R}^m} \sum_{i=1}^m b_i y_i \text{ si } C - \sum_{i=1}^m y_i A_i \in S_n^+ \\ -\infty \text{ ailleurs} \end{cases} .$$

D'où par convention le dual du problème SDP est un problème SDP défini par :

$$m_d = \begin{cases} \max b^t y \\ C - \sum_{i=1}^m y_i A_i \in S_n^+ , \\ y \in \mathfrak{R}^m \end{cases} ,$$

$$\Leftrightarrow \begin{cases} \max b^t y \\ C - \sum_{i=1}^m y_i A_i = S \quad (DSDP). \\ y \in \mathfrak{R}^m, S \in S_n^+ \end{cases}$$

**Définition 23** Une solution réalisable de DSDP est le couple  $(y, S) \in \mathfrak{R}^m \times S_n^+$  tel que :

$$C - \sum_{i=1}^m y_i A_i = S.$$

De même le couple  $(y, S)$  est dit strictement réalisable pour DSDP si :

$$C - \sum_{i=1}^m y_i A_i \in S_n^{++}.$$

On note  $F$  ( $\widehat{F}$ ) l'ensemble des solutions réalisables pour DSDP ( resp. l'ensemble des solutions strictement réalisables pour DSDP).

**Définition 24** La valeur optimale de DSDP est définie par

$$d^* = m_d = \sup \left\{ b^t y : C - \sum_{i=1}^m y_i A_i \in S_n^+, y \in \mathfrak{R}^m \right\} ,$$

avec

$$(y^*, S^*) \in F \text{ et } b^* y^* = d^* ,$$

et

$$S^* = C - \sum_{i=1}^m y_i^* A_i.$$

**Proposition 9 "Dualité faible"**

Soit  $X \in Y$  et  $(y, S) \in F$  alors

$$C \cdot X - b^t y = X \cdot S \geq 0.$$

**Remarque 20**  $X \cdot S$  est appelé l'écart ou "**Saut de dualité**".

**Remarque 21** Certains résultats de la Programmation Linéaire ne sont pas valables en SDP.

**Exemple 4**  $E = S_3$  l'espace des matrice symétriques  $3 \times 3$ ,  $m = 2$

$$m_p = \min_x [C \cdot X, X \in S_3^+, \langle A_i, X \rangle = b_i, i = 1..m], \quad (SDP)$$

où :

$$C = \begin{pmatrix} 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, A_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, A_2 = \begin{pmatrix} 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix},$$

$$X = \begin{pmatrix} x_{11} & x_{12} & 0 \\ x_{12} & x_{22} & 0 \\ 0 & 0 & x_{33} \end{pmatrix} \text{ et } b = (0, -1)^t.$$

Ce problème s'écrit aussi sous la forme suivante :

$$m_p = \min_x \left[ x_{12}, X = \begin{pmatrix} 0 & x_{12} & 0 \\ x_{12} & x_{22} & 0 \\ 0 & 0 & 1 + x_{12} \end{pmatrix} \succeq 0 \right]. \quad (SDP)$$

L'ensemble des solutions réalisables correspond à l'ensemble

$$Y = \{(x_{12}, x_{22}) \in \mathfrak{R}^2 : x_{12} = 0, x_{22} \geq 0\}.$$

En outre l'ensemble des solutions optimales est

$$X^* = \left\{ \left( \begin{array}{ccc} 0 & x_{12} & 0 \\ x_{12} & x_{22} & 0 \\ 0 & 0 & 1 + x_{12} \end{array} \right) ; x_{22} \geq 0 \right\}.$$

Il s'ensuit que

$$m_p = 0.$$

Ecrivons le problème dual

$$m_d = \max_{y \in \mathfrak{R}^2} \left[ b^t y : C - \sum_{i=1}^2 y_i A_i \in S_3^+ \right], \quad (DSDP)$$

qui correspond à la forme

$$m_d = \max_{y \in \mathfrak{R}^2} \left[ -y : \begin{pmatrix} -y_1 & \frac{1-y_2}{2} & 0 \\ \frac{1-y_2}{2} & 0 & 0 \\ 0 & 0 & y_2 \end{pmatrix} \in S_3^+ \right], \quad (DSDP)$$

l'ensemble des solutions réalisables correspond à l'ensemble

$$\{(y_1, y_2) \in \mathfrak{R}^2 : y_1 \leq 0, y_2 = 1\}.$$

En outre l'ensemble des solutions optimales est

$$Y^* = \{(y_1, 1); y_1 \leq 0\}.$$

Il s'ensuit que

$$m_d = -1 \neq 0 = m_p.$$

Les problèmes (SDP) et (DSDP) sont tous deux réalisables,  $X^*$  et  $Y^*$  sont tous deux non vides mais

$$-\infty < m_d < m_p < +\infty.$$

Ainsi la réalisabilité des deux problèmes ne suffit pas pour avoir les mêmes valeurs optimales.

Pour conserver le résultat de la dualité forte en SDP ; une condition plus forte est indispensable ; il s'agit de la stricte-réalisabilité comme le montre le théorème suivant :

**Théorème 3** 1. Si le problème (SDP) primal est strictement réalisable, c-à-d :

$$\exists X \in S_n^{++}(\mathfrak{R}) : \langle A_i, X \rangle = b_i, i = 1..m,$$

alors

$$m_d = m_p.$$

2. Si en outre  $m_p$  est fini, alors l'ensemble des solutions optimales du problème dual (DSDP) est compact non vide. Si le problème (DSDP) est strictement réalisable, c-à-d :

$$\exists y \in \mathfrak{R}^m : C - \sum_{i=1}^m y_i A_i \in S_n^{++},$$

alors

$$m_d = m_p.$$

3. Si en outre  $m_d$  est fini, alors l'ensemble des solutions optimales du problème primal (SDP) est compact non vide.

**Remarque 22** Le problème de la recherche d'une solution optimale de (SDP) est équivalent à trouver :  $X \in S_n^+(\mathbb{R}), S \in S_n^+(\mathbb{R}), y \in \mathbb{R}^m$  tel que

$$\left\{ \begin{array}{l} \langle A_i, X \rangle = b_i, \quad i = 1..m \\ \sum_{i=1}^m y_i A_i + S = C \\ X \cdot S = 0 \end{array} \right. ; \quad (PCL)$$

(PCL) est appelé "**Problème de Complémentarité Linéaire Semi-Défini**" qui fait l'objet de plusieurs recherches récentes basées essentiellement sur les approches newtoniennes par résolution numérique.

## 2.2.2 Exemples de problèmes convertibles en SDP

### Exemple 5 "Le problème Max-cut"

Nous disposons d'un graphe  $G(X, U)$  orienté et nous noterons  $n$  le nombre de sommets de ce graphe. De plus, nous le supposons valué, c-à-d qu'à chaque arc  $G$  ( du noeud  $i$  au noeud  $j$  ) est associé un poids  $P_{ij}$ .

Le problème que nous souhaitons résoudre est de trouver une coupe de capacité maximum, c-à-d trouver comment séparer les sommets en deux ensembles  $S$  et  $\bar{S}$ , de manière à ce que la capacité de cette coupe soit maximale.

A rappeler que la capacité d'une coupe étant définie comme la somme des capacités des arcs interconnectant  $S$  et  $\bar{S}$ .

Mathématiquement, le problème peut être formulé de la manière suivante :

Soit  $A$  la matrice  $n \times n$  adjacente du graphe.

On définit la matrice :

$$\begin{cases} L = \text{diag}(Ae) - A, \\ e = (1, 1, \dots, 1)^t \in \mathfrak{R}^n \end{cases}$$

La matrice  $L$  est dite la matrice laplacienne associée au graphe.

On représente la coupe  $S$  par le vecteur  $x$  tel qu'à chacun des sommets  $i$  du graphe, nous associons une valeur  $x_i$  de la manière suivante :

$x_i = 1$  si le sommet  $i$  est dans  $S$

$x_i = -1$  si le sommet  $i$  est dans  $\bar{S}$ .

On aura donc la formulation suivante du problème :

$$\begin{cases} \max \frac{1}{4} x^t L x \\ x = \{-1, 1\}^n \end{cases} \quad (MC)$$

On posant :  $X = \frac{1}{4} x x^t$ , on a l'équivalence suivante :

$$\begin{cases} \max \text{Tr} L X \\ \text{diag} (X) = \frac{1}{4} e \\ \text{rg}(X) = 1 \\ X \succeq 0 \end{cases}$$

En négligeant la condition ou la contrainte du rang ( $\text{rg}(X) = 1$ ), on obtient un problème de la forme SDP suivant :

$$m_d = \max \left[ \langle L, X \rangle : X \in S_n^+(\mathfrak{R}), \langle A_i, X \rangle = \frac{1}{4}, i = 1..n \right],$$

où :

$$A_i [j, k] = \begin{cases} 1, \text{ si } i = j = k = 1..n \\ 0 \text{ ailleurs} \end{cases}.$$

**Exemple 6 "Le problème min-max des valeurs propres"**

Ce problème a été étudié vers les années 1950 en Algèbre linéaire comme suit :

Chercher une valeur optimale

$$\lambda^* = \min_{y \in \mathfrak{R}^n} \lambda_{\max}(C + A(y)), \quad (VP)$$

où  $C \in M_n$ ,

$$\begin{aligned} A_{\text{opérateur}} : \mathfrak{R}^n &\longrightarrow M_n \\ y &\longmapsto A(y) = \sum_{i=1}^n y_i A_i, \quad A_i \in M_n \end{aligned}$$

En fait, trouver la plus grande valeur propre d'une matrice symétrique  $A$  correspond au problème suivant :

$$\max_x [x^t A x : \|x\| = 1],$$

où  $x \in \mathfrak{R}^n - \{0\}$  est un vecteur propre de la matrice  $A$ .

Soit encore en considérant sur les matrices symétriques le produit scalaire :  $\langle A, B \rangle = \text{Tr}(AB)$ ,

$$\max_x [\langle A, x x^t \rangle : \langle I, x x^t \rangle = 1]. \quad (P1)$$

Ou encore,  $X$  étant symétrique :

$$\max_X [\langle A, X \rangle : \langle I, X \rangle = 1, X \succeq 0 \text{ de rang } 1]. \quad (P2)$$

Si on considère le problème semi-défini linéaire suivant :

$$\max_X [\langle A, X \rangle : \langle I, X \rangle = 1, X \succeq 0]. \quad (P2)$$

Ce problème admet des solutions optimales car l'ensemble des solutions réalisables est compact. La fonction objective étant linéaire et l'ensemble des solutions réalisables étant convexe, l'optimum est atteint au moins en un point extrémal, donc en un  $X$  matrice de rang 1.  $X$  est aussi solution optimale du problème (P1). Bien sur, toute solution optimale

de (P1) est solution optimale de (P2).

Le problème (VP) s'écrit sous forme d'un problème SDP comme suit :

$$\left( \lambda^* = \min_{y \in \mathfrak{R}^n} \lambda_{\max}(C + A(y)) \right) \iff \begin{cases} \min \lambda \\ \lambda I \succeq C + A(y) \text{ ,} \\ y \in \mathfrak{R}^n \end{cases}$$

$$\iff \begin{cases} \min \lambda \\ \lambda I - C - A(y) \succeq 0 \text{ .} \\ y \in \mathfrak{R}^n \end{cases}$$

Le dual de ce problème est un problème SDP formulé par :

$$\begin{cases} \max \langle C, X \rangle \\ A^t(X) = 0 \\ X \succeq 0 \\ \text{Tr}(X) = 1 \end{cases} .$$

### 2.2.3 Résolution de SDP

Nous avons déjà dit que l'ensemble  $S_n^+$  est un cône non-polyédrique, et la notion de sommet n'est plus valable pour les problèmes SDP ce qui favorise d'avantage l'extension des méthodes de points intérieurs en programmation linéaire pour SDP.

Plusieurs travaux ont été réalisés depuis les années 90 et le grand nombre d'articles parus dans des revues internationales en témoigne, tout particulièrement les travaux de Shapiro, Fletcher-Craven (1996) qui s'intéressent aux conditions d'optimalité du problème SDP, les travaux de Ramana et Wolkowic (1997) qui ont étudié la dualité forte pour ces problèmes. Du point de vue algorithmique, on rencontre les méthodes de points intérieurs qui sont relativement nouvelles et qui s'apparentent à la méthode projective de Karmarkar pour la programmation linéaire. Ces dernières années, plusieurs chercheurs ont proposé des méthodes pour résoudre les problèmes SDP qui sont généralement des

extensions des méthodes de points intérieurs pour la programmation linéaire. En effet, en 1994 Farid ALIZADEH est le 1<sup>er</sup> chercheur qui a fait une étude profonde pour SDP et a proposé un algorithme primal-dual de points intérieurs du type projectif appelé "Algorithme de réduction du potentiel". Depuis, plusieurs algorithmes sont proposés dans la littérature, on cite par exemple :

- F. Alizadeh (1995) propose une méthode projective primale-duale.
- Vanderbeghe et Boyd (1996) ont proposé un algorithme primal-dual.
- Monteiro (1997) propose une méthode de type trajectoire centrale.
- Todd & All (1998) ont proposé des variantes newtoniennes pour résoudre le problème de complémentarité linéaire.
- J. Ji & All (1999) ont étudié la convergence de la méthode de prédicteur-correcteur.
- M. Halicka et autres (2002) ont étudié la convergence de la méthode de trajectoire centrale.
- Dj. Benterki, J.P. Crouzeix et B. Merikhi (2004) ont proposé une étude théorique et numérique de certains algorithmes projectifs de Alizadeh.

Toutes ces méthodes de différents types à savoir les méthodes de trajectoire centrale réalisable et non-réalisable, primale, duale et primale-duale ont un problème majeur commun celui de l'initialisation. En 2007 Dj. Benterki, J.P. Crouzeix et B. Merikhi ont proposé une méthode de point intérieur réalisable pour résoudre SDP.

L'avantage principal de cette méthode est le calcul d'une solution réalisable initiale pour SDP.

# Chapitre 3

## Les fonctions majorantes

### 3.1 Introduction

On propose dans cette partie une approche barrière pour résoudre SDP. La méthode est basée sur une fonction de pénalisation logarithmique donnant lieu à un algorithme de type Newton dont l'efficacité est mise en valeur à travers des expérimentations numériques encourageantes. La difficulté est au niveau de la recherche linéaire : la présence du déterminant dans la définition de la fonction barrière logarithmique entraîne un coût très élevé dans les procédures classiques de la recherche linéaire : Exactes ou approchées. Dans notre approche, au lieu de la minimiser, le long de la direction de descente en un point courant  $y$ , on propose des fonctions majorantes  $G$  pour lesquelles la solution optimale du pas de déplacement  $t$  est obtenue explicitement.

On s'intéresse au problème suivant :

$$m_D = \inf_y [b^t y : \sum_{i=1}^m y_i A_i - C \in S_n^+, y \in \mathfrak{R}^m], \quad (D)$$

où  $S_n^+$  désigne le cône des matrices  $n \times n$  symétriques semi-définies positives et les matrices  $C, A_i, i = 1, \dots, m$ , sont des matrices symétriques données.

Le problème  $(D)$  est le dual du problème suivant :

$$m_P = \sup_X [\langle C, X \rangle : X \in S_n^+, \langle A_i, X \rangle = b_i, \forall i = 1, \dots, m]. \quad (P)$$

A priori, un des avantages du problème  $(D)$  par rapport au problème  $(P)$  est que l'argument de la fonction que l'on minimise est un vecteur, alors qu'il est une matrice dans  $(P)$ . D'autre part sous des hypothèses convenables, la résolution de  $(D)$  est équivalente à celle de  $(P)$  au sens que la solution optimale de l'un des deux problèmes se déduit directement de l'autre via le théorème des écarts complémentaires.

Dans toute la suite, on représentera les ensembles des solutions réalisables par :

$$Y = \{y \in \mathfrak{R}^m : \sum_{i=1}^m y_i A_i - C \in S_n^+\}, \quad \hat{Y} = \{y \in \mathfrak{R}^m : \sum_{i=1}^m y_i A_i - C \in S_n^{++}\},$$

$$F = \{X \in S_n^+ : \langle A_i, X \rangle = b_i, \forall i = 1, \dots, m\}, \quad \hat{F} = \{X \in F : X \in S_n^{++}\}.$$

Tous ces ensembles sont convexes.

Le problème  $(D)$  est approximé par le problème  $(D_r)$

$$(D_r) \quad m(r) = \inf [f_r(y) : y \in \mathfrak{R}^m],$$

avec  $r > 0$  paramètre de pénalisation, et  $f_r : \mathfrak{R}^m \rightarrow ]-\infty, +\infty]$  la fonction barrière définie par :

$$f_r(y) = \begin{cases} b^t y + nr \ln r - r \ln[\det(\sum_{i=1}^m y_i A_i - C)] & \text{si } y \in \hat{Y}, \\ +\infty & \text{sinon.} \end{cases}$$

Pour résoudre  $(D_r)$ , on utilise une méthode de descente classique de type Newton.

On va minimiser une fonction  $G$  telle que :

$$\frac{1}{r} [f_r(y + td) - f_r(y)] = \theta(t) \leq G(t) \quad \forall t > 0, \quad \theta(0) = G(0), \quad \theta'(0) = G'(0) < 0,$$

et la bonne qualité des approximations de  $\theta$  est assurée par la condition  $\theta''(0) = G''(0)$ .

### 3.2 Problème $(D_r)$ : Aspects théoriques

Rappelons que  $(D_r)$ ,  $r > 0$  est le problème

$$(D_r) \quad m(r) = \inf [ f_r(y) : y \in \mathfrak{R}^m ],$$

où  $f_r : \mathfrak{R}^m \rightarrow ]-\infty, +\infty]$  est définie par :

$$f_r(y) = \begin{cases} b^t y + nr \ln r - r \ln [\det(\sum_{i=1}^m y_i A_i - C)] & \text{si } y \in \widehat{Y}, \\ +\infty & \text{sinon.} \end{cases}$$

Les hypothèses suivantes sont nécessaires :

- H1) Le système d'équations  $\langle A_i, X \rangle = b_i$ ,  $i = 1, \dots, m$  est de rang  $m$ .
- H2) Les ensembles  $\widehat{Y}$  et  $\widehat{F}$  sont non vides.

On sait alors que :

- a)  $-\infty < m_P = m_D < +\infty$ ;
- b) les ensembles de solutions optimales de  $(P)$  et  $(D)$  sont convexes compacts non vides ;
- c) si  $\bar{X}$  est solution optimale de  $(P)$ , alors  $\bar{y}$  est solution optimale de  $(D)$  si et seulement si

$$\bar{y} \in Y \text{ et } (\sum_{i=1}^m \bar{y}_i A_i - C)\bar{X} = 0 ;$$

- d) si  $\bar{y}$  est solution optimale de  $(D)$ , alors  $\bar{X}$  est solution optimale de  $(P)$  si et seulement si

$$\bar{X} \in F \text{ et } (\sum_{i=1}^m \bar{y}_i A_i - C)\bar{X} = 0.$$

Dans ces conditions, la résolution du problème  $(D)$  permet d'obtenir celle de  $(P)$  et vice versa.

### 3.2.1 Propriétés fondamentales de $f_r$

Pour  $y \in \widehat{Y}$ , on introduit la matrice  $B(y)$  symétrique définie positive de taille  $m$ , et la matrice  $L(y)$  triangulaire inférieure telles que :

$$B(y) = \sum_{i=1}^m y_i A_i - C = L(y)L^t(y).$$

Et pour,  $i, j = 1, 2, \dots, m$ , on définit :

$$\widehat{A}_i(y) = [L(y)]^{-1} A_i [L^t(y)]^{-1},$$

$$b_i(y) = Tr(\widehat{A}_i(y)) = Tr(A_i B^{-1}(y)),$$

$$\Delta_{ij}(y) = Tr(B^{-1}(y) A_i B^{-1}(y) A_j) = Tr(\widehat{A}_i(y) \widehat{A}_j(y)).$$

Ainsi,  $b(y)$  est un vecteur de  $\Re^m$  et  $\Delta(y)$  une matrice symétrique de taille  $m$ .

**Théorème 4** [8] *La fonction  $f_r$  est deux fois continûment différentiable sur  $\widehat{Y}$ . Plus précisément, pour tout  $y \in \widehat{Y}$  on a :*

- a)  $\nabla f_r(y) = b - rb(r)$ ;
- b)  $\nabla^2 f_r(y) = r\Delta(y)$ ;
- c) *La matrice  $\nabla^2 f_r(y)$  est définie positive.*

### 3.2.2 Le problème $D_r$ possède une solution unique

Puisque la fonction  $f_r$  prend la valeur  $+\infty$  sur la frontière du domaine réalisable et différentiable à l'intérieur, alors elle est semi-continue inférieurement.

Pour prouver que  $D_r$  possède une solution, il suffit de prouver que le cône de récession est réduit à l'origine, avant cela on montre le résultat suivant :

**Proposition 10** [8]  *$d = 0$  quand  $[b^t d \leq 0$  et  $\sum_{i=1}^m d_i A_i \in K$ ].*

**Théorème 5** [8]  *$d = 0$  si  $(f_r)_\infty \leq 0$ .*

### 3.3 Comportement de la solution lorsque $r \rightarrow 0$

Dans ce qui suit, on s'intéresse au comportement de la valeur optimale  $m(r)$  et la solution optimale  $y(r)$  du problème  $(D_r)$ . Pour cela, on introduit la fonction

$$h : \mathfrak{R}^m \times \mathfrak{R} \rightarrow ]-\infty, +\infty],$$

définie par :

$$h(y, t) = \begin{cases} b^t y - \ln \det \left[ \sum_{i=1}^m y_i A_i - tC \right] & \text{si } \sum_{i=1}^m y_i A_i - tC \in \text{int}(S_n^+), \\ +\infty & \text{sinon.} \end{cases}$$

Il est facile de vérifier que la fonction  $h$  est convexe et semi-continue inférieurement sur  $\mathfrak{R}^m \times \mathfrak{R}$ . On considère ensuite la fonction  $\phi : \mathfrak{R}^m \times \mathfrak{R} \times \mathfrak{R} \rightarrow ]-\infty, +\infty]$  définie par :

$$\phi(y, t, r) = \begin{cases} rh(r^{-1}y, r^{-1}t) & \text{si } r > 0, \\ h_\infty(y, t) & \text{si } r = 0, \\ +\infty & \text{si } r < 0. \end{cases} \quad (1)$$

Alors,  $\phi$  est également convexe et semi-continue inférieurement sur  $\mathfrak{R}^m \times \mathfrak{R} \times \mathfrak{R}$ , voir à ce propos Rockafellar([23], page 35).

On définit ensuite  $f : \mathfrak{R}^m \times \mathfrak{R} \rightarrow ]-\infty, +\infty]$  par :

$$f(y, r) = \phi(y, 1, r).$$

$f$  est alors convexe et semi-continue inférieurement. Par construction

$$f(y, r) = \begin{cases} f_r(y) & \text{si } r > 0, \\ b^t y & \text{si } r = 0, y \in Y, \\ +\infty & \text{sinon.} \end{cases} \quad (2)$$

On définit la fonction  $m : \mathfrak{R} \rightarrow ]-\infty, +\infty]$  par :

$$m(r) = \inf_y [f(y, r) : y \in \mathfrak{R}^m].$$

Cette fonction est convexe. En outre on a  $m(0) = m_D$  et  $m(r)$  est la valeur optimale de  $(D_r)$  pour  $r > 0$ .

Il est clair que pour  $r > 0$  on a :

$$m(r) = f_r(y(r)) = f(y(r), r)$$

et

$$0 = \nabla f_r(y(r)) = \nabla_y f(y(r), r) = b - rb(y_r).$$

On s'intéresse maintenant à la différentiabilité des fonctions  $m(r)$  et  $y(r)$  sur  $(0, +\infty)$ .

**Théorème 6** [8] *Les fonctions  $m$  et  $y$  sont continûment différentiables sur  $]0, +\infty[$ . On a pour tout  $r > 0$ ,*

$$r\Delta(y_r)y'(r) - b(y_r) = 0,$$

$$m'(r) = n + n \ln(r) - \ln \det(B(y_r)).$$

En outre,

$$m_D = m(0) \leq b^t y(r) \leq m_D + nr. \quad (3)$$

Désignons par  $S_D$  l'ensemble des solutions optimales de  $(D)$ , on sait que cet ensemble est un convexe compact non vide. La distance du point  $y$  à  $S_D$  est définie par :

$$d(y, S_D) = \inf_z [ \|y - z\| : z \in S_D ].$$

Le résultat suivant concerne le comportement de  $y_r$  et  $m(r)$  lorsque  $r \rightarrow 0$ .

**Théorème 7** [8] *Lorsque  $r \rightarrow 0$ ,  $d(y, S_D) \rightarrow 0$  et  $m(r) \rightarrow m_D$ .*

### 3.4 Direction de descente de Newton et recherche linéaire

De par la présence de la fonction barrière, le problème  $(D_r)$  peut être considéré comme sans contraintes. On peut donc le résoudre par une méthode de descente classique. Comme  $f_r$  prend la valeur  $+\infty$  sur la frontière de  $Y$ , alors les itérés  $y$  sont dans  $\widehat{Y}$ . Ainsi la méthode que nous proposons est une méthode de points intérieurs.

Soit  $y \in \widehat{Y}$  l'itéré en cours. Comme direction de descente en  $y$ , prenons la direction de Newton  $d$  solution du système linéaire :

$$[\nabla^2 f_r(y)]d = -\nabla f_r(y). \quad (4)$$

En vertu du théorème 4, le système linéaire est équivalent au système

$$\Delta(y)d = b(y) - \frac{1}{r}b, \quad (3.1)$$

$b(y)$  et  $\Delta(y)$  sont définies en 3.2.1.

La matrice  $\Delta(y)$  étant symétrique, définie positive, le système linéaire (3.1) peut être efficacement résolu via la décomposition de Cholesky. Bien évidemment, on admet que  $\nabla f(y) \neq 0$  (sinon l'optimum est atteint). Il s'ensuit que  $d \neq 0$ . La direction  $d$  étant calculée, on cherche  $\bar{t} > 0$  donnant une décroissance significative à  $f_r$  sur la demi-droite  $y + td$ ,  $t > 0$ , tout en conservant la définie positivité de la matrice  $B(y + \bar{t}d)$ .

Pour se faire, on considère la fonction

$$\begin{aligned} \theta(t) &= \frac{1}{r}[f_r(y + td) - f_r(y)], \quad y + td \in \widehat{Y}, \\ \theta(t) &= \frac{1}{r} b^t d - \ln \det(B(y + td)) + \ln \det(B(y)). \end{aligned}$$

Puisque  $\nabla^2 f_r(y)d = -\nabla f_r(y, )$  on a :

$$d^t \nabla^2 f_r(y)d = -d^t \nabla f_r(y) = d^t b(y) - r d^t b.$$

Pour simplifier les notations, on prend :

$$B = B(y) = \sum_{i=1}^m y_i A_i - C \quad \text{et} \quad H = \sum_{i=1}^m d_i A_i.$$

$B$  étant symétrique définie positive, il existe une matrice triangulaire inférieure  $L$  telle que  $B = LL^t$ .

Par la suite, on pose

$$E = L^{-1}H(L^{-1})^t.$$

Puisque  $d \neq 0$ , l'hypothèse  $(H_1)$  implique que  $H \neq 0$  et par suite  $E \neq 0$ .

En prenant compte de cette notation, pour tout  $t > 0$ , tel que  $I + tE$  est définie positive, on a :

$$\theta(t) = t[Tr(E) - Tr(E^2)] - \ln \det(I + tE). \quad (3.2)$$

Désignons par  $\lambda_i$  les valeurs propres de la matrice symétrique  $E$ , alors :

$$\theta(t) = \sum_{i=1}^n [t(\lambda_i - \lambda_i^2) - \ln(1 + t\lambda_i)], \quad t \in [0, \widehat{t}[. \quad (3.3)$$

avec

$$\widehat{t} = \sup[t : 1 + t\lambda_i > 0 \text{ pour tout } i] = \sup[t : y + td \in \widehat{Y}]. \quad (3.4)$$

Observons que  $\widehat{t} = +\infty$  si  $E$  est semi-définie positive, et  $0 < \widehat{t} < \infty$  sinon. Il est clair que  $\theta$  est convexe sur  $[0, \widehat{t}[$ ,  $\theta(0) = 0$  et

$$0 < \sum_i \lambda_i^2 = \theta''(0) = -\theta'(0).$$

En outre,  $\theta(t) \rightarrow +\infty$  lorsque  $t \rightarrow \widehat{t}$ . Il s'ensuit, qu'il existe un point unique  $t_{opt}$  tel que

$\theta'(t_{opt}) = 0$ ,  $\theta$  atteint son minimum en ce point.

Malheureusement, il n'existe pas de formule explicite donnant  $t_{opt}$ , et la résolution de l'équation  $\theta'(t_{opt}) = 0$  par des méthodes itératives nécessite à chaque itération le calcul de  $\theta$  et  $\theta'$ . Ces calculs sont trop coûteux du fait que l'expression de  $\theta$  en (2) contient le déterminant qui n'est pas facile à calculer et (3) nécessite la connaissance des valeurs propres de  $E$ , c'est un problème numérique de taille. Ces difficultés nous ont conduit à chercher d'autres alternatives. À partir de la donnée de la matrice  $E$ , il est facile d'obtenir les quantités suivantes

$$\text{trace}(E) = \sum_i e_{ii} = \sum_i \lambda_i \quad \text{et} \quad \text{trace}(E^2) = \sum_{i,j} e_{ij}^2 = \sum_i \lambda_i^2.$$

Dans la section 3.2, on profitera de ces données pour proposer des bornes inférieures pour  $\hat{t}$  et des majorantes pour la fonction  $\theta$ . Avant cela, on introduit quelques inégalités utiles concernant un ensemble de nombres réels positifs dont on connaît la somme et la somme des carrés.

### 3.5 Quelques inégalités utiles

On suppose connaître la moyenne  $\bar{x}$  et l'écart type  $\sigma_x$  d'une série statistique  $\{x_1, x_2, \dots, x_n\}$  de  $n$  nombres réels et, on cherche certaines informations sur les  $x_i$  à partir de ces données.

On rappelle que :

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad \text{et} \quad \sigma_x^2 = \frac{1}{n} \sum_{i=1}^n x_i^2 - \bar{x}^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2.$$

Le résultat suivant qui concerne le maximum et le minimum des  $x_i$  est dû à H. Wolkowicz et G.P.H. Styan dans [27], voir aussi J.-P. Crouzeix et Alberto Seeger dans [?] pour des résultats complémentaires.

**Proposition 11**

$$\bar{x} - \sigma_x \sqrt{n-1} \leq \min_i x_i \leq \bar{x} - \frac{\sigma_x}{\sqrt{n-1}},$$

$$\bar{x} + \frac{\sigma_x}{\sqrt{n-1}} \leq \max_i x_i \leq \bar{x} + \sigma_x \sqrt{n-1},$$

En particulier, dans le cas où tous les  $x_i$  sont positifs, on déduit que :

$$n \ln(\bar{x} - \sigma_x \sqrt{n-1}) \leq \sum_{i=1}^n \ln(x_i) \leq n \ln(\bar{x} + \sigma_x \sqrt{n-1}),$$

où par convention,  $\ln(t) = -\infty$  si  $t \leq 0$ .

**Théorème 8** [8] *On suppose que  $x_i > 0$  pour tout  $i = 1, \dots, n$ . Alors*

$$n \ln(\bar{x} - \sigma_x \sqrt{n-1}) \leq A \leq \sum_{i=1}^n \ln(x_i) \leq B \leq n \ln(\bar{x}),$$

avec

$$A = (n-1) \ln\left(\bar{x} + \frac{\sigma_x}{\sqrt{n-1}}\right) + \ln(\bar{x} - \sigma_x \sqrt{n-1}),$$

et

$$B = \ln(\bar{x} + \sigma_x \sqrt{n-1}) + (n-1) \ln\left(\bar{x} - \frac{\sigma_x}{\sqrt{n-1}}\right).$$

### 3.6 Calcul des fonctions majorantes

Revenons aux équations (3.3) et (3.4), notons par  $\bar{\lambda}$  et  $\sigma_\lambda$  la moyenne et l'écart type (resp.) des  $\lambda_i$  et par  $\|\lambda\|$  la norme euclidienne du vecteur  $\lambda$ . Alors :

$$\|\lambda\|^2 = n(\bar{\lambda}^2 + \sigma_\lambda^2) = \theta''(0) = -\theta'(0),$$

et

$$\theta(t) = nt\bar{\lambda} - t\|\lambda\|^2 - \sum_{i=1}^n \ln(1 + t\lambda_i). \quad (3.5)$$

Notre problème consiste à chercher  $\bar{t} \in ]0, \hat{t}[$  donnant une décroissance significative à la fonction convexe  $\theta$ . Nous avons souligné que le choix le plus naturel,  $\bar{t} = t_{opt}$  où  $\theta'(t_{opt}) = 0$ , présente des complications numériques. On peut en revanche trouver approximativement  $\bar{t}$ , mais cette procédure nécessite également plusieurs évaluations de  $\theta$  et  $\theta'$ . Toutefois, si on utilise une recherche linéaire, il convient de connaître la borne supérieure  $\hat{t}$  du domaine de  $\theta$ , qui est numériquement difficile à obtenir. On prendra une borne inférieure de  $\hat{t}$  fournie par la proposition 11

$$\hat{t}_1 = \sup [t : 1 + t\beta_1 > 0] \quad \text{avec } \beta_1 = \bar{\lambda} - \sigma_\lambda \sqrt{n-1}.$$

Une autre borne inférieure  $\hat{t}_2$  est dûe au fait que  $|\lambda_i| \leq \|\lambda\|$  pour tout  $i$ .

$$\hat{t}_2 = \sup [t : 1 + t\beta_2 > 0] \quad \text{avec } \beta_2 = -\|\lambda\|.$$

Alors,  $0 < \hat{t}_2 \leq \hat{t}_1 \leq \hat{t} \leq +\infty$ . L'inégalité  $\hat{t} \geq \hat{t}_1$  est une conséquence de la proposition 11.

Pour montrer que  $\hat{t}_1 \geq \hat{t}_2$ , il suffit de prouver que  $\|\lambda\|^2 \geq \beta_1^2$ . Cette inégalité est équivalente à

$$0 \leq (n-1)\bar{\lambda}^2 + \sigma_\lambda^2 + 2\sigma_\lambda\bar{\lambda}\sqrt{n-1} = (\bar{\lambda}\sqrt{n-1} + \sigma_\lambda)^2.$$

### 3.6.1 La première fonction majorante

Cette stratégie, consiste à minimiser des approximations majorantes  $G$  de  $\theta$  sur  $[0, \hat{t}[$ . Pour être efficace, cette approximation doit être simple et suffisamment proche de  $\theta$ . Dans notre cas on exige

$$0 = G(0), \quad \|\lambda\|^2 = G''(0) = -G'(0).$$

Alors pour tout  $x_i = 1 + t\lambda_i$ , on a

$$\bar{x} = 1 + t\bar{\lambda} \quad \text{et} \quad \sigma_x = t\sigma_\lambda$$

En appliquant le théorème 8 on obtient

$$-\sum_{i=1}^n \ln(1 + t\lambda_i) \leq -(n-1) \ln(1 + \alpha_0 t) - \ln(1 + \beta_0 t),$$

avec

$$\alpha_0 = \bar{\lambda} + \frac{\sigma_\lambda}{\sqrt{n-1}} \quad \text{et} \quad \beta_0 = \bar{\lambda} - \sigma_\lambda \sqrt{n-1}.$$

Les logarithmes sont bien définis dès que  $t \leq \hat{t}_0$  avec

$$\hat{t}_0 = \begin{cases} -\frac{1}{\beta_0} & \text{si } \beta_0 < 0 \\ +\infty & \text{sinon.} \end{cases}$$

En remplaçant dans (3.5) on en déduit la majoration suivante pour tout  $t \in [0, \hat{t}_0[$

$$\theta(t) \leq G_0(t),$$

tel que

$$G_0(t) = \gamma_0 t - (n-1) \ln(1 + \alpha_0 t) - \ln(1 + \beta_0 t).$$

Rappelons que l'on a

$$\gamma_0 = n\bar{\lambda} - \|\lambda\|^2.$$

On a alors les relations suivantes

$$\gamma_0 + (n-1)\alpha_0 + \beta_0 = (n-1)\alpha_0^2 + \beta_0^2 = \sum_{i=1}^n \lambda_i^2,$$

d'où

$$\gamma_0 = (n-1)(\alpha_0 - \alpha_0^2) + (\beta_0 - \beta_0^2).$$

D'autre part pour tout  $t \in [0, \widehat{t}_0[$

$$G'_0(t) = \gamma_0 - \frac{(n-1)\alpha_0}{1 + \alpha_0 t} - \frac{\beta_0}{1 + \beta_0 t},$$

et

$$G''_0(t) = \frac{(n-1)\alpha_0^2}{(1 + \alpha_0 t)^2} + \frac{\beta_0^2}{(1 + \beta_0 t)^2}.$$

On sait alors que :

$$\begin{aligned} \theta'(0) &= G'_0(t) = -n(\bar{\lambda}^2 + \sigma_\lambda^2) = -\sum_{i=1}^n \lambda_i^2, \\ \theta''(0) &= G''_0(t) = n(\bar{\lambda}^2 + \sigma_\lambda^2) = \sum_{i=1}^n \lambda_i^2 = \text{trace}(E^2). \end{aligned}$$

$G_0$  est strictement convexe sur  $]0, +\infty[$  et  $G'_0(t) < 0$ . Si  $t$  tend vers  $\infty$  et puisque  $G_0$  majore  $\theta$  qui est inf-compacte,  $G_0$  admet un minimum sur  $[0, \widehat{t}_0[$ .

Si  $\widehat{t}_0 < \infty$  alors  $G_0(t)$  tend vers  $\infty$  si  $t$  tend vers  $\widehat{t}_0$ ; Donc  $G_0$  admet un minimum unique sur  $[0, \widehat{t}_0[$ . Ce minimum est obtenu en  $t_{opt}$  telle que  $G'_0(t_{opt}) = 0$ .

On est donc ramené à résoudre l'équation du second degré

$$\alpha_0 \beta_0 \gamma_0 t^2 + ((\alpha_0 + \beta_0)\gamma_0 - n\alpha_0 \beta_0)t + (\gamma_0 - (n-1)\alpha_0 - \beta_0) = 0,$$

les racines de cette équation sont du type

$$t = \frac{-1}{2\alpha_0 \beta_0 \gamma_0} \left[ \alpha_0 \gamma_0 + \beta_0 \gamma_0 - n\alpha_0 \beta_0 \pm \sqrt{[(\alpha_0 - \beta_0)\gamma_0 - n\alpha_0 \beta_0]^2 + 4\alpha_0 \beta_0 \gamma_0 (\alpha_0 - \beta_0)} \right].$$

On prend une seule des deux racines qui appartient à  $[0, \widehat{t}_0[$ .

### 3.6.2 La deuxième fonction majorante

Puisque la fonction

$$\theta(t) = \frac{1}{r}[f_r(y + td) - f_r(y)], \quad y + td \in \widehat{Y},$$

et que la fonction  $f_r$  est inf-compacte et strictement convexe,  $\theta$  est aussi inf-compacte et strictement convexe sur son domaine de définition,  $\theta$  atteint son minimum en un point unique, en outre on a (3.5)

alors :

$$\theta'(t) = n\bar{\lambda} - \|\lambda\|^2 - \sum_{i=1}^n \frac{\lambda_i}{1 + t\lambda_i},$$

et donc :

$$\begin{aligned} \theta'(0) &= -\|\lambda\|^2, \\ \theta''(t) &= \sum_{i=1}^n \frac{\lambda_i^2}{(1 + t\lambda_i)^2}, \end{aligned}$$

d'où :

$$\theta''(0) = \|\lambda\|^2 \geq 0.$$

L'idée consiste à utiliser l'inégalité suivante :

$$-\sum_{i=1}^n \ln(1 + t\lambda_i) \leq (-\|\lambda\| - \sum_{i=1}^n \lambda_i)t - \ln(1 - t\|\lambda\|).$$

En remplaçant dans (3.5) on obtient :

$$\theta(t) \leq -\|\lambda\|(1 + \|\lambda\|)t - \ln(1 - t\|\lambda\|),$$

d'où

$$G_2(t) = -\|\lambda\|(1 + \|\lambda\|)t - \ln(1 - t\|\lambda\|),$$

définie sur  $[0, \widehat{t}_2[$  avec  $\widehat{t}_2 = \frac{1}{\|\lambda\|}$

**Proposition 12** *Pour tout  $t \in [0, \widehat{t}_2[$  on a :*

- a)  $G_2(0) = \theta(0) = 0$  et  $G_2'(0) = \theta'(0) = -\|\lambda\|^2 < 0$ ,
- b)  $G_2''(0) = \theta''(0) = \|\lambda\|^2 > 0$ .
- c)  $\theta(t) \leq G_2(t)$ .

**Preuve 13** a) *A partir de la définition on a :  $G_2(0) = \theta(0) = 0$  et*

$$G_2'(t) = \frac{\|\lambda\|}{1 - t\|\lambda\|} - \|\lambda\|(1 + \|\lambda\|),$$

*ce qui donne  $G_2'(0) = -\|\lambda\|^2 < 0$ , d'où  $\theta'(0) = G_2'(0) < 0$ .*

b) *D'autre part*

$$G_2''(t) = \frac{\|\lambda\|^2}{(1 - t\|\lambda\|)^2} > 0,$$

*d'où  $G_2''(0) = \theta''(0) = \|\lambda\|^2 > 0$ .*

c) *On considère la fonction*

$$\begin{aligned} h(t) &= G_2(t) - \theta(t) \\ &= (-\|\lambda\| - \sum_{i=1}^n \lambda_i)t - \ln(1 - t\|\lambda\|) + \sum_{i=1}^n \ln(1 + t\lambda_i). \end{aligned}$$

*On a par définition  $h(0) = 0$  et pour étudier le signe de la fonction  $h$  on distingue deux cas :*

1. *s'il existe  $i$  tel que  $\|\lambda\| = -\lambda_i$  alors  $h(t) = 0$  pour tout  $t \in [0, \widehat{t}[$ ,*
2. *dans le cas contraire, on sait que  $-\|\lambda\| \leq \lambda_i \leq \|\lambda\|$ .*

*En outre*

$$h'(t) = t \sum_{i=1}^n \lambda_i^2 [(1 - t\|\lambda\|)^{-1} - (1 + t\lambda_i)^{-1}].$$

*Puisque  $1 - t\|\lambda\| < 1 + t\lambda_i$ , pour tout  $i$ , alors  $h'(t) > 0$  pour tout  $t \in [0, \widehat{t}[$ , d'où la fonction  $h(t)$  est strictement croissante et comme  $h(0) = 0$ , alors  $h(t) \geq 0$  pour*

tout  $t \in [0, \widehat{t}]$ , ce qui donne

$$\theta(t) \leq G_2(t) \text{ pour tout } t \in [0, \widehat{t}].$$

**Théorème 9** *La fonction  $G_2$  est strictement convexe. Elle atteint son minimum sur  $[0, \widehat{t}_2[$  au point  $\dot{t} = (1 + \|\lambda\|)^{-1}$  et on a :*

$$G_2(\dot{t}) = \ln(1 + \|\lambda\|) - \|\lambda\|.$$

**Preuve 14** *On note que l'on a :*

$$\begin{aligned} G_2'(t) &= \frac{\|\lambda\|}{1 - t\|\lambda\|} - \|\lambda\|(1 + \|\lambda\|), \\ G_2''(t) &= \frac{\|\lambda\|^2}{(1 - t\|\lambda\|)^2} > 0. \end{aligned}$$

*D'où  $G_2$  est strictement convexe et alors,  $G_2$  atteint son minimum sur  $[0, \widehat{t}_2[$  au point  $\dot{t}$  si et seulement si  $G_2'(\dot{t}) = 0$ , ce qui donne  $\dot{t} = (1 + \|\lambda\|)^{-1}$  et*

$$G_2(\dot{t}) = \ln(1 + \|\lambda\|) - \|\lambda\|.$$

### 3.6.3 La troisième fonction majorante

On peut aussi penser à d'autres fonctions plus simples que  $G_0$ , qui font intervenir un seul logarithme et qui nous permettent de trouver une relation entre  $G_0$  et  $G_2$ .

On considère les fonctions suivantes :

$$G_1(t) = \tilde{\gamma}t - \tilde{\delta} \ln(1 + \tilde{\beta}t), \quad t \in [0, \tilde{t}[ , \quad \tilde{t} = \sup [t : 1 + \tilde{\beta}t > 0],$$

tels que

$$\tilde{\beta} = \beta_0 = \beta_1 = \bar{\lambda} - \sigma_\lambda \sqrt{n-1},$$

et

$$\|\lambda\|^2 = \tilde{\delta}\tilde{\beta}^2 = \tilde{\delta}\tilde{\beta} - \tilde{\gamma}, \quad (3.6)$$

alors

$$\tilde{\delta} = \frac{\|\lambda\|^2}{\tilde{\beta}^2},$$

et

$$\tilde{\gamma} = \tilde{\delta}\tilde{\beta} - \|\lambda\|^2.$$

En outre l'introduction des fonctions de type

$$G(t) = \tilde{\gamma}t - \tilde{\delta}\ln(1 + \tilde{\beta}t), \quad t \in [0, \tilde{t}[,$$

permettent d'avoir une comparaison entre  $G_0, G_1$  et  $G_2$  en admettant (3.6) qui se traduit dans la proposition suivante où l'on voit clairement l'efficacité et l'intérêt majeur apporté par l'introduction de telles fonctions.

**Proposition 15**  $G_i, i = 0; 1; 2$ , est strictement convexe sur  $,G_i(t) \rightarrow +\infty$  lorsque  $t \rightarrow \hat{t}_i$ . En outre,  $\theta(t) \leq G_0(t) \leq G_1(t) \leq G_2(t) \leq +\infty$  pour tout  $t > 0$ .

**Preuve 16** La première inégalité est immédiate. L'inégalité  $\theta(t) \leq G_0(t)$  est une conséquence directe du théorème 8. Posons  $\nu(t) = G_1 - G_0$ . Puisque  $\beta_0 = \beta_1$  et  $\alpha_0 \geq \beta_0$ , on a pour  $t > 0$

$$\nu''(t) = \frac{\delta_1\beta_1^2 - \beta_0^2}{(1 + \beta_0t)^2} - \frac{(n-1)\alpha_0^2}{(1 + \alpha_0t)^2} = \frac{(n-1)\alpha_0^2}{(1 + \beta_0t)^2} - \frac{(n-1)\alpha_0^2}{(1 + \alpha_0t)^2} \geq 0,$$

comme  $\nu(0) = \nu'(0) = 0$ , il vient  $\nu(t) \geq 0$  pour  $t > 0$ .

Puis, on pose  $\mu(t) = G_2 - G_1$ . Alors,  $\mu(0) = \mu'(0) = 0$  et

$$\mu''^2 \left[ \frac{1}{(1 + \beta_2t)^2} - \frac{1}{(1 + \beta_1t)^2} \right] \geq 0.$$

D'où  $\mu(t) \geq 0$  pour tout  $t > 0$ . □

En on déduit que la fonction  $G_i$  atteint son minimum en un point unique  $\bar{t}_i$  qui est la racine de  $G'_i(t) = 0$ . Pour  $i = 1, 2$ , on a

$$\bar{t}_i = \frac{\delta_i}{\gamma_i} - \frac{1}{\beta_i} \quad \text{et} \quad G_i(\bar{t}_i) = \frac{\|\lambda\|^2}{\beta_i} + \frac{\|\lambda\|^2}{\beta_i^2} \ln(1 - \beta).$$

En particulier

$$\bar{t}_2 = \frac{1}{1 + \|\lambda\|} \quad \text{et} \quad G_2(\bar{t}_2) = -\|\lambda\| + \ln(1 + \|\lambda\|).$$

La solution de l'équation  $G_0(t) = 0$  nous ramène à résoudre l'équation du deuxième degré  $t^2 - 2bt + ct = 0$ , où

$$b = \frac{1}{2} \left( \frac{n}{\gamma_0} - \frac{1}{\alpha_0} - \frac{1}{\beta_0} \right) \quad \text{et} \quad c = -\frac{\|\lambda\|^2}{\alpha_0 \beta_0 \gamma_0},$$

dont les racines sont données par  $t = b \pm \sqrt{b^2 - c}$ . Pour  $\bar{t}_0$  on prend la racine qui appartient à l'intervalle  $(0, \hat{t}_0)$ .

Ainsi, les trois racines  $\bar{t}_0$ ,  $\bar{t}_1$  et  $\bar{t}_2$  sont explicitement calculées. Il est clair que

$$\theta(\bar{t}_2) \leq G_2(\bar{t}_2), \quad \theta(\bar{t}_1) \leq G_1(\bar{t}_1) \leq G_1(\bar{t}_2) \leq G_2(\bar{t}_2)$$

et

$$\theta(\bar{t}_0) \leq G_0(\bar{t}_0) \leq G_0(\bar{t}_1) \leq G_1(\bar{t}_1) \leq G_2(\bar{t}_2).$$

### 3.7 Description de l'algorithme

**Initialisation :** On décide pour la stratégie du pas de déplacement et on fixe les paramètres  $\varepsilon > 0$ ,  $r > 0$ ,  $\rho > 0$ ,  $\sigma \in ]0, 1[$ . On démarre avec  $y \in \hat{Y}$ .

**Itération :**

- (a) Calculer  $B = B(y)$  et  $L$  tq  $LL^t = B$ .

- (b) Calculer  $g = b - rb(y)$  et  $H = r\Delta(y)$ .
- (c) Résoudre le système linéaire  $Hd = -g$ . Calculer  $E$ ,  $\text{trace}(E)$  et  $\text{trace}(E^2)$ .
- (d) Calculer  $\bar{\lambda}$  et  $\bar{\sigma}_\lambda$ .
- (e) Obtenir  $\bar{t}$  en utilisant la stratégie des majorantes ou la RL.  
Prendre  $\bar{y} = y + \bar{t}d$ .
- (f) Si  $|b^t y - b^t \bar{y}| > \rho nr$ , alors  $y = \bar{y}$  et aller en (a).

si  $nr > \varepsilon$ , alors  $y = \bar{y}$ ,  $r = \sigma r$  et aller en (a).

(h) **Stop** :  $\bar{y}$  est une solution approximative de (D).

On sait de ce qui précède, que la solution optimale de  $(D_r)$  est une approximation de la solution de (D), plus  $r$  est proche de zéro plus l'approximation est bonne. Malheureusement quand  $r$  s'approche de zéro, le problème  $(D_r)$  devient mal conditionné. C'est la raison pour laquelle on utilisera au début de l'itération des valeurs de  $r$  pas très proches de zéro, vérifiant le test  $nr < \varepsilon$ . On peut expliquer la mise à jour de  $r$  comme suit : si  $y(r)$  est une solution exacte de  $(D_r)$ , alors  $b^t y(r) \in [m_D, m_D + nr]$ , il est donc inutile de continuer le calcul des itérés quand  $|b^t y - b^t \bar{y}| \leq \rho nr$ . Pour  $\rho$ , on peut considérer les valeurs 1, 2,  $\sigma = 0.125$ .

Le pas de déplacement  $\bar{t}$  sera déterminé par l'une des quatre stratégies suivantes :

- Ls : une recherche linéaire classique de type Armijo-Goldstein.
- Stratégies  $S_i, i = 0, 1, 2$  :  $\bar{t} = \bar{t}_i$  avec  $\bar{t}_i$  minimisant la majorante  $G_i$ .

## 3.8 Tests numériques

La programmation est réalisée sur une station D810 en langage Delphi 5.

### 3.8.1 Exemple 1 "Exemple cube"

$n = 2m$ ,  $C$  est la matrice identité  $n \times n$ ,  $b = (2, \dots, 2)^t \in \mathfrak{R}^m$  et les éléments de la matrice  $A_k$ ,  $k = 1, \dots, m$ , sont données par :

$$A_k[i, j] = \begin{cases} 1 & \text{si } i = j = k & \text{ou } i = j = k + m, \\ a^2 & \text{si } i = j = k + 1 & \text{ou } i = j = k + m + 1, \\ -a & \text{si } i = k, j = k + 1 & \text{ou } i = k + m, j = k + m + 1, \\ -a & \text{si } i = k + 1, j = k & \text{ou } i = k + m + 1, j = k + m, \\ 0 & \text{dans les autres cas.} \end{cases}$$

$a \in \mathfrak{R}$  est une constante donnée.

**Test 1 :**  $(m, n) = (50, 100)$  et  $a = 1, C = -I_n, b = (2, \dots, 2)^t$ .  $y_0 = (3, \dots, 3, \frac{3}{2})^t$  est un point réalisable. on prend  $\rho = 1, \sigma = 0.125, r = 0.3, \varepsilon = 0.1$ .

Le tableau suivant résume les résultats obtenus.

Stratégies	Temps	Nbre d'itération
$S_0$	376 s	24
$S_1$	380 s	24
$S_2$	820 s	32
Ls	divg	divg

divg : l'algorithme diverge.

**Calcul de la fonction  $\theta$  et des trois fonctions majorantes  $G_0, G_1, G_2$**

$$\theta(t) = -415.98t - \ln(1 - 2.4930t) - \ln(1 - 8.6555t) - \ln(1 - 12.090t) - \ln(1 - 12.$$

372t)

$$G_0(t) = -415.97t - 3 \ln(1 - 6.6050t) - \ln(1 - 15.795t)$$

$$G_1(t) = -404.44t - 1.5246 \ln(1 - 15.795t)$$

$$G_2(t) = -399.87t - \ln(1 - 19.503t)$$

**Dessin des 4 fonctions pour  $(m, n) = (2, 4)$  à l'initialisation :**

`:/swp55/temp/graphics/swp00001.pdf`

Le noir pour  $\theta$

Le rouge pour  $G_0$

Le vert pour  $G_1$

Le violet pour  $G_2$

**Test 2 :**  $(m, n) = (50, 100)$  et  $a = 2, C = -I_n, b = (2, \dots, 2)^t$ .  $y_0 = (\frac{3}{2}, \dots, \frac{3}{2})^t$  est un point réalisable. on prend  $\rho = 1, \sigma = 0.125, r = 0.3, \varepsilon = 0.1$ .

Le tableau suivant résume les résultats obtenus.

Stratégies	Temps	Nbre d'itération
$S_0$	289 s	19
$S_1$	292 s	19
$S_2$	572 s	36
Ls	dvg	dvg

dvg : l'algorithme diverge.

**Calcul de la fonction  $\theta$  et des trois fonctions majorantes  $G_0, G_1, G_2$**

$$\theta(t) = -165.27t - \ln(1 - 0.12286t) - \ln(1 - 4.7752t) - \ln(1 - 5.8458t) - \ln(1 - 9.3888t)$$

$$G_0(t) = -165.27t - 3 \ln(1 - 3.1227t) - \ln(1 - 10.765t)$$

$$G_1(t) = -158.62t - 1.2524 \ln(1 - 10.765t)$$

$$G_2(t) = -157.18t - \ln(1 - 12.6419t)$$

**Dessin des 4 fonctions pour  $(m, n) = (2, 4)$  à l'initialisation :**

`:/swp55/temp/graphics/swp00002.pdf`

Le noir pour  $\theta$

Le rouge pour  $G_0$

Le vert pour  $G_1$

Le violet pour  $G_2$

**Test 3 :**  $(m, n) = (50, 100)$  et  $a = 5, C = -2I_n, b = (2, \dots, 2)^t. y_0 = (0, \dots, 0)^t$  est un point réalisable. on prend  $\rho = 1, \sigma = 0.125, r = 0.3, \varepsilon = 0.1.$

Le tableau suivant résume les résultats obtenus.

Stratégies	Temps	Nbre d'itération
$S_0$	120 s	9
$S_1$	130 s	9
$S_2$	288 s	14
Ls	divg	divg

divg : l'algorithme diverge.

**Calcul de la fonction  $\theta$  et des trois fonctions majorantes  $G_0, G_1, G_2$**

$$\theta(t) = 0.59699t - \ln(1 + 0.79501t) - \ln(1 + 0.7264t) - \ln(1 + 0.37869t) - \ln(1 + 1.$$

$$0207 \times 10^{-7}t)$$

$$G_0(t) = 0.59702t - 3 \ln(1 + 0.65771t) - \ln(1 - 7.3026 \times 10^{-2}t)$$

$$G_1(t) = -19.148t - 244.36 \ln(1 - 7.3026 \times 10^{-2}t)$$

$$G_2(t) = -2.4445t - \ln(1 - 1.1415t)$$

**Dessin des 4 fonctions pour  $(m, n) = (2, 4)$  à l'initialisation :**

Le noir pour  $\theta$

Le rouge pour  $G_0$

Le vert pour  $G_1$

Le violet pour  $G_2$

**Test 4 :**  $(m, n) = (50, 100)$  et  $a = 2, C = -2I_n, b = (2, \dots, 2)^t. y_0 = (0, \dots, 0)^t$  est un point réalisable. on prend  $\rho = 1, \sigma = 0.125, r = 0.3, \varepsilon = 0.1.$

Le tableau suivant résume les résultats obtenus.

Stratégies	Temps	Nbre d'itération
$S_0$	220 s	15
$S_1$	240 s	15
$S_2$	326 s	19
Ls	divg	divg

divg : l'algorithme diverge.

**Calcul de la fonction  $\theta$  et des trois fonctions majorantes  $G_0, G_1, G_2$**

$$\theta(t) = -3.7525t - \ln(1 + 5.0471 \times 10^{-3}t) - \ln(1 + 5.778 \times 10^{-5}t) - \ln(1 - 0.27797t) - 3.$$

$$4023t - \ln(1 - 1.4111t)$$

$$G_0(t) = -3.7525t - 3 \ln(1 - 8.4411 \times 10^{-2}t) - \ln(1 - 1.4308t)$$

$$G_1(t) = -3.5142t - 1.0104 \ln(1 - 1.4308t)$$

$$G_2(t) = -3.5066t - \ln(1 - 1.4382t)$$

**Dessin des 4 fonctions pour  $(m, n) = (2, 4)$  à l'initialisation :**

Le noir pour  $\theta$

Le rouge pour  $G_0$

Le vert pour  $G_1$

Le violet pour  $G_2$

**Test 5 :**  $(m, n) = (50, 100)$  et  $a = 0, C = I_n, b = (2, \dots, 2)^t$ .  $y_0 = (\frac{3}{2}, \dots, \frac{3}{2})^t$  est un point réalisable. on prend  $\rho = 1, \sigma = 0.125, r = 0.3, \varepsilon = 0.1$ .

Le tableau suivant résume les résultats obtenus.

Stratégies	Temps	Nbre d'itération
$S_0$	38 s	3
$S_1$	38 s	4
$S_2$	60 s	dvg
Ls	dvg	dvg

dvg : l'algorithme diverge.

**Calcul de la fonction  $\theta$  et des trois fonctions majorantes  $G_0, G_1, G_2$**

$$\theta(t) = -4.4444t - 4 \ln(1 - 0.66665t)$$

$$G_0(t) = -4.4443t - 3 \ln(1 - 0.66569t) - \ln(1 - 0.66954t)$$

$$G_1(t) = -4.4328t - 3.9656 \ln(1 - 0.66954t)$$

$$G_2(t) = -3.1110t - \ln(1 - 1.3333t)$$

**Dessin des 4 fonctions pour  $(m, n) = (2, 4)$  à l'initialisation :**

Le noir pour  $\theta$

Le rouge pour  $G_0$

Le vert pour  $G_1$

Le violet pour  $G_2$

### 3.8.2 Exemple 2 "Exemple artificiel"

Soit le problème suivant pour  $m = n$  :

$$\begin{cases} \max \text{trace} CX \\ x_{ii} + x_{i+1,i+1} - x_{i,i+1} - x_{i+1,i} = 2 \dots\dots\dots(P) \\ X \in S_n^+(\mathfrak{R}) \end{cases}$$

Le dual de ce problème s'écrira :

$$\inf_y [b^t y : \sum_{i=1}^m y_i A_i - C \in S_n^+, y \in \mathfrak{R}^m] \dots\dots\dots(D)$$

avec :

$$A_k[i, j] = \begin{cases} 1 & \text{si } i = j = k & \text{ou } i = j = k + 1, \\ -1 & \text{si } i = k, j = k + 1 & \text{ou } i = k + 1, j = k, \text{ pour } k = 1..n - 1 \\ 0 & \text{dans les autres cas.} \\ & A_k \text{ est la matrice identité pour } k = n \end{cases}$$

$$y_0[i] = \begin{cases} n + 1 & \text{si } i = n \\ 0 & \text{dans les autres cas.} \end{cases},$$

$$C[i, j] = 1 \text{ pour } i = 1..n \text{ et } j = 1..n,$$

$$b[i] = 2 \text{ pour } i = 1..n$$

$$\text{Pour } n = 3, y_0 = (0, 0, 4)^t, C = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, b = (2, 2, 2)^t$$

$$A_1 = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, A_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{pmatrix}, A_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Le tableau suivant résume les résultats obtenus.

Stratégies	Temps	Nbre d'itération
$S_0$	1 s	5
$S_1$	1 s	5
$S_2$	1 s	5
Ls	dvg	dvg

dvg : l'algorithme diverge.

**Calcul de la fonction  $\theta$  et des trois fonctions majorantes  $G_0, G_1, G_2$**

$$\theta(t) = -1133.4t - \ln(1 + 4.211t) - \ln(1 - 1.8449t) - \ln(1 - 9.6678t) - \ln(1 - 19.$$

$$565t) - \ln(1 - 24.185t)$$

$$G_0(t) = -1133.4t - 4 \ln(1 - 4.9135t) - \ln(1 - 31.396t)$$

$$G_1(t) = -1116.8t - 1.0980 \ln(1 - 31.396t)$$

$$G_2(t) = -1115.2t - \ln(1 - 32.898t)$$

**Dessin des 4 fonctions pour  $(m, n) = (2, 4)$  à l'initialisation :**

Le noir pour  $\theta$

Le rouge pour  $G_0$

Le vert pour  $G_1$

Le violet pour  $G_2$

# Conclusion

Dans notre étude, nous avons abordé des questions ouvertes concernant la programmation semi-définie (*SDP*) linéaire.

Les réponses que nous avons apporté sont d'une valeur algorithmique et numérique très intéressante et ouvrent plusieurs perspectives.

L'approche barrière logarithmique pour (*SDP*), est un véritable exploit, sur l'aspect théorique et algorithmique. L'aspect numérique pourra être poussé à un niveau de performance très appréciable pour la pratique.

La technique de fonctions majorantes pour déterminer le pas du déplacement suivant la direction de descente, est une alternative très fiable qui devra se confirmer comme la technique de choix aussi bien pour (*SDP*) que pour d'autres classes de problèmes d'optimisation.

# Bibliographie

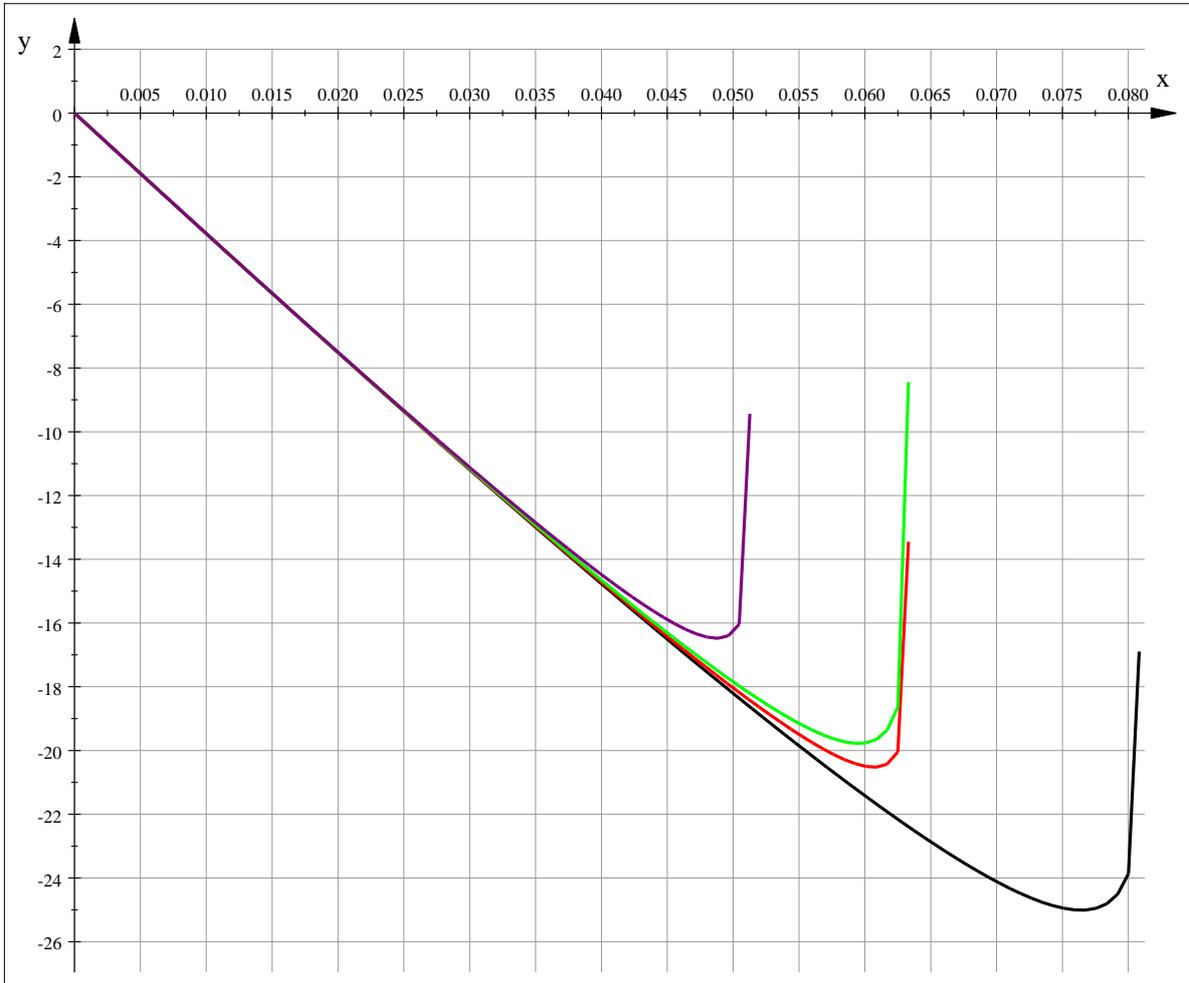
- [1] F. Alizadeh, Interior point methods in semidefinite programming with application to combinatorial optimization, *SIAM Journal on Optimization* 5 (1995), 13–55.
- [2] F. Alizadeh, J.-P.Haberly, and M.-L. Overton, Primal-dual interior-point methods for semidefinite programming, convergence rates, stability and numerical results, *SIAM Journal on Optimization* 8 (1998), 746-768.
- [3] D. Benterki, Sur l'étude des performances de l'algorithme de Karmarkar pour la programmation linéaire, Thèse de magister, Département de mathématique, Université de Annaba (1992).
- [4] D. Benterki, Résolution des problèmes de programmation semi-définie par des méthodes de réduction du potentiel, Thèse de doctorat, Département de mathématique, Université Ferhat Abbas, Sétif (2004).
- [5] D. Benterki, J.P. Crouzeix, B. Merikhi, A feasible primal algorithm for linear semidefinite programming, *Modelling, Computation and optimization in information systems and managements sciences* (2004), pp. 114-120.
- [6] D. Benterki, B. Merikhi, A modified algorithm for the strict feasibility problem, *RAIRO Oper. Res.* 35 (2001), pp. 395-399.
- [7] J.F. Bonnans, J.-C. Gilbert, C. Lemaréchal, and C. Sagastizabal, *Numerical optimization, theoretical and practical aspects*. Springer-Verlag, 2003.
- [8] J.P.Crouzeix, B.Merikhi, A logarithm barrier method for semidefinite programming, *R.A.I.R.O-Oper. Res.* 42 (2008), pp. 123-139.

- [9] A. Coulibaly, Méthode de points intérieurs en programmation linéaire, Thèse de doctorat, Université Blaise Pascal, Clermont-Ferrand (1994).
- [10] J.-P. Crouzeix, A. Seeger, New bounds for the extreme values of a finite sample of real numbers, *Journal of Mathematical Analysis and Applications* 197 (1996), pp. 411-426.
- [11] A. V. Fiacco and Mc Cormick, *The nonlinear programming sequential unconstrained minimization techniques* (1968), Wiley, New-York.
- [12] M. Halicka, E. De Klerk, C. Roos, On the convergence of the central path in semi-definite optimization, *SIAM Journal on Optimization*, 12 (2002), pp. 1090-1099.
- [13] P. Huard, Resolution of mathematical programming with non linear constraints by the method of centers, *Non linear programming* (J. Abadie, ed) North-Holland, Amsterdam(1967).
- [14] J. Ji, F.A. Potra, R. Sheng, On the local convergence of a predictor-corrector method for semidefinite programming, *SIAM Journal on Optimization*, 10 (1999), pp. 195-210.
- [15] L.G. Kachyan, A polynomial algorithm in linear programming, *Dokl. Akad. Nauk SSSR*, 244 1093-1096 (in Russian); trans. in *Soviet Math. Dokl.* 20. pp. 191-194.
- [16] N.K. Karmarkar : A new polynomial time 395. algorithm for linear programming. *Combinatorica* 4, pp. 373-395.
- [17] A. Keraghel, Étude adaptative et comparative des principales variantes dans l'algorithme de Karmarkar, Thèse de doctorat, Université Joseph Fourier, Grenoble, France (1989).
- [18] A. Keraghel, *Analyse convexe, Théorie fondamentale et exercices*, Université de Sétif, Sétif, Algérie.
- [19] M. Kojima, S. Shindoh, S. Hara, Interior point methods for the monotone semi-definite linear complementarity problem in symmetric matrices. *SIAM Journal on Optimization* 7 (1997), pp. 86-125.

- [20] M. Minoux, *Programmation mathématique : théorie et algorithmes*, tome1, Dunod, Paris (1983).
- [21] R.D.C. Monteiro, Primal-dual path-following algorithms for semidefinite programming, *SIAM Journal on Optimization*, 7 (1997), pp. 663-678.
- [22] Y.-E. Nesterov, A. Nemirovski, *Optimization over positive semidefinite matrices : Mathematical background and user's manual*, Technical report, Central economic and mathematical institute, USSR academy of science, Moscow, USSR(1990).
- [23] R.-T. Rockafellar, *Convex analysis*, Princeton University Press, New Jerzy, (1970).
- [24] H. Shor, Utilization of the operations of space dilatation in the minimization of convex functions. *Cybernetics* 6 (1970), pp. 7-15.
- [25] M. J. Todd, K. C. Toh, R. H. Tütüncü, On the Nesterov-Todd direction in semidefinite programming, *SIAM Journal on Optimization*, 8 (1998), pp. 769-796.
- [26] L. Vanderbeghe, S. Boyd, Positive definite programming, *SIAM Review*, 38 (1996), pp. 49-95.
- [27] H.Wolkowicz, G.-P.-H. Styan, Bounds for eigenvalues using traces, *Linear Algebra and Appl.*29 (1980), pp. 471-506.

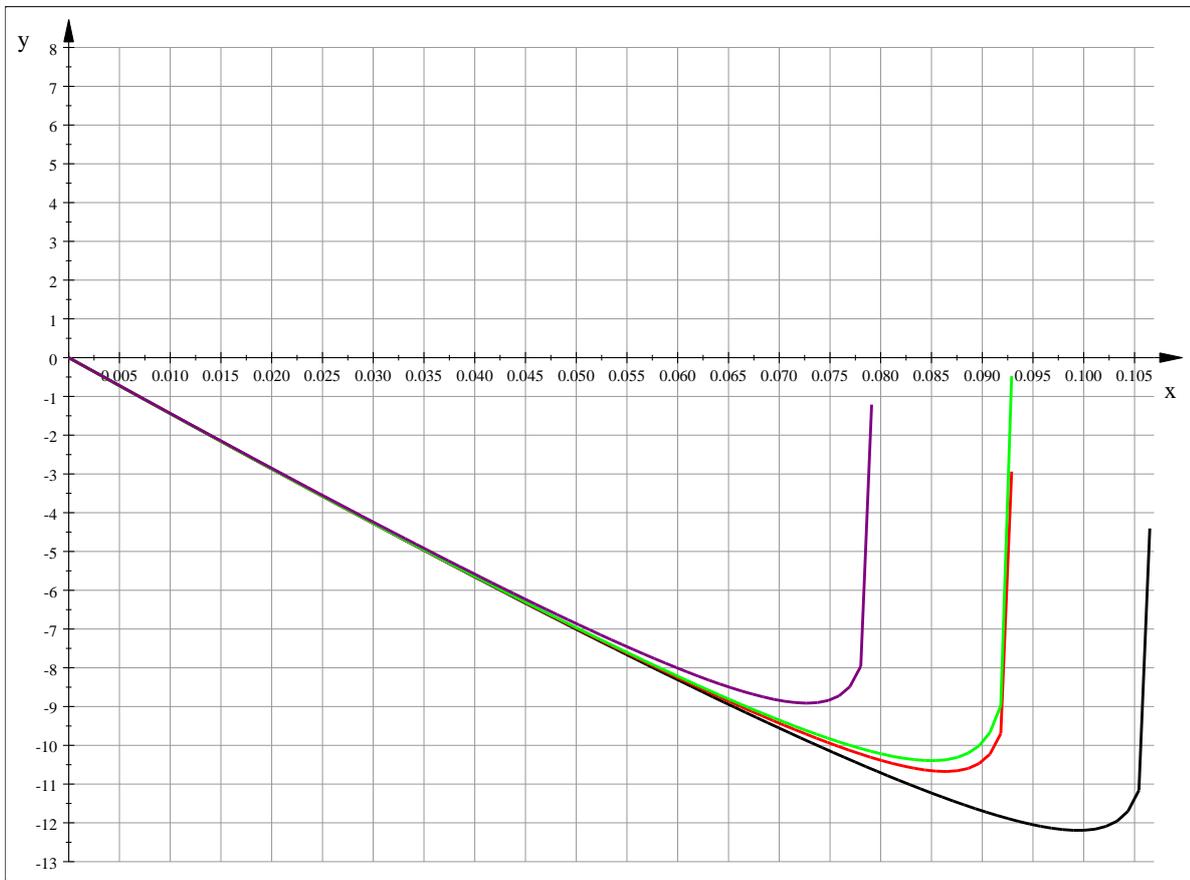
Exemple cube :

Test 1 : Dessin des 4 fonctions  $\theta, G_0, G_1$  et  $G_2$  à l'initialisation pour  $(m,n)=(2,4)$ :



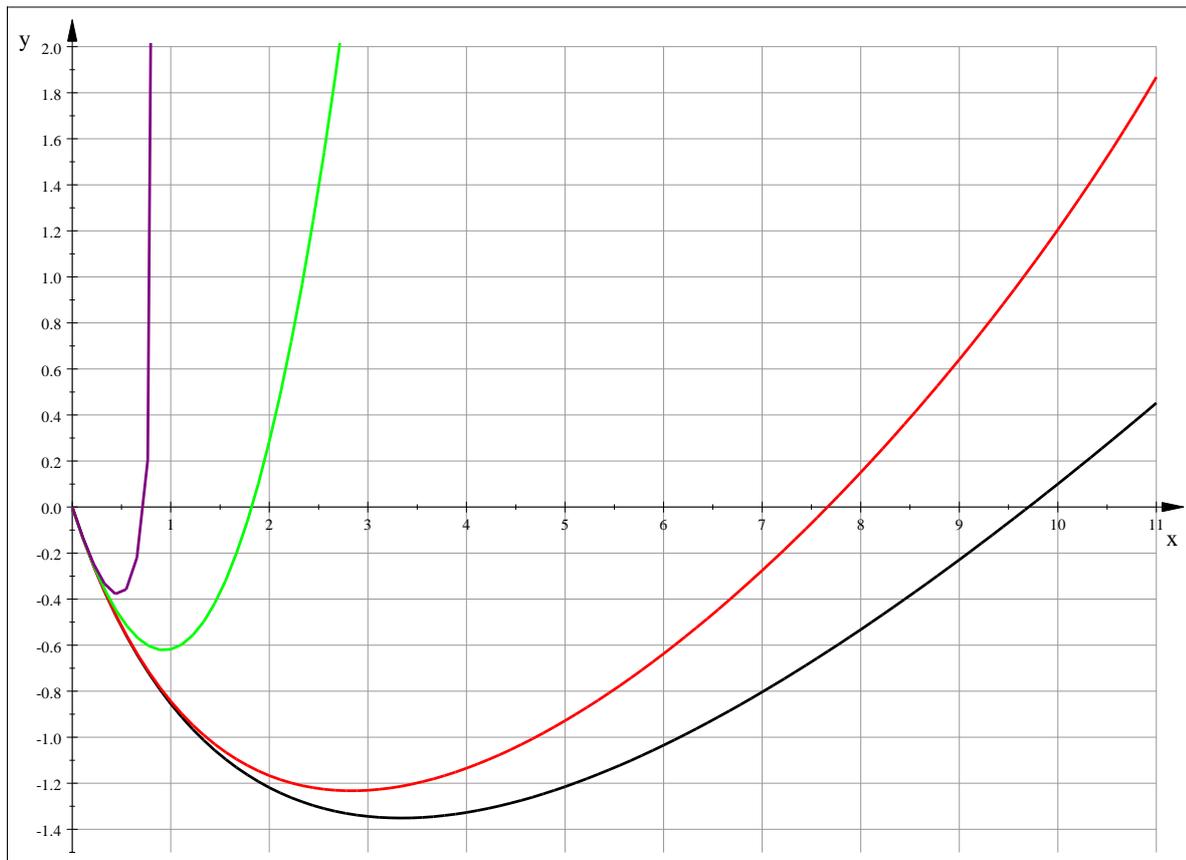
- pour  $\theta$
- pour  $G_0$
- pour  $G_1$
- pour  $G_2$

**Test 2 : Dessin des 4 fonctions  $\theta, G_0, G_1$  et  $G_2$  à l'initialisation pour  $(m,n)=(2,4)$ :**



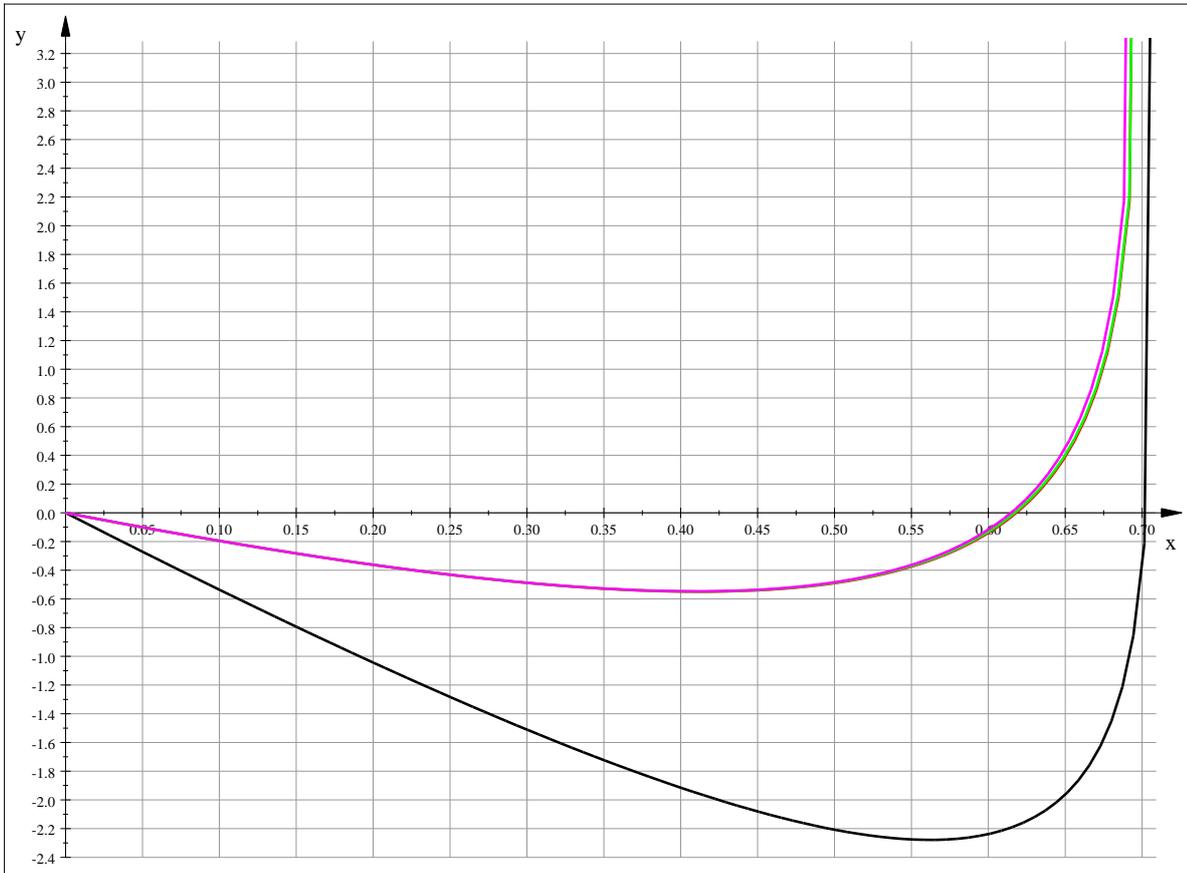
- pour  $\theta$
- pour  $G_0$
- pour  $G_1$
- pour  $G_2$

**Test 3 : Dessin des 4 fonctions  $\theta, G_0, G_1$  et  $G_2$  à l'initialisation pour  $(m,n)=(2,4)$ :**



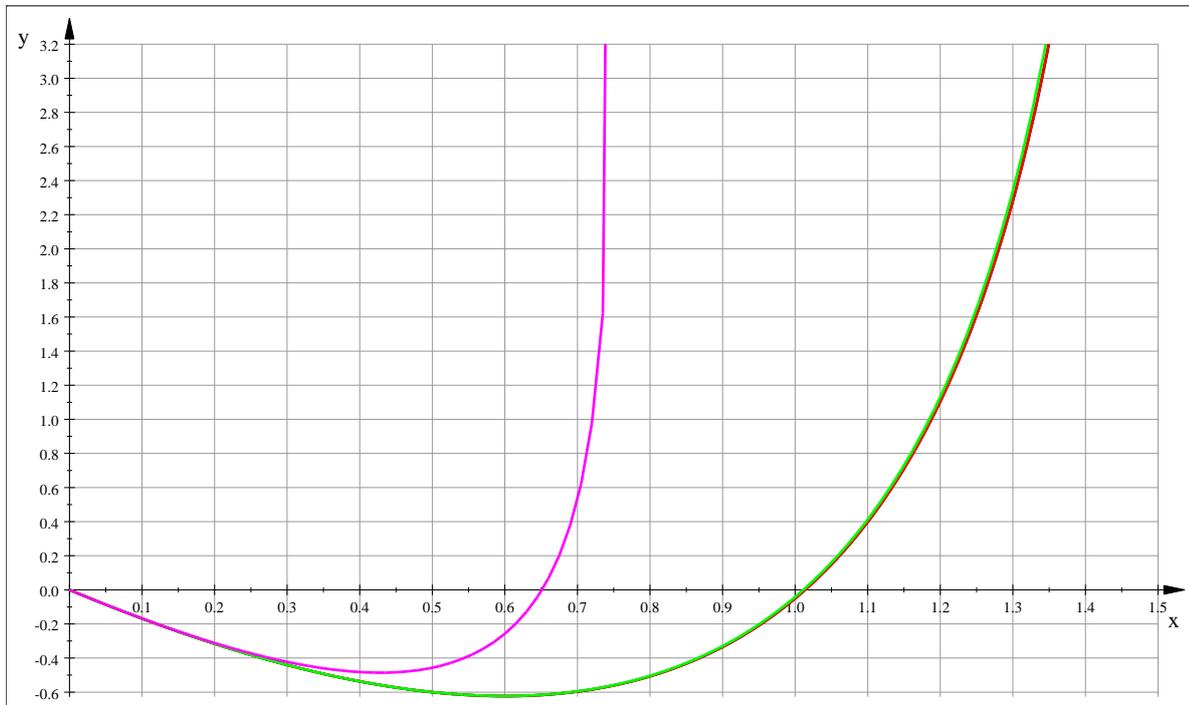
- pour  $\theta$
- pour  $G_0$
- pour  $G_1$
- pour  $G_2$

**Test 4 : Dessin des 4 fonctions  $\theta, G_0, G_1$  et  $G_2$  à l'initialisation pour  $(m,n)=(2,4)$ :**



- pour  $\theta$
- pour  $G_0$
- pour  $G_1$
- pour  $G_2$

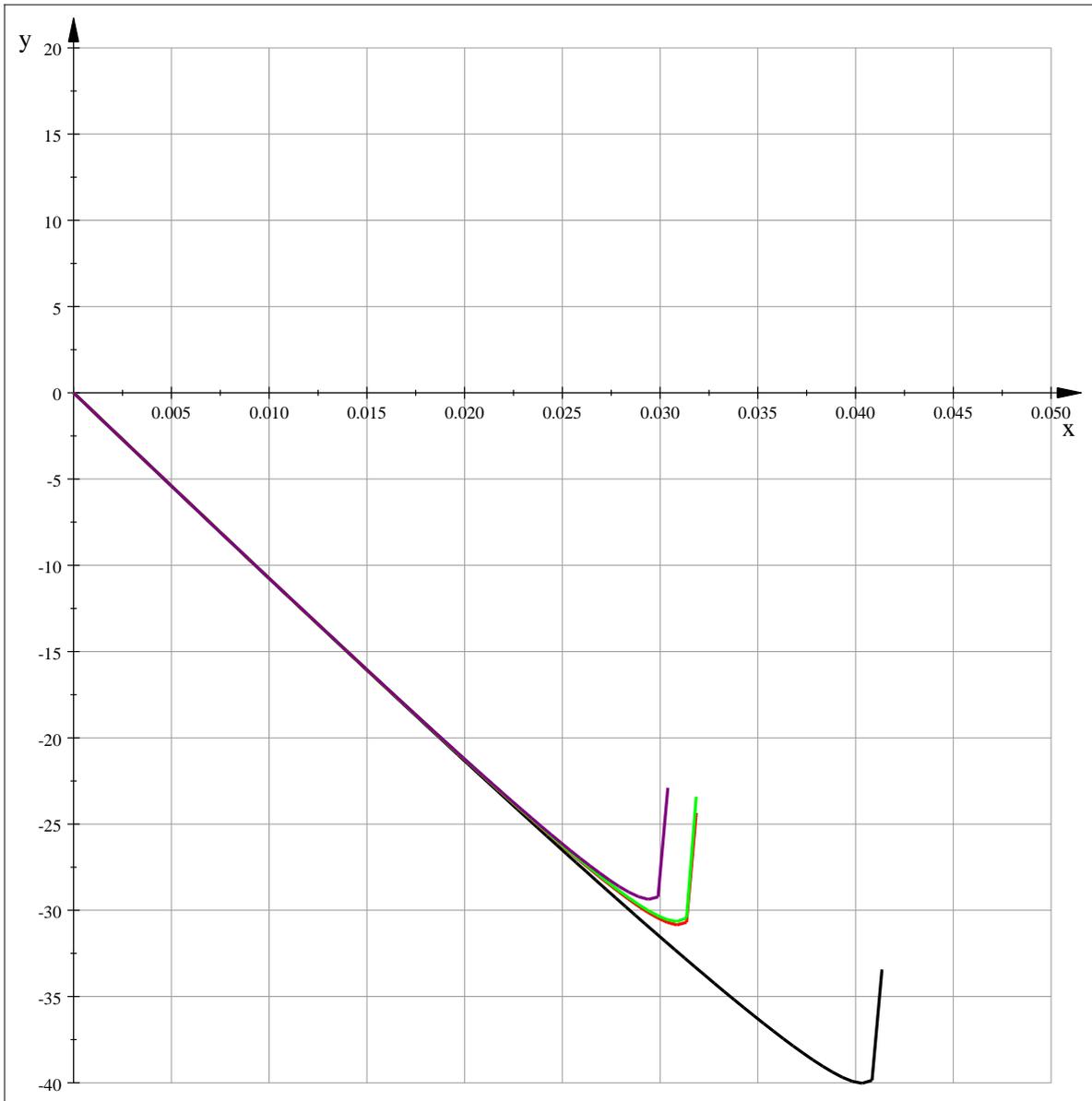
**Test 5 : Dessin des 4 fonctions  $\theta$ ,  $G_0$ ,  $G_1$  et  $G_2$  à l'initialisation pour  $(m,n)=(2,4)$ :**



- pour  $\theta$
- pour  $G_0$
- pour  $G_1$
- pour  $G_2$

**Exemple artificiel :**

**Dessin des 4 fonctions  $\theta, G_0, G_1$  et  $G_2$  à l'initialisation pour  $(m,n)=(2,4)$ :**



- pour  $\theta$
- pour  $G_0$
- pour  $G_1$
- pour  $G_2$

# UNE RECHERCHE LINEAIRE AVEC DES FONCTIONS MAJORANTES DANS CERTAINS

## PROBLEMES D'OPTIMISATION

### ملخص:

البرمجة الرياضية نصف المعرفة هي تعميم للبرمجة الخطية, تترجم العديد من المسائل التطبيقية, كما تشمل مسائل رياضية هامة معروفة ببعض الصعوبات عند المعالجة.

تطور هذا الميدان خلال العشرية الأخيرة بوتيرة مذهلة, منطلقها النظرية الأساسية للبرمجة الخطية وكذلك العناصر التجديدية الهائلة التي أتت بها طرق البرمجة الخطية, البحث الخطي و النقاط الداخلية.

تتناول هذه المذكرة دراسة مسألة البرمجة نصف المعرفة الخطية و تحديدا طريقة الحاجز اللوغاريتمي حيث تطرقنا بالبحث إلى الجانب النظري و الخوارزمي. و كان لإدراج الدوال الحدية لحساب الخطوات نتائج فعالة في تقليل كلفة العمليات الحسابية مقارنة بطرق البحث التقليدية.

التجارب العددية المرفقة بهاته المساهمة كانت مشجعة و نقترح مجالات بحث واسعة.

**كلمات مفتاحية:** البرمجة الخطية، البرمجة نصف المعرفة، البحث الخطي، طريقة الحاجز اللوغاريتمي، الدوال الحدية.

### Abstract:

The semidefinite programming is an extension of the linear programming which has many applications and covers important problems of optimization which present some difficulties in finding the solution.

Based on the fundamental theory of linear programming and the great success of the algorithmic interior point methods, this methodology has developed very rapidly in the last decade.

In this memory we deal with the linear semidefinite programming problems in particularly the logarithm barrier method for semidefinite programming and some theoretical and algorithmic contribution of this method by introducing in this case a new approach based on upper-approximaty functions which reduce the computational cost of the algorithm compared with classical methods.

Several associated experimentations encourage and propose large domains of research.

**Key words:** Linear Programming, Semidefinite Programming, Linear Research, Logarithm Barrier Method, Boundary functions.

### Résumé :

La programmation semi-définie est une extension de la programmation linéaire qui traduit plusieurs problèmes d'applications, mais aussi couvre des problèmes d'optimisation importants, qui présentent des difficultés de résolution.

Sa méthodologie, évolue depuis une décennie avec un rythme impressionnant en prenant pour base la théorie fondamentale de la programmation linéaire, la recherche linéaire et les développements algorithmiques grandioses des méthodes de points intérieurs.

On traite dans ce mémoire les problèmes de programmation semi-définie linéaire et en particulier les méthodes barrières logarithmiques sur le plan théorique et algorithmique. L'introduction de la nouvelle approche des fonctions majorantes a contribué efficacement à la réduction du coût calculatoire de l'algorithme comparativement aux méthodes de recherche linéaire classique.

Les expérimentations numériques présentées dans ce mémoire sont encourageantes et ouvrent de nouvelles perspectives dans ce domaine de recherche.

**Mots clés :** Programmation Linéaire, Programmation Semi-Définie, Recherche Linéaire, Méthode Barrière Logarithmique, Fonctions barrières.