Problèmes d'approximation matricielle linéaires coniques: Approches par Projections et via Optimisation sous contraintes de semi-définie positivité

Pawoumodom Ledogada Takouda

THÈSE

présentée en vue de l'obtention du

Doctorat de l'Université Paul Sabatier - Toulouse III.

Section : Mathématiques Appliquées.

Spécialité : Analyse Convexe et Optimisation numérique.

par

Pawoumodom Ledogada TAKOUDA

Problèmes d'approximation matricielle linéaires coniques : Approches par projections et via Optimisation sous contraintes de semidéfinie positivité.

Rapporteurs :

P. L. Combettes	Professeur à l'Université Pierre et Marie Curie - Paris VI
A. Lewis	Professeur à la Simon Fraser University, Vancouver, Canada

Thèse soutenue le lundi 29 Septembre 2003 devant le jury composé de :

D. Azé	Professeur à l'Université Paul Sabatier - Toulouse III	(Examinateur)
P. L. Combettes	Professeur à l'Université Pierre et Marie Curie - Paris VI	(Rapporteur)
J.-B. Hiriart-Urruty	Professeur à l'Université Paul Sabatier - Toulouse III	(Co-directeur de Thèse)
M. Mongeau	Maître de Conférences HDR à l'Université Paul Sabatier - Toulouse III	(Co-directeur de Thèse)
D. Noll	Professeur à l'Université Paul Sabatier - Toulouse III	(Examinateur)
J-P. Penot	Professeur à l'Université de Pau et Pays de l'Adour	(Examinateur)

Laboratoire de Mathématiques appliqués à l'Industrie et la Physique (MIP)
Equations aux Dérivées Partielles - Optimisation - Modélisation - Calcul Scientifique
UMR 5640 Université P. Sabatier UFR MIG
118, Route de Narbonne 31062 Toulouse Cedex 04 - France

Problèmes d'approximation matricielle linéaires coniques : Approches par projections et via Optimisation sous contraintes de semidéfinie positivité.

Pawoumodom Ledogada TAKOUDA

4 février 2004

Table des matières

Table des figures

Introduction

Nous présentons dans cette thèse l'étude et la comparaison de deux approches numériques de résolutions de problèmes d'approximation matricielle linéaire conique. Nous appelons problème d'approximation tout problème dans un espace normé $\mathbb{E}$ qui consiste à trouver, pour un point a donné, le point d'un sous-ensemble $\mathcal{V}$ de $\mathbb{E}$, formés par des éléments ayant tous une certaine propriété, qui en est le plus proche au sens d'une norme donnée. On parle de problème matriciel lorsque l'on se restreint à considérer un espace formé de matrices. Les problèmes d'approximation matricielle proviennent de différentes situations pratiques dans des domaines aussi variés que l'Analyse numérique, les Statistiques et la Finance, les Sciences sociales, etc.

Nous nous sommes placé dans un espace de matrices euclidien, et nous nous sommes intéressé aux cas où le sous-ensemble $\mathcal{V}$ évoqué ci-dessus a la particularité d'être l'intersection d'un sous-espace (affine ou linéaire) et d'un cône convexe fermé. De nombreux problèmes présentent cette structure particulière. En Théorie du choix social, une des procédures destinées à agréger en une préférence collective des préférences individuelles exprimées sur un certain nombre de possibilités conduit à chercher la matrice bistochastique la plus proche d'une matrice dépendante des données du problème. En analyse de risques financiers, un des plus anciens modèles de mesure de ce risque nécessite la connaissance de la matrice de corrélation associée à un portefeuille d'actions, laquelle doit être calculée à partir de cours d'actions dont on ne dispose pas forcément en totalité. La matrice effectivement calculée doit être calibrée pour maintenir ses propriétés de matrice de corrélation.

D'une manière générale, on peut voir que les problèmes d'approximation matricielle interviennent à l'intérieur d'un processus de décision. Ils doivent donc pouvoir être résolus rapidement, et si nécessaire, autant de fois que souhaité par l'utilisateur. Il faut donc dériver pour eux des solutions algorithmiques et numériques capables de répondre positivement à ce cahier de charges. C'est l'objectif que nous nous donnons dans ce travail.

Cette thèse est organisée comme suit. Nous présentons au chapitre 1, de manière plus concise, la notion de problème d'approximation matricielle. Nous y précisons les hypothèses que nous avons faites, et le contexte dans lequel nous allons travailler. Le chapitre se termine par une présentation rapide des problèmes concrets d'approximation qui vont nous intéresser, ainsi que des différentes approches possibles pour leur résolution. Le chapitre 2 introduit les notions de projections, ainsi que les algorithmes dits de projections. Nous présentons plus succinctement ces mé-

thodes, leurs principes, et nous insistons plus particulièrement sur les algorithmes de projections alternées. Le chapitre 3 porte sur l'étude du problème d'approximation par matrices bistochastiques. Nous rappelons pour commencer quelques propriétés de ces matrices, et nous proposons en particulier une démonstration originale de Théorème de Birkhoff. Nous envisageons alors une étude directe, par calculs, de ce problème. Puis, devant notre échec, nous étudions et mettons en œuvre différentes approches numériques de résolution. Nous terminons le chapitre par une application pratique : la résolution de problèmes d'agrégation de préférences généraux, en utilisant l'une des approches numériques que nous avons testées. Ceci permet de voir l'intérêt des solutions algorithmiques que nous avons mises en œuvre. Le chapitre suivant est d'un tout autre ordre. Il présente les problèmes dits d'optimisation sous contraintes de semi-définie positivité, qui ont connu un boom en termes de recherche ces dix dernières années. Nous nous intéressons au plus près aux algorithmes de points intérieurs qui servent à les résoudre. Nous présentons une démarche classique de ces méthodes, puis une nouvelle, qui n'a connu jusqu'à présent qu'une seule expérimentation, qui tente du mieux possible d'utiliser l'expertise accumulée depuis des années par l'Analyse numérique. Enfin, nous terminons, au chapitre 5, avec l'étude de notre second problème d'approximation : l'approximation par matrices de corrélation. Nous résolvons ce problème en utilisant l'optimisation sur les cônes homogènes auto-duaux, dans un premier temps. Puis, nous dérivons pour lui un algorithme de type points intérieurs suivant la démarche nouvelle que nous avons évoquée plus haut. Finalement, nous comparons les performances de ces algorithmes entre eux, puis avec celui provenant de l'approche par projection alternées.

Chapitre 1

Notions d'approximation matricielle

1.1 Introduction et notations

1.1.1 Notion d'approximation linéaire conique

Dans de nombreux domaines, on est confronté à des situations qui, une fois modélisées, se ramènent à chercher un élément ayant des propriétés données qui soit "le plus proche" (dans un sens à préciser) d'un autre élément arbitraire. On est ainsi face à un problème d'approximation. Dans le cadre de cette thèse, nous nous intéressons à de tels problèmes ayant pour cadre des espaces de matrices.

Dans [74], HIGHAM propose la définition suivante pour un problème d'approximation (matricielle) (*matrix nearness problem*, en anglais) :

Définition 1.1.1 *Soit E un espace (de matrices) muni d'une norme $\|\cdot\|$.*
Soit H une partie de E constituée d'éléments ayant certaines propriétés particulières.
Considérons pour un vecteur a quelconque de E la quantité suivante :

$$d(a) = \min\{\|e\| \ : \ e + a \in H\}.$$

On appelle problème d'approximation (matricielle) celui consistant en les questions suivantes :

1. *Peut-on déterminer une formule explicite ou une caractérisation "pratique" de $d(a)$?*
2. *Peut-on déterminer $x = a + e_{min}$ où e_{min} est* **un** *vecteur pour lequel le minimum dans $d(a)$ est atteint ? Ce vecteur x est-il unique ?*
3. *Peut-on développer des algorithmes efficaces pour calculer ou estimer $d(a)$ et x ?*

Résoudre un problème d'approximation (matricielle) consiste donc à répondre aux trois questions précédentes.

L'espace E (sous-entendu matriciel dans le reste de cette thèse) et la partie H dans la définition 1.1.1 sont considérés arbitrairement. Selon qu'ils ont en plus certaines propriétés ou qu'ils sont particuliers, on peut résoudre (au moins partiellement) les problèmes induits.

Par exemple, lorsque l'espace E est $\mathbb{R}^n$, muni de la norme euclidienne, et que la partie H s'avère être un polytope, par exemple de la forme

$$\{(x_1, \ldots, x_n) \in \mathbb{R}^n \mid \sum_{j=1}^{n} a_{ij}x_j \leq b_i,\ i = 1, \ldots, m, \quad x_i \geq 0, \forall i = 1, \ldots, n\}$$

on est tout simplement face à un problème de *moindres carrés*. Ce genre de problèmes apparaît dans de nombreux domaines, notamment en Statistiques et en Sciences expérimentales où ils portent le nom de problèmes de régression.

Plus généralement, lorsque E est un espace de Hilbert muni de sa norme induite, et que le sous-ensemble H est convexe et fermé, on est en présence d'un problème dit **de projection**. Nous reviendrons sur ces problèmes au prochain chapitre.

De tout temps, les problèmes d'approximation ont fait l'objet de beaucoup d'attention en Mathématiques. Il en a résulté une abondante littérature sur le domaine. Cela s'explique par le fait que, quelle que soit la théorie à laquelle on s'intéresse, on peut être amené à chercher une approximation d'une quantité à laquelle on ne peut avoir accès directement. Toutefois, les problèmes d'approximation portant sur des matrices ont longtemps été laissés de côté. Ceci peut s'expliquer entre autres par le fait qu'ils nécessitent un gros investissement numérique (notamment en terme de mémoire : stockage d'objets de taille n^2 pour des problèmes de taille n), et surtout par le fait qu'on n'a pas su pendant longtemps traiter les contraintes particulières aux matrices comme, par exemple, les contraintes portant sur les valeurs propres, sur le rang de matrices, etc.

Depuis quelques années, les problèmes d'approximation matricielle ont connu un regain d'intérêt. Cela est dû au développement des moyens informatiques qui ont permis de repousser grandement les limites en termes de stockage mémoire et de mettre en œuvre des logiciels permettant de traiter "globalement" les matrices (sans les transformer en "longs" vecteurs). Une raison plus fondamentale de cet essor est que l'on a appris, ces dernières années, à traiter de manière efficace les contraintes portant sur les valeurs propres et les rangs de matrices, comme par exemple avec la mise au point d'algorithmes de points intérieurs pour les problèmes présentant des contraintes de type semi-définie positivité de matrices. Ainsi, il existe de nombreux travaux sur les problèmes d'approximation matricielle que l'on appelle aussi problèmes de **complétion matricielle**. En Analyse numérique par exemple (voir [74], [73]), on sait que les méthodes itératives de résolution de systèmes linéaires nécessitent que les matrices de ces sytèmes soient définies positives. Lorsqu'une telle matrice est obtenue au moyen d'une boîte noire (c'est à dire que la matrice est obtenue d'une manière opaque pour l'optimiseur), il arrive que la matrice n'ait pas la propriété de définie positivité. On remédie à cela en la remplaçant par exemple par la matrice définie positive la plus proche d'elle au sens d'une norme à préciser. De même, en Chimie moléculaire, on est amené à chercher la bonne configuration spatiale pour une molécule pour laquelle on connaît toutes ou une partie des distances interatomiques. Ce problème peut, par exemple, être modélisé comme un problème d'approximation par des *matrices distances euclidiennes* où on se ramène à compléter (d'où la terminologie *problèmes de complétion*) une matrice dont on ne

connaît pas toutes les composantes de manière à ce que le résultat obtenu ait certaines propriétés. Ce type de problèmes de complétion a été étudié par de nombreux auteurs : on pourra se referer à LAURENT [85], ALFAKIH et WOLKOWICZ [2] et aux articles qui y sont cités.

Il existe d'innombrables autres domaines dans lesquels apparaissent les problèmes d'approximation matricielle. Nous pouvons citer entre autres le Traitement de signal (voir [34], [35], [36], [60], [62], [86]), la théorie des Equations aux Dérivées Partielles (voir [15]), les Statistiques (voir [15]), les Mathématiques financières [88], etc.

Devant la multiplicité des situations où on a des problèmes d'approximation matricielle, nous avons dû faire des choix. Nous nous intéressons aux problèmes pour lesquels :

Hypothèse 1.1.1 (Hypothèses de travail)

- *E est muni d'une structure d'espace de Hilbert,*
- *le convexe $\mathcal{C}$ peut s'écrire comme une* **intersection** *d'un sous-espace affine et d'un cône convexe fermé de E.*

Le convexe $\mathcal{C}$ peut être illustré par la figure 1.1.

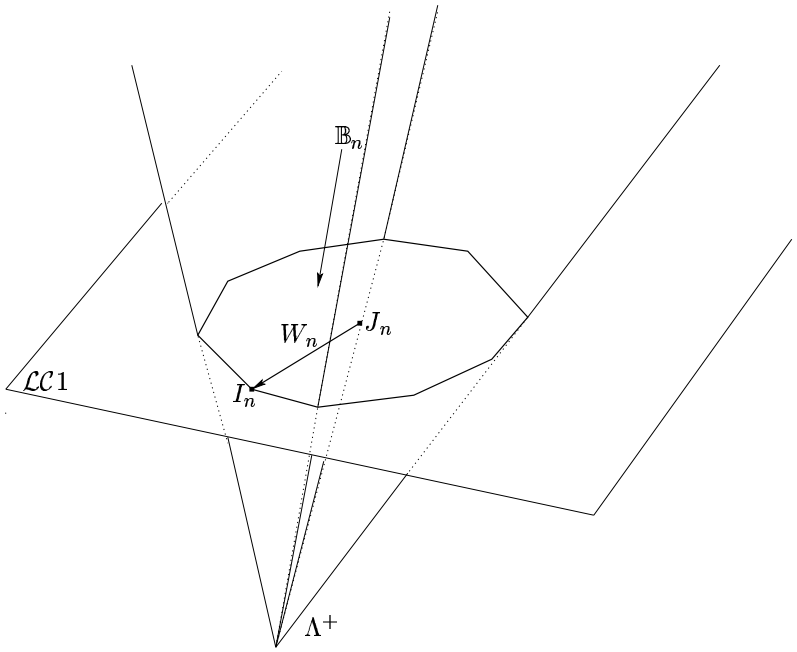


FIG. 1.1 – Ensemble réalisable en approximation linéaire conique

Nous appellerons problèmes d'approximation "linéaires coniques" les problèmes d'approximation vérifiant notre hypothèse de travail. En pratique, l'espace de Hilbert que nous considérerons sera celui des matrices carrées réelles $\mathcal{M}_n(\mathbb{R})$ d'ordre n ($n \in \mathbb{N}^*$) ou celui se restreignant aux matrices symétriques, noté $\mathcal{S}_n(\mathbb{R})$. En ce qui concerne le cône, ce sera celui des matrices à composantes positives ou

celui des matrices symétriques semi-définies positives.

Dans toute la suite, sauf indication contraire, nous nous placerons toujours dans un espace de Hilbert matriciel $(\mathbb{H}, \langle\cdot,\cdot\rangle)$ dont la norme associée est $\|\cdot\|$. Rappelons que lorsqu'un espace de Hilbert est de dimension finie, il est aussi appelé **espace euclidien**. Lorsque ce sera le cas, nous utiliserons indifféremment ces deux terminologies.

1.1.2 Notations

Avant d'aller plus loin, précisons les notations que nous utilisons.

1. **Ensembles**

   Nous notons :
   - $\mathbb{R}^n$ l'espace euclidien des n-uplets $(x_1, \ldots, x_n)$ de réels,
   - $\mathcal{M}_{n\times m}(\mathbb{R})$ ou $\mathbb{R}^{n\times m}$ l'espace des matrices réelles à n lignes et m colonnes,
   - $\mathcal{M}_{n\times n}(\mathbb{R}) = \mathcal{M}_n(\mathbb{R}) \equiv \mathbb{R}^{n^2}$,
   - $\mathcal{S}_n$ ou $\mathcal{S}_n(\mathbb{R})$ l'espace des matrices carrées symétriques d'ordre n,
   - $\mathcal{S}_n^+$ (resp. $\mathcal{S}_n^-$) le cône convexe des matrices symétriques semi-définies positives (resp. négatives),
   - $\mathcal{S}_n^{++}$ (respectivement $\mathcal{S}_n^{--}$) le cône des matrices symétriques définies positives (respectivement négatives).
   - étant donné un sous-espace V d'un espace de Hilbert $(\mathbb{H}, \langle,\rangle)$, nous notons $V^\perp$ son sous-espace orthgonal défini par

   $$V^\perp = \{x \in \mathbb{H} \,|\, \langle x, v\rangle = 0,\ \forall v \in V\}.$$

2. **Vecteurs**

   Les vecteurs sont désignés par des lettres minuscules. Si x est un vecteur de $\mathbb{R}^n$, on désigne par :
   - x^T le vecteur transposé du vecteur x,
   - x_i la $i$ème composante du vecteur x,
   - x^k le $k$ème vecteur d'une suite de vecteurs,
   - $\langle x, y\rangle = x^T y$ le produit scalaire canonique de deux vecteurs,

   $$\langle x, y\rangle = \sum_{k=1}^{n} x_k y_k,$$

   - e_i le $i$ème vecteur de base de $\mathbb{R}^n$,
   - 1_n ou $e \in \mathbb{R}^n$ le vecteur dont toutes les composantes sont égales à 1.

3. **Matrices**

   Les matrices sont désignées par des lettres majuscules. Si A est une matrice, on désigne par :
   - A^T la matrice transposée de la matrice A,
   - A_{ij} la composante située sur la $i$ème ligne et la $j$ème colonne de la matrice A,
   - A^k la $k$ème matrice d'une suite de matrices,

- E_{ij} la (i,j)ème matrice de base de $\mathcal{M}_{n\times m}(\mathbb{R})$,
- $I_n = Diag(1_n)$ la matrice identité. Notons que $Diag$ est l'opérateur qui, à $x \in \mathbb{R}^n$, associe la matrice diagonale D telle que $D_{ii} = x_i$.

4. **Opérations**
   - $\geq$ la relation d'ordre partiel portant sur les vecteurs (respectivement matrices) à composantes positives : $A \geq B \Leftrightarrow A - B$ est à composantes positives.
   - $\succeq$ la relation d'ordre partiel de Löwner portant sur les matrices semi-définies positives : $A \succeq B \Leftrightarrow A - B$ est semi-définie positive.
   - $\succ$ la relation d'ordre partiel (strict) de Löwner portant sur les matrices semi-définies positives : $A \succ B \Leftrightarrow A - B$ est définie positive.
   - $\otimes$ le produit de Kronecker,

$$A \otimes B = \begin{pmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & & \vdots \\ a_{n1}B & \dots & a_{nn}B \end{pmatrix}.$$

   - $\circ$ le produit de Hadamard :

$$A \circ B = C \text{ tel que } C_{ij} = A_{ij}B_{ij},$$

   - $tr(A)$ la trace de la matrice A, c'est-à-dire la somme de tous le termes diagonaux de A : $tr(A) = \sum_{i=1}^{n} A_{ii}$,
   - $\langle\langle A, B\rangle\rangle = tr(A^T B)$ le produit scalaire de Fröbenius sur l'espace $\mathcal{M}_{n\times m}(\mathbb{R})$:

$$\langle\langle A, B\rangle\rangle = \sum_{i=1}^{n}\sum_{j=1}^{m} A_{ij}B_{ij},$$

5. Si $\mathcal{A} : (\mathbb{H}, \langle,\rangle) \to \mathbb{R}^m$ est un opérateur sur un ensemble de matrices, $\mathcal{A}^*$ désigne son opérateur adjoint défini par :

$$\forall X \in \mathbb{H}, \quad \forall y \in \mathbb{R}^m, \quad \langle \mathcal{A}X, y\rangle = \langle\langle X, \mathcal{A}^* y\rangle\rangle.$$

Toute autre notation utilisée dans cette thèse qui n'aurait pas été précisée ci-dessus sera comprise au sens usuel.

1.2 Motivations et exemples

La motivation première de notre étude des problèmes d'approximation est classique dans ce genre de situation. Imaginons, comme cela arrive dans de nombreux domaines, que l'on souhaite disposer d'une matrice X dont on sait qu'elle possède une certaine propriété. Pour différentes raisons, dues par exemple à la manière dont la matrice X est obtenue (erreurs dues aux calculs, données manquantes, etc.), on dispose en réalité d'une matrice A qui n'a pas la proriété voulue. Une des manières, intuitive, de remédier à cette situation consiste à *remplacer la matrice A par une matrice $\widetilde{X}$ ayant la propriété voulue et qui soit la plus proche, dans un certain sens, de A.*

De manière duale, on peut, au contraire, avoir des applications dans lesquelles il est important qu'une certaine matrice A *n'ait pas* une certaine propriété $\mathcal{P}$. On peut chercher alors à estimer l'écart qui sépare A des matrices ayant la propriété $\mathcal{P}$. C'est exactement la quantité que nous avons désigné par $d(A)$ dans la définition 1.1.1.

D'autre part, certains problèmes d'approximation peuvent aussi provenir directement de la modélisation de problèmes provenant de la pratique. Il en est ainsi par exemple du problème d'aggrégation de préférences que nous évoquerons au chapitre 3 et pour lequel nous proposons une modélisation matricielle qui conduit à résoudre un problème d'approximation matricielle. Ce problème se pose en Recherche Opérationnelle, plus précisément en théorie des choix collectifs et du choix social.

Dans les deux prochaines sections (section 1.2.1 et 1.2.2), nous présentons deux problèmes d'approximation matriciels que nous nous attacherons à résoudre entièrement.

1.2.1 Approximation par matrices bistochastiques

Nous nous intéresserons dans un premier temps aux matrices dites **bistochastiques**.

Définition 1.2.1 *On appelle* **matrice bistochastique** *toute matrice réelle dont toutes les composantes sont positives, et dont les lignes et les colonnes ont la particularité d'avoir la somme de leurs composantes qui vaut* 1.

La notion de matrice bistochastique est très connue dans la communauté mathématique, parce qu'elle apparait naturellement en théorie des Probabilités, plus précisément dans l'étude des chaînes de Markov sur un nombre fini d'états.

En dehors de la théorie des Probabilités, on retrouve les matrices bistochastiques dans différents domaines : Recherche opérationnelle [117], Analyse matricielle (théorie de la majorisation) [90], etc.

Dans le prochain chapitre nous nous attacherons à résoudre le problème d'approximation par ces matrices bistochastiques, puis nous présenterons un problème provenant de la théorie du choix social, dans lequel ce problème d'approximation apparaît naturellement.

1.2.2 Approximation par matrices de corrélation

Ensuite, nous nous intéresserons aux matrices dites de corrélation.

Définition 1.2.2 *On appelle* **matrice de corrélation** *toute matrice réelle symétrique semi-définie positive dont tous les termes diagonaux sont égaux à* 1.

Ce genre de matrices apparaît dans différents domaines, notamment en Théorie du contrôle optimal (approximation des équations aux dérivées partielles par "Proper Orthogonal Decomposition" (POD)) où elles portent aussi le nom de matrice de masses), en Statistiques et en Finance comme nous l'expliciterons au chapitre 5.

1.3 Quelques rappels d'Analyse convexe

Nous rappelons quelques résultats d'Analyse convexe dans le cadre d'un espace de Hilbert.

Définition 1.3.1 *Une partie $\mathcal{C}$ de $\mathbb{H}$ est dite convexe si :*

$$\forall t \in]0,1[, \quad \forall x, y \in \mathcal{C}, \quad (1-t)x + ty \in \mathcal{C}.$$

Une fonction $f : \mathbb{H} \to \mathbb{R} \cup \{+\infty\}$ est dite convexe si :

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

Nous ferons appel au cours de nos travaux à différentes notions d'Analyse.

Définition 1.3.2 (Points extrêmes) *Soit $\mathcal{C}$ un ensemble convexe.*

Un point x de $\mathcal{C}$ est un point extrême ou extrémal (ou sommet) de $\mathcal{C}$ si et seulement si il ne peut pas s'écrire comme une combinaison convexe $x = \frac{1}{2}(x_1 + x_2)$ d'éléments différents x_1 et x_2 de $\mathcal{C}$.

On rappelle qu'une partie $\mathcal{K}$ de $\mathbb{H}$ est un *cône* si $\forall x \in \mathcal{K}, \ \forall t \in \mathbb{R}^+, \quad tx \in \mathcal{K}$.

Définition 1.3.3 (Cône polaire) *Soit $\mathcal{K}$ un cône convexe.*

On appelle cône polaire de $\mathcal{K}$, et on note $\mathcal{K}^\circ$, l'ensemble

$$\mathcal{K}^\circ = \{s \in \mathbb{H} \mid \langle s, x \rangle \leq 0 \ \ \forall x \in \mathcal{K}\}$$

Définition 1.3.4 (Cône normal) *Soit $\mathcal{C}$ un ensemble convexe.*

On appelle cône normal à $\mathcal{C}$ en un point x de $\mathcal{C}$, noté $\mathcal{N}(x, \mathcal{C})$, l'ensemble des directions d de $\mathbb{H}$ telle que

$$\langle d, y - x \rangle \leq 0 \quad \forall y \in \mathcal{C}.$$

Notons que lorsque $\mathcal{C}$ est un sous-espace, le cône normal en tout point à $\mathcal{C}$ coincide avec son orthogonal $\mathcal{C}^\perp$.

Proposition 1.3.1 *Soit $\mathcal{K}$ un cône convexe fermé. Alors*

$$\mathcal{N}(x, \mathcal{K}) = \begin{cases} \mathcal{K}^\circ & \text{si} \quad x = 0, \\ \{s \in \mathcal{K}^\circ \mid \langle s, x \rangle = 0\} & \text{si} \quad x \neq 0. \end{cases}$$

Définition 1.3.5 (cône du second ordre) *On appelle* **cône du second ordre** *ou* **cône de Lorentz** *ou encore* **cône quadratique**, *le cône de $\mathbb{R}^{n+1}$ défini par :*

$$\{(x_0, x) \in \mathbb{R}^{n+1} \mid \|x\|_{\mathbb{R}^n} \leq x_0\}.$$

Définition 1.3.6 (sous-différentiel) *Soit $f : \mathbb{H} \to \mathbb{R} \cup \{+\infty\}$ une fonction convexe.*

On dit que $d \in \mathbb{H}$ est un sous-gradient de f au point a si on a :

$$f(y) \geq f(a) + \langle d, y - a \rangle \quad \forall y \in \mathbb{H}.$$

L'ensemble des sous-gradients d'une fonction f en un point a est noté $\partial f(a)$ et s'appelle le sous-différentiel (au sens de l'Analyse convexe) de f au point a.

Rappelons que pour une partie A de $\mathbb{H}$, on définit la fonction suivante :

$$i_A : x \mapsto \begin{cases} 0 & \text{si } x \in A, \\ +\infty & \text{sinon.} \end{cases}$$

Elle est appelé fonction indicatrice de A.

Proposition 1.3.2 *Soit $\mathcal{C}$ un ensemble convexe.*

$$\partial i_{\mathcal{C}}(x) = \mathcal{N}(x, \mathcal{C}), \quad \forall x \in \mathcal{C}.$$

Pour toute autre notion d'Analyse convexe qui n'aurait pas été précisée ci-dessus, on pourra se referer à [77].

1.4 Approches théoriques de résolution

1.4.1 Formulations pratiques du problème.

Nous précisons dans un premier temps les différentes formes sous lesquelles nous présenterons et utiliserons les problèmes d'approximation "linéaire conique".

Définition 1.4.1 *Nous appelons donc problème d'approximation linéaire conique le problème suivant : trouver $\overline{X}$ tel que :*

$$\begin{array}{rcll} \frac{1}{2}\|A - \overline{X}\|^2 & = & \min & \frac{1}{2}\|A - X\|^2 \\ & & \textit{tq.} & X \in \mathcal{S} \\ & & & X \in \mathcal{K} \end{array} \tag{1.1}$$

où $\mathcal{S}$ et $\mathcal{K}$ désignent respectivement un sous-espace affine et un cône convexe fermé de l'espace de Hilbert (matriciel) $\mathbb{H}$.

Remarquons qu'un sous-espace affine $\mathcal{S}$ de $\mathbb{H}$ peut être décrit sous la forme

$$\mathcal{S} = \{X \in \mathbb{H} \,|\, \mathcal{A}X = b, \quad b \in \mathbb{R}^m\}$$

où $\mathcal{A} : \mathbb{H} \to \mathbb{R}^m$ est un opérateur linéaire défini par :

$$\mathcal{A}X = (\langle A_i, X \rangle)_{i=1,\ldots,m}$$

avec A_i matrices données de $\mathbb{H}$.

D'autre part, étant donné un cône convexe fermé $\mathcal{K}$, nous pouvons introduire la relation d'ordre $\succeq_{\mathcal{K}}$ suivante :

Définition 1.4.2

$$\forall A, B \in \mathbb{H}, \qquad A \succeq_{\mathcal{K}} B \Leftrightarrow A - B \in \mathcal{K}.$$

La relation d'ordre $\succeq_{\mathcal{K}}$ ci-dessus généralise les relations d'ordre $\geq$ et $\succeq$ précédemment définies : il suffit de prendre respectivement

$$\mathcal{K} = \{M \in \mathbb{H} \,|\, M = (a_{ij}) \text{ avec } a_{ij} \geq 0\},$$

et

$$\mathcal{K} = \mathcal{S}_n^+ .$$

Compte tenu de la définition 1.4.2 ci-dessus et de la remarque précédente, on a alors la formulation équivalente suivante pour un problème d'approximation linéaire conique :

Proposition 1.4.1 *Le problème (1.1) peut s'écrire sous la forme équivalente suivante : trouver* $\overline{X}$

$$\begin{array}{rlll} & \frac{1}{2}\|A-\overline{X}\|^2 & = & \frac{1}{2}\min \quad \|A-X\|^2 \\ telque: & & & tq. \quad \mathcal{A}X = b \\ & & & \quad\;\; X \succeq_{\mathcal{K}} 0 \end{array} \tag{1.2}$$

La contrainte $X \succeq_{\mathcal{K}} 0$ peut être remplacée par $i_{\mathcal{K}}(X) = 0$.

1.4.2 Existence et caractérisation des solutions

Avant d'aller plus loin, assurons nous que notre problème d'approximation matricielle a un sens et n'est pas trivial. Pour cela, nous faisons la première hypothèse suivante :

Hypothèse 1.4.1 *Il existe des solutions réalisables.*

Cette hypothèse est équivalente à
- $\mathcal{S} \cap \mathcal{K} \neq \emptyset$, pour le problème (1.1).
- $(Ker(\mathcal{A}) + X_0) \cap \mathcal{K} \neq \emptyset$ pour le problème (1.2) où X_0 est un point particulier tel que $\mathcal{A}X_0 = b$.

Nous allons considérer dans la suite de cette partie la formulation (1.2) du problème. Nous sommes en présence d'un problème de minimisation d'une fonction quadratique convexe différentiable sous des contraintes affines et coniques convexes. Différents résultats permettent de répondre à la question de l'existence de solutions optimales au problème et de leur caractérisation. Ainsi par exemple, (voir [77]), considérons un problème de minimisation sous la forme suivante

$$\begin{array}{ll} \min & f(x) \\ \text{tq.} & \mathcal{A}x = b \\ & c_j(x) \leq 0, \quad \forall j = 1, \ldots, p, \end{array} \tag{1.3}$$

où f,$c_j,\ j = 1, \ldots, p$ sont des fonctions convexes.

On a alors :

Théorème 1.4.2 (Karush-Kuhn-Tucker [77], [100]) *Sous réserve de qualification de contraintes, les proposition suivantes sont équivalentes :*

(i) $\bar{x}$ *est un minimiseur du problème* (1.2)

(ii) Il existe $\lambda = (\lambda_1, \ldots, \lambda_m) \in \mathbb{R}^m$ *et* $\mu = (\mu_1, \ldots, \mu_p) \in \mathbb{R}^p$ *tels que*

$$0 \in \partial f(\bar{x}) + \mathcal{A}^*\lambda + \sum_{j=1}^{p} \mu_j \partial c_j(\bar{x}) \tag{1.4}$$

avec $\mu_j \geq 0$ *et* $\mu_j c_j(\bar{x}) = 0 \ \ \forall j = 1, \ldots, p.$

□

Ce théorème est un des principaux résultats sur les conditions d'optimalité pour un problème d'optimisation convexe sous contraintes convexes. On peut se référer à [77], [100] pour de plus amples détails.

Nous supposons dans toute la suite que l'opérateur $\mathcal{A}$ et le cône $\mathcal{K}$ sont tels que :

Hypothèse 1.4.2 (Slater (fort))

$$\exists X_0 \in \mathbb{H} \mid \mathcal{A}X_0 = b \text{ et } X_0 \succ_{\mathcal{K}} 0.$$

Ceci revient juste à dire que les contraintes de notre problème sont (fortement) qualifiées au sens de Slater. Remarquons que cette hypothèse 1.4.2 est vérifiée pour chacun des problèmes auxquels nous allons nous intéresser. Dans les deux, la matrice identité I_n peut être la matrice X_0. Cette hypothèse 1.4.2 étant vérifiée, nous pouvons donc appliquer le théorème 1.4.2 au problème (1.2).

Théorème 1.4.3 *On suppose l'hypothèse 1.4.2 vérifiée.*

$\overline{X}$ est un minimiseur du problème (1.2) *si et seulement si il existe $\lambda \in \mathbb{R}^m$ tel que*

$$\overline{X} - A + \mathcal{A}^*\lambda \in \mathcal{N}(\mathcal{K}, \overline{X}) \tag{1.5}$$

□

Preuve : Il suffit d'appliquer le théorème 1.4.2 avec :

$$f(X) = \frac{1}{2}\|X - A\|^2, \quad p = 1 \text{ et } c_1(X) = i_{\mathcal{K}}(X).$$

Or, f est différentiable, de gradient $\nabla f(X) = X - A$ pour tout X, puisque nous avons ici une norme hilbertienne.

De plus, d'après la proposition 1.3, $\partial i_{\mathcal{K}}(\overline{X}) = \mathcal{N}(\mathcal{K}, \overline{X})$.

On en déduit qu'il existe $\lambda \in \mathbb{R}^m$ et $\mu \in \mathbb{R}$ tel que

$$\overline{X} - A + \mathcal{A}^*\lambda \in -\mu\mathcal{N}(\mathcal{C}, \overline{X}).$$

De la condition de complémentarité $\mu \cdot i_{\mathcal{K}}(\overline{X}) = 0$, on déduit $\mu > 0$, puisque $\overline{X} \in \mathcal{K} \Rightarrow i_{\mathcal{K}}(\overline{X}) = 0$. Par suite,

$$\overline{X} - A + \mathcal{A}^*\lambda \in -\mathcal{N}(\mathcal{C}, \overline{X}).$$

puisque $\mathcal{N}(\mathcal{K}, \overline{X})$ est un cône convexe fermé. D'où le Théorème.

■

Nous disposons donc d'une caractérisation des solutions optimales. Une fois assurée l'existence d'une solution optimale se pose la question de son calcul effectif. Cela consisterait à résoudre l'équation multivoque (1.5), ce qui n'est pas évident. Il est possible d'obtenir d'autres caractérisations d'optimalité (plus simple), notamment en passant par le théorème de projection (voir chapitre suivant) et par la dualité lagrangienne (voir chapitre 5). Néanmoins, nous verrons que bien souvent ces caractérisations seront peu pratiques lorsqu'il s'agira de calculer les solutions optimales.

1.4.3 Unicité des solutions

Une fois assurée l'existence d'**une** solution optimale se pose la question du nombre de ces solutions optimales. Dans notre cas, ce nombre est facile à déterminer.

Théorème 1.4.4 *Il existe une unique solution optimale au problème d'approximation linéaire conique.*

□

La justification de ce résultat tient essentiellement au fait que la fonction-objectif du problème est strictement convexe, puisque la carré de la norme $\|\cdot\|$ l'est.

1.5 Approches numériques de résolution

Nous introduisons dans cette partie différentes approches numériques de résolution que nous proposons ou bien dont nous avons pu prendre connaissance dans la littérature. Nous les présentons rapidement, en nous contentant d'en évoquer les lignes directrices. Nous reviendrons sur chacune de ces approches dans les chapitres qui suivent lorsque nous les appliquerons. Rappelons que le problème que nous cherchons à résoudre peut s'écrire sous la forme suivante :

$$\begin{array}{rcll} \frac{1}{2}\|A-\overline{X}\|^2 & = & \frac{1}{2}\min & \|A-X\|^2 \\ & & \text{tq.} & X \in \mathcal{S} \\ & & & X \in \mathcal{K} \end{array} \tag{1.6}$$

où $\mathcal{S}$ et $\mathcal{K}$ désignent respectivement un sous-espace affine et un cône convexe fermé. La contrainte $X \in \mathcal{S}$ sera souvent présentée sous la forme $\mathcal{A}X = b$ où $b \in \mathbb{R}^m$ et $\mathcal{A}$ est un opérateur linéaire sur l'espace $\mathbb{H}$.

1.5.1 Approches directes par moindres carrés

Cette approche est la première à laquelle on songe lorsque l'on est face à un problème d'approximation matricielle dans lequel la norme considérée est la norme de Fröbénius. Elle est basée sur le fait topologique suivant :

l'espace $\mathcal{M}_n(\mathbb{R})$ muni de la norme de Fröbénius $\|\cdot\|_F$ s'identifie immédiatement à l'espace $\mathbb{R}^{n^2}$ muni de la norme $\|\cdot\|_2$.

Compte tenu de cette identification, notre problème d'approximation peut se ramener à un problème de moindres carrés.

L'intérêt de cette transformation est, comme souvent en mathématiques, qu'elle permet de se ramener à un type de problèmes pour lesquels on dispose d'outils de résolution performants. C'est le cas des méthodes de moindres carrés pour la résolution desquels existent des codes, qu'ils soient commerciaux ou du domaine public, et notammant des routines sous Matlab.

On peut cependant déjà préjuger du peu d'efficacité que devrait avoir cette approche dans la pratique. En effet, il peut dans un premier temps être très difficile

de ramener de manière explicite les contraintes matricielles de (3.12) sous la forme des contraintes de type moindres carrés. Un deuxième inconvénient, peut-être le plus important, consiste en ce qu'on se ramène à travailler dans $\mathbb{R}^{n^2}$, ce qui conduit à un problème dont la taille peut se révéler très vite prohibitive. Ceci empêcherait de résoudre le problème d'approximation pour des matrices d'ordre n relativement modeste ($n \approx 50$) au regard des ordres de matrices que l'on est amené à rencontrer dans les cas pratiques ($n \geq 1000$) que l'on voudrait résoudre.

Face à ce constat, il apparaît nécessaire, si l'on veut résoudre ces problèmes d'approximation de manière optimale, de conserver autant que possible la structure matricielle des variables du problème. De plus, il faudra penser à utiliser au mieux la (les) structure(s) propre(s) au problème. Nous présentons dans cette thèse les quatre autres approches énumérées ci-dessous. Les deux premières sont présentées de manière assez rapide pour des raisons différentes. L'approche duale n'est pas de notre fait, mais au regard de son efficacité et de la nouveauté, à notre connaissance, de la démarche et de certains résultats, nous avons pensé intéressant de la présenter. Ce choix est aussi dicté par le fait qu'elle inspire l'approche par points fixes. En ce qui concerne celle-ci, les travaux étant encore à leurs débuts, nous nous contentons d'en montrer les principes et une illustration.

1.5.2 Approche duale par Quasi-Newton

Cette approche est due à J. MALICK [88]. Elle peut être décrite comme suit : tout d'abord, on applique un procédé de relaxation lagrangienne au problème au cours duquel seules les contraintes linéaires sont dualisées. Cela permet de récupérer un problème dual de maximisation qui est concave et, contrairement à l'habitude, **différentiable**. Ce dernier résultat, nouveau, est très important puisqu'il est le nœud central de cette approche numérique. En effet, compte tenu de cette différentiabilité, le problème dual peut être résolu de manière efficace en utilisant une méthode numérique de minimisation convexe de type quasi-Newton.

1.5.3 Approche par points fixes

Cette approche découle directement de la précédente et fait appel à des notions d'opérateurs non expansifs (contractants) et de points fixes. La condition d'optimalité obtenue par la dualisation précédente est réexprimée à l'aide d'opérateurs. Moyennant une hypothèse sur l'opérateur linéaire $\mathcal{A}$ qui définit le sous-espace affine $\mathcal{S}$ qui se vérifie facilement, la condition d'optimalité devient alors une condition d'existence de points fixes d'un opérateur contractant. Cette approche donnant actuellement lieu à des travaux (voir [22]), nous ne nous appesantirons pas sur elle.

1.5.4 Approche par projections alternées

L'approche par projections alternées est une approche directe de résolution. Elle peut être vue comme une manière naturelle d'aborder le problème. Sous nos hypothèses, celui-ci peut être vu comme un problème de projection sur l'intersection de deux convexes. L'approche par projections alternées peut être décrite comme

suit : on cherche à effectuer une projection sur un convexe qui est l'intersection de convexes plus "simples" sur lesquels on sait justement effectuer des projections ; la meilleure solution consiste à utiliser ces projections connues pour construire itérativement la projection que nous cherchons.

1.5.5 Approche par points intérieurs

Cette approche par points intérieurs est motivée par la contrainte conique présente dans notre problème. En effet, compte tenu de cette contrainte, le problème peut être écrit sous la forme d'un problème mixte d'optimisation sur le cône du second ordre (Définition 1.3.5) et, selon les exemples, sur le cône des matrices à composantes positives ou symétriques semi-définie positives. Ceci nous permettra de résoudre, au chapitre 5, le problème en utilisant les méthodes de points intérieurs, méthodes qui ont connu un regain d'intérêt ces dix dernières années, en grande partie à cause justement de leur remarquable efficacité dans la résolution de problèmes d'optimisation sous contraintes de semi-définie positivité.

Chapitre 2

Algorithmes de projections

Certaines des approches de résolution que nous aurons à mettre en œuvre et à présenter dans cette thèse sont intimement liées à la notion de projection dans un espace de Hilbert $\mathbb{H}$. Nous rappelons donc dans un premier temps quelques résultats, propriétés et algorithmes liés aux opérateurs de projections.

Dans tout ce chapitre, sauf indication contraire, nous nous placerons toujours dans le cadre d'un espace de Hilbert $\mathbb{H}$ muni du produit scalaire $\langle \cdot, \cdot \rangle$. Nous noterons $\| \cdot \|$ la norme associée à ce produit scalaire.

2.1 Notions de projections

Pour présenter la notion de projection dans un espace de Hilbert, on peut se placer du point de vue de l'Analyse hilbertienne ou de celui de l'Optimisation convexe. Nous associerons ces deux points de vue.

Etant donné un point x et un convexe fermé C non vide de $\mathbb{H}$, on montre :

Théorème 2.1.1 (Théorème de projection [29], [77],[100]) *Considérons une partie C convexe fermée non vide de $\mathbb{H}$.*

Pour tout point x de $\mathbb{H}$, il existe un et un seul point $\overline{c}$ de C tel que :

$$\|x - \overline{c}\| = \inf\{\|x - c\|, \quad c \in C\}. \tag{2.1}$$

De plus, $\overline{c}$ est caractérisé par :

$$\begin{cases} \overline{c} \in C, \\ \langle x - \overline{c}, c - \overline{c} \rangle \leq 0 \quad \forall c \in C. \end{cases} \tag{2.2}$$

□

Ce théorème se prouve, soit en utilisant des outils d'Analyse hilbertienne, notamment les propriétés du produit scalaire et celles des espaces réflexifs (voir [29]), soit, comme décrit ci-après, au moyen de l'Optimisation convexe : si nous introduisons la fonction indicatrice i_C de l'ensemble C, le problème (2.1) est équivalent à :

$$\frac{1}{2}\|x - \overline{c}\|^2 = \inf\{h(c) = \frac{1}{2}\|x - c\|^2 + i_C(c), \quad c \in \mathbb{H}\},$$

qui est un problème de minimisation convexe sans contraintes. Sa solution optimale $\overline{c}$ est donc caractérisée par la condition de stationnarité :

$$0 \in \partial h(\overline{c}), \tag{2.3}$$

où $\partial h(\overline{c})$ désigne le sous-différentiel de h au sens de l'Analyse convexe (voir définition 1.3.6). La caractérisation (2.2) découle par des règles de calcul sous-différentiel de l'inclusion (2.3) ci-dessus.

Le point $\overline{c}$ ci-dessus est appelé *projeté* de x sur l'ensemble C, d'où le nom du théorème. Il existe un corollaire très utile de ce théorème.

Corollaire 2.1.1 *Si, de plus, C est un sous-espace fermé de $\mathbb{H}$, alors la caractérisation (2.2) devient*

$$\left\{ \begin{array}{c} \overline{c} \in C, \\ x - \overline{c} \in C^{\perp}. \end{array} \right. \tag{2.4}$$

□

En pratique, lorsque C est un sous-espace **vectoriel**, la caractérisation utilisée est :

$$\left\{ \begin{array}{l} \overline{c} \in C, \\ \langle x - \overline{c}, c \rangle = 0, \quad \forall c \in C, \end{array} \right. \tag{2.5}$$

tandis que lorsque c'est un sous-espace **affine**, on a :

$$\left\{ \begin{array}{l} \overline{c} \in C, \\ \langle x - \overline{c}, c \rangle = cte, \quad \forall c \in C. \end{array} \right. \tag{2.6}$$

Pour un élément x de $\mathbb{H}$, on note $\overline{c} = \mathcal{P}_C(x)$ ou $\mathcal{P}_C x$, où $\overline{c}$ est le projeté défini dans le théorème (et le corollaire) précédent. Ceci nous définit au passage un opérateur

$$\begin{array}{rccc} P_C : & \mathbb{H} & \to & \mathbb{H} \\ & x & \mapsto & \mathcal{P}_C(x) \end{array}$$

que nous appellerons opérateur de projection sur l'ensemble C. On peut montrer les résultats suivants :

Proposition 2.1.2 *Pour tous x, y dans $\mathbb{H}$, pour tout convexe C de $\mathbb{H}$,*

$$\begin{array}{rcl} \|x - y\|^2 & = & \|\mathcal{P}_C x - \mathcal{P}_C y\|^2 + \|(x - y) - (\mathcal{P}_C x - \mathcal{P}_C y)\|^2 \\ & & +2\langle x - \mathcal{P}_C x, \mathcal{P}_C x - \mathcal{P}_C y \rangle + 2\langle y - \mathcal{P}_C y, \mathcal{P}_C y - \mathcal{P}_C x \rangle. \end{array} \tag{2.7}$$

□

Démonstration :

L'égalité précédente vient du développement suivant :

$$\|a + b\|^2 = \|a\|^2 + \|b\|^2 + 2\langle a, b \rangle,$$

classique en Analyse hilbertienne. Il suffit d'écrire

$$x - y = [(x - y) - (\mathcal{P}_C x - \mathcal{P}_C y)] + (\mathcal{P}_C x - \mathcal{P}_C y)$$

et de poser

$$a = (x - y) - (\mathcal{P}_C x - \mathcal{P}_C y) \text{ et } b = \mathcal{P}_C x - \mathcal{P}_C y.$$

■

Corollaire 2.1.2 *Pour tous x, y dans $\mathbb{H}$, on a :*

$$\|\mathcal{P}_C x - \mathcal{P}_C y\| \leq \|x - y\|. \tag{2.8}$$

□

Démonstration :
Ce résultat vient de la proposition 2.1.2 précédente. Il suffit de remarquer que d'après (2.2), on a :

$$\langle x - \mathcal{P}_C x, \mathcal{P}_C y - \mathcal{P}_C x\rangle \leq 0 \text{ et } \langle y - \mathcal{P}_C y, \mathcal{P}_C x - \mathcal{P}_C y\rangle \leq 0,$$

car $\mathcal{P}_C x, \mathcal{P}_C y \in C$.

■

Proposition 2.1.3 *Soit C une partie convexe fermée de $\mathbb{H}$.*
(i) *Si $x \in \mathbb{H}$, on a : $x - \mathcal{P}_C(x) \in \mathcal{N}(\mathcal{P}_C(x), C)$.*
(ii) *On suppose que C est un sous-espace vectoriel (resp. affine), alors $\mathcal{P}_C$ est linéaire (resp. affine).*

□

La proposition (i) est juste la traduction de la condition de stationnarité (2.3).
La proposition (ii) découle de la caractéristion (2.5).

Notons au passage que la caractérisation **(i)** de la proposition précédente est équivalente à la caractérisation (1.5) du Théorème 1.4.2 du chapitre 1 pour nos problèmes d'approximation linéaires coniques. En effet, dans ce théorème, on est dans le cas où C est l'intersection d'un cône convexe fermé $\mathcal{K}$ et d'un sous-espace affine défini par la contrainte $\mathcal{A}x = b$. Par une règle de calcul sous-différentiel, si l'hypothèse de Slater 1.4.2 est vérifiée, le cône normal de C est en fait la somme des cônes normaux à $\mathcal{K}$ et au sous-espace affine. Il suffit alors de remarquer que le cône normal à un sous-espace affine s'identifie à l'orthogonal de sa direction, qui est exactement égal ici à l'image de l'opérateur adjoint $\mathcal{A}^*$ de $\mathcal{A}$, pour obtenir (1.5) à partir de **(i)**.

Une fois connues ces différentes propriétés de l'opérateur $\mathcal{P}_C$, se pose la question du calcul effectif du projeté $\mathcal{P}_C(x)$ d'un point x donné. Comme nous allons le voir tout au long de cette thèse, cette question est loin d'être anodine. Toutefois, dans quelques cas particuliers, les caractérisations (2.2), (2.5) ou (2.6) permettent de connaître explicitement $\mathcal{P}_C(x)$. On peut par exemple montrer :

Proposition 2.1.4 *Dans l'espace euclidien $\mathbb{R}^n$, notons $\Lambda = \{x \in \mathbb{R}^n \mid x_i \geq 0, \forall i\}$. Alors, pour tout $x \in \mathbb{R}^n$,*

$$\mathcal{P}_\Lambda(x) \in \mathbb{R}^n \text{ tel que } (\mathcal{P}_\Lambda(x))_i = x_i^+ = \max\{x_i, 0\},\ \forall i.$$

□

De même, si on introduit la notation suivante : si $A = (a_{ij})$ est une matrice de réels, on note $A^+ = (a_{ij}^+)$ où $a_{ij}^+ = \max\{a_{ij}, 0\}$.

Proposition 2.1.5 *Dans l'espace euclidien $\mathcal{S}_n$, muni du produit scalaire de Frö-benius, on note $\mathcal{S}_n^+$ le cône des matrices semidéfinies positives. Alors, pour toute matrice X, on a :*

$$\mathcal{P}_{\mathcal{S}_n}(X) = U^T D^+ U,$$

où $X = U^T DU$ avec $U^T U = I_n$ et D diagonale. □

On peut montrer des résultats du même type pour des opérateurs de projection sur différents types de sous-ensembles convexes fermés dans un espace de Hilbert : cônes, sous-espaces, polyèdres convexes, épigraphes et sous-niveau de fonctions convexes, etc. On pourra se référer à [15] pour de plus amples détails.

Une des applications des projections est qu'elles permettent de calculer la distance entre un point et un sous-ensemble convexe.

Définition 2.1.1 *Soit A une partie de $\mathbb{H}$ et $x \in \mathbb{H}$.*

On appelle distance de x à A, et on note $d(x, A)$, la quantité suivante :

$$d(x, A) = \inf\{\|x - a\| \mid a \in A\}.$$

Cette quantité $d(x, A)$ est identique à la quantité $d(A)$ de la définition 1.1.1. On peut alors définir une fonction

$$\begin{array}{rccc} d_A : & \mathbb{H} & \to & \mathbb{R} \\ & x & \mapsto & d(x, A) \end{array}$$

que nous appellerons *fonction distance* à A.

Proposition 2.1.6 *Soit C une partie convexe fermée de $\mathbb{H}$.*

1. *d_C est une fonction convexe, finie et vérifie*

$$d_c(x) = \|x - \mathcal{P}_C(x)\|.$$

2. *Pour tout x dans $\mathbb{H}$,*

$$\partial d_C(x) = \begin{cases} \left\{\frac{x - \mathcal{P}_c(x)}{\|x - \mathcal{P}_c(x)\|}\right\} & \text{si } x \notin C \\ B_X \cap \mathcal{N}(x, C) & sinon \end{cases}$$

Résultats classiques d'Analyse convexe ([15], [77]).

Les opérateurs de projection ont fait l'objet d'études nombreuses et variées que nous ne pouvons pas toutes décrire ou évoquer dans cette thèse. Nous renvoyons pour plus de détails aux travaux de BAUSCHKE, notamment sa thèse [15], et de ZARANTONELLO [118]. D'autre part, signalons que la notion classique de projection que nous avons présentée ici a été généralisée : en quasi-projection [15], en projection de Bergman[23], etc.

2.2 Les méthodes de projections

2.2.1 Motivations : problèmes de faisabilité convexe

Soit à résoudre dans $\mathbb{R}^n$ un système d'inéquations linéaires définies par :

$$\sum_{j=1}^{n} a_{ij}x_j \leq b_i, i = 1, \ldots, m.$$

On peut se ramener à chercher un point $x = (x_1, \ldots, x_n)$ qui appartient à tous les demi-espaces définis par

$$E_i = \{x \in \mathbb{R}^n \mid \sum_{j=1}^{n} a_{ij}x_j \leq b_i\}.$$

Le problème consiste alors en fait à chercher **un** point qui appartient à l'intersection d'un nombre fini de demi-espaces. On définit, d'une manière générale, un **problème de faisabilité ou de réalisabilité convexe** (*Convex feasibility problem (CFP)*) comme suit :

On se place dans un espace de Hilbert $\mathbb{H}$ et, dans cet espace, on considère une famille finie ou dénombrable de convexes $\{C_i\}_{i\in I}$ d'intersection non vide. On considère dans $\mathbb{H}$ le problème suivant :

$$(CFP) \text{ Trouver un } x \in C = \cap_{i\in I} C_i.$$

Les convexes C_i évoqués ci-dessus sont supposés "simples" en comparaison avec C. En général, "simple" est compris dans le sens où la projection sur C_i est facilement calculable. Typiquement, C_i sera un sous-espace, un demi-espace, un cône, etc.

Les algorithmes de projection ont d'abord été introduits pour faire face à ce type de problèmes. Une telle approche est par exemple mise en œuvre par POLYAK [99] pour un système d'équations et/ou d'inéquations linéaires dans $\mathbb{R}^n$. Plus généralement, les problèmes de faisabilité apparaissent dans différents domaines :

- en théorie de l'approximation : les convexes sont souvent des sous-espaces et on a des applications en Statistiques, en Analyse complexe (noyaux de Bergman, transformations conformes), dans l'étude des équations aux dérivées partielles, (voir [15]),
- en reconstruction d'images discrète et continue : applications en tomographie, en électronique, en traitement du signal [39], [40], [41], [42], [46],
- en optimisation convexe via les algorithmes de sous-gradients [81], [82], entre autres.

2.2.2 Principes

Dans la suite, nous effectuerons la présentation des méthodes de projection dans le cas où on a deux convexes, c'est-à-dire $I = \{1, 2\}$ et, pour alléger les écritures, nous allons noter

$$C = A \cap B.$$

Nous notons respectivement $\mathcal{P}_C$, $\mathcal{P}_1$ et $\mathcal{P}_2$ les projections sur C, A et B.

L'idée est de construire itérativement la solution de (CFP) de la manière suivante : on part d'un point initial x_0 et,
étant donné l'itéré courant x_n, construire l'itéré suivant x_{n+1} qui doit être "meilleur" que x_n en utilisant les projections calculables $\mathcal{P}_1$ et $\mathcal{P}_2$.

Dans la pratique, il est nécessaire de préciser le sens du mot "meilleur" dans l'énoncé précédent. Il semble raisonnable de demander que le nouvel itéré x_{n+1} nous rapproche plus du convexe C que l'itéré courant. En d'autres termes, une bonne mesure du caractère "meilleur" précédent serait que l'on ait :

$$d(x_{n+1}, C) \leq d(x_n, C).$$

Il en vient la définition suivante :

Définition 2.2.1 *Soit (x_n) une suite de $\mathbb{H}$ et soit C une partie convexe fermée de $\mathbb{H}$.*

On dit que (x_n) est **monotone au sens de Fejér ou Fejér-monotone par rapport à** *C si :*

$$(*) \qquad \forall c \in C,\ \forall n \in \mathbb{N},\ \ \|x_{n+1} - c\| \leq \|x_n - c\|. \tag{2.9}$$

Ainsi, dans l'énoncé précédent, le fait pour x_{n+1} d'être meilleur que x_n peut être exprimé par

$$\forall c \in C = A \cap B,\ \forall n \in \mathbb{N},\ \ \|x_{n+1} - c\| \leq \|x_n - c\|.$$

On se ramène donc à construire itérativement la solution de (CFP) de manière à ce que la suite (x_n) générée soit monotone au sens de Fejér par rapport à C.

Un exemple de schéma de projection conduisant à une suite monotone au sens de Fejér est le suivant :
Etant donné x_n (itéré courant), on calcule :

$$\begin{array}{lcl} x_{n+1} = \mathcal{P}_1 x_n, & \text{si} & x_n \notin A, \\ & \text{ou} & \\ x_{n+1} = \mathcal{P}_2 x_n, & \text{si} & x_n \notin B. \end{array}$$

Ce schéma entre bien dans le cadre que nous avons annoncé, puisque x_{n+1} est construit à partir de x_n en utilisant les projections calculables $\mathcal{P}_i$. De plus, il est facile de voir que (x_n) est monotone au sens de Fejér.

En effet,

$$\forall n,\ \text{on a : } x_{n+1} = \mathcal{P}_i x_n,\ \ i = 1 \text{ ou } 2.$$

et d'autre part, comme $C \subset C_i$, pour tout $i = 1, 2$, pour tout $c \in C,\ \mathcal{P}_i(c) = c$.

Or, d'après le corollaire de la Proposition 2.1.2, on a :

$$\forall x, y,\ \ \|\mathcal{P}_C x - \mathcal{P}_C y\| \leq \|x - y\|.$$

Par suite

$$\|x_{n+1} - c\| \leq \|\mathcal{P}_i x_n - \mathcal{P}_i c\| \leq \|x_n - c\|.$$

Il est facile de voir que ce schéma consiste à projeter alternativement l'itéré courant sur A ou B. De là lui vient le nom de méthode de projections alternées. On la doit à VON NEUMANN [113] (1933). Nous reparlerons de cet algorithme dans la partie suivante.

Plus généralement, BAUSCHKE montre qu'une bonne condition pour que ceci soit réalisé est d'exiger que :

$$x_{n+1} \in x_n + \text{cône}\{\mathcal{P}_A x_n - x_n, \mathcal{P}_B x_n - x_n\}$$

où cône(X) désigne le cône convexe fermé engendré par la partie X de $\mathbb{H}$. Ceci nous induit par exemple une relation de récurrence du type :

$$x_{n+1} = x_n + \rho\,[w_1(\mathcal{P}_A x_n - x_n) + w_2(\mathcal{P}_B x_n - x_n)]$$

où $\rho, w_1, w_2 \geq 0$ et $w_1 + w_2 = 1$. Le réel ρ est un paramètre de relaxation et w_1, w_2 sont des poids vérifiant

$$0 \leq \rho \leq \frac{w_1\|\mathcal{P}_A x_n - x_n\| + w_2\|\mathcal{P}_B x_n - x_n\|}{\|w_1(\mathcal{P}_A x_n - x_n) + w_2(\mathcal{P}_B x_n - x_n)\|}.$$

Signalons enfin que le fait de considérer une suite d'itérés Fejér-monotones a, en outre, l'avantage de mettre à notre disposition un certain nombre de résultats sur les propriétés de la suite générée, notamment des résultats de convergence. L'étude des propriétés des suites monotones au sens de Fejér, constitue une bonne partie de l'Analyse Fejérienne. On pourra se référer à propos de tout ce qui précède aux travaux de BAUSCHKE [15], [19], [20], [21] et COMBETTES [43], [44], [45] notamment. Il existe évidemment des manières différentes et variées d'effectuer la mise à jour :

$$x_n \to x_{n+1}$$

en respectant les règles évoquées. Pour en savoir plus, on peut se référer à [7], [11], [15], [16],[23], [43], [99].

2.3 Méthodes de projection pour l'approximation

Le point commun des méthodes de projection que nous avons évoquées ci-dessus est qu'elles permettent de construire un point de l'intersection $C = A \cap B$ des convexes A et B. On obtient **un** point de C dont on ne peut rien dire d'autre. En particulier, on n'obtient donc pas forcément le point de C le plus proche d'un point $x \in \mathbb{H}$ donné, sauf dans certains cas particuliers, évidemment.

Toutefois, ces dernières années, de nombreuses recherches ont été effectuées qui ont permis d'aboutir à des méthodes de projections permettant de construire itérativement le projeté d'un point quelconque sur l'intersection de convexes fermés non vides. On peut d'une manière générale distinguer deux types de méthodes : les méthodes de projections alternées (ou cycliques) dues à BOYLE et DYKSTRA et les méthodes de projections parallèles relaxées de BAUSCHKE et COMBETTES. Nous avons utilisés dans nos travaux les méthodes de projections alternées que nous présentons ci-après. Nous nous proposons de tester les méthodes de projections parallèles dans des travaux futurs. Signalons que les recherches concernant les méthodes

de projections qui permettent de calculer les projections sur des intersections de convexes sont toujours en cours. On peut ainsi noter les travaux récents de BREGMAN, CENSOR,REICH et ZEPKOWITZ-MALACHI [28]. On trouvera notamment en introduction à cet article une historique des méthodes de projection sur les intersections de convexes avec de nombreuses références bibliographiques.

Le but de cette section est de présenter une méthode de projections alternées qui permet de construire itérativement le point de C le plus proche d'un point x donné. Cette méthode a été introduite par DYKSTRA en 1983 dans le cas particulier où les convexes C_i sont des cônes et où on est en dimension finie. Puis, il l'a étendue avec BOYLE en 1986 au cas général où on a des convexes quelconques dans un espace de Hilbert. Elle a été popularisée notamment par BAUSCHKE et BORWEIN qui en ont explicité les propriétés de convergence (essentiellement dans le cas de deux ensembles), et par GLUNT *et al.* [64], [65], ESCALANTE [54] entre autres qui l'ont appliquée à différents problèmes.

2.3.1 Algorithme de Von Neumann

Nous revenons à la méthode de Von Neumann que nous avons introduite à la section 2.2.2

Algorithme 2.3.1 *On peut la décrire sous la forme suivante :*

$$\begin{array}{l} a_n \in A, b_n \in B, \\ a_{n+1} = \mathcal{P}_A(b_n) \\ b_{n+1} = \mathcal{P}_B(a_{n+1}) \\ \textit{avec } b_0 = x \in \mathbb{H} \textit{ et } a_0 = 0. \end{array} \tag{2.10}$$

Nous avons vu précédemment que cette méthode pouvait permettre de construire un point de l'intersection C. En fait, on montre, voir [17], [113], que lorsque A et B sont des sous-espaces (vectoriels ou affines) fermés et que les suites (a_n) et (b_n) sont définies ci-dessus en (2.10), on a :

$$a_n, b_n \to \mathcal{P}_C(b_0).$$

Remarquons qu'on a :

$$b_{n+1} = \mathcal{P}_B(a_{n+1}) = \mathcal{P}_B \circ \mathcal{P}_A(b_n). \tag{2.11}$$

Ainsi, la méthode de von Neumann peut se ramener à la construction d'une suite unique (b_n) définie comme en (2.11) et qui vérifie donc :

$$b_n \to \mathcal{P}_C(b_0).$$

Ce résultat est facile à visualiser lorsqu'on se situe dans un espace de dimension 2. Ceci est illustré par la figure 2.1.

En conclusion, lorsque les convexes fermés A et B sont des sous-espaces, on sait comment construire itérativement le projeté d'un point quelconque. Historiquement, on peut dire que la méthode de von Neumann a constitué la première solution,

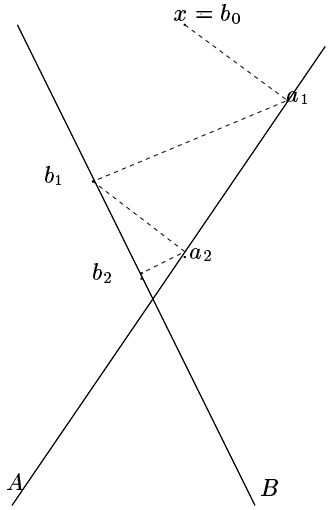


FIG. 2.1 – Illustration de l'algorithme de Von Neumann

mais surtout une des plus efficaces, au problème qui consiste à trouver la projection d'un point donné dans un espace de Hilbert sur l'intersection non vide d'un nombre fini de sous-espaces fermés.

Remarquons qu'on peut réécrire (2.11) sous la forme :

$$b_{n+1} = Tb_n$$

en posant $T = \mathcal{P}_B\mathcal{P}_A$. Ainsi T est un opérateur de $\mathbb{H}$, linéaire (ou affine) dans le cas où A et B sont des sous-espaces (voir section 2). On voit qu'on peut interpréter (b_n) comme étant une suite d'approximations successives par rapport à T. On sait qu'une telle suite, si elle converge, le fait vers un point fixe de T. D'autre part, on peut remarquer que

$$c \in C = A \cap B \Leftrightarrow Tc = \mathcal{P}_B\mathcal{P}_A c = c$$

et (b_n) converge donc vers un point fixe de T. Ceci a induit le fait que la méthode de von Neumann, et les méthodes de projection en général, ont été étendues et adaptées à la recherche d'un point fixe d'un opérateur et surtout à celle d'un point fixe commun à un nombre fini d'opérateurs monotones (voir [14], [15], [45]).

La méthode de von Neumann introduite dans le cas de deux sous-espaces se généralise de manière naturelle au cas d'un nombre fini de sous-espaces : on passe de projections alternées à des projections cycliques. BREGMAN [27] a étendu les résultats de convergence à ce cas.

Que se passe-t-il si on n'a plus les hypothèses de von Neumann, c'est-à-dire si l'un des convexes n'est pas un sous-espace ?

Regardons la figure 2.2 : on cherche le projeté d'un point x sur l'intersection d'un cône A et d'une droite (sous-espace) B.

Il est facile de voir que le projeté sur $A \cap B$ est l'extrémité droite du segment qui représente $C = A \cap B$, tandis que l'algorithme de von Neumann conduit à un

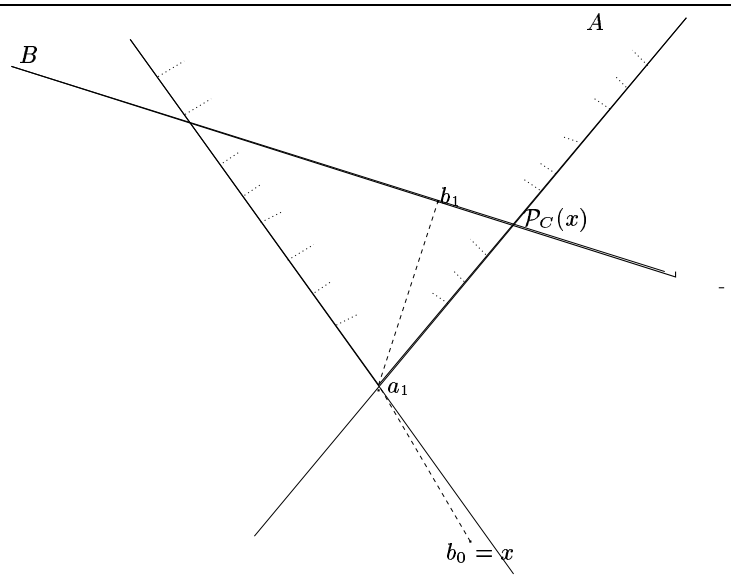


FIG. 2.2 – Von Neumann sur l'intersecton d'un cône et d'un sous-espace

point intérieur au segment.

Il y apparaît bien que si l'un des convexes n'est pas un sous-espace, les conclusions de convergence précédentes ne sont plus assurées. On montre (voir [17], [18]) que dans le cas général, on a toujours convergence au moins faible de l'algorithme de von Neumann ; mais le point limite obtenu est un point *quelconque* de C.

Que faire donc dans le cas général ?

2.3.2 Algorithme de Boyle-Dykstra

Pour répondre à cette question, DYKSTRA a proposé une modification de l'algorithme de von Neumann. Le schéma en est le suivant : on construit quatre suites : (a_n), (b_n) (appelées *suites principales*) et (p_n), (q_n) (appelées *suites auxiliaires*) comme suit :

Algorithme 2.3.2

$$\left\{\begin{array}{l} a_0 = 0; b_0 = x \in \mathbb{H}; p_0 = 0; q_0 = 0; \\ \mathbf{a_{n+1}} = \mathcal{P}_{\mathbf{A}}(\mathbf{b_n} + \mathbf{p_n}) \\ p_{n+1} = b_n + p_n - a_{n+1} \\ \mathbf{b_{n+1}} = \mathcal{P}_{\mathbf{B}}(\mathbf{a_{n+1}} + \mathbf{q_n}) \\ q_{n+1} = a_{n+1} + q_n - b_{n+1} \\ \textit{avec } b_0 = x \in \mathbb{H} \textit{ et } a_0 = 0. \end{array}\right. \tag{2.12}$$

Comme première remarque, notons les différences avec l'algorithme précédent de von Neumann. Elles tiennent essentiellement en la présence à chaque itération des vecteurs p_n et q_n. Ceux-ci sont calculés après projection sur chaque convexe et représentent, d'un point de vue géométrique, le *déplacement* effectué pour aller

du nouvel itéré au point dont cet itéré est le projeté. En nous rappelant la Proposition 2.1.3, on sait que ce vecteur appartient au cône normal au convexe (A ou B) sur lequel on a projeté, au point résultat de la projection. En d'autres termes, on a donc :

$$\forall n \geq 1, \;\; p_n \in \mathcal{N}(a_n, A) \text{ et } q_n \in \mathcal{N}(b_n, B).$$

La figure 2.3 donne une illustration de l'algorithme de Boyle-Dykstra. Une itération de l'algorithme (par exemple, celle qui permet de passer de b_1 à a_2) peut être décrite de la manière suivante :

- on déplace le point courant (par exemple b_1 sur la figure) dans la dernière direction normale (p_1) au convexe sur lequel on doit projeter (A) gardée en mémoire,
- on effectue la projection (sur A) du point obtenu ($b_1 + p_1$),
- on garde en mémoire la nouvelle direction normale (p_2) obtenue ainsi que le résultat de la projection (a_2) qui est le nouvel itéré courant.

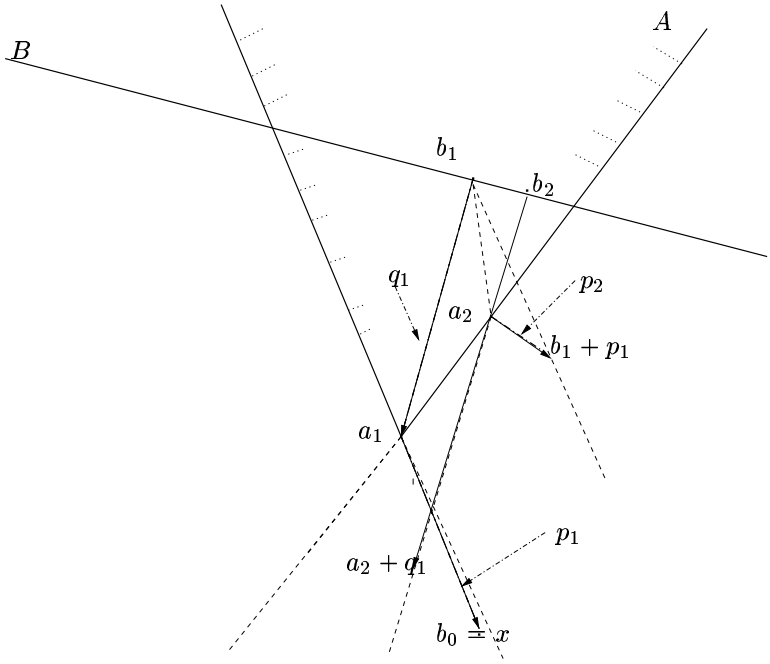


FIG. 2.3 – Illustration de l'algorithme de Boyle-Dykstra

Ce schéma a été proposé par DYKSTRA [52] en 1983 pour la recherche du projeté sur l'intersection (finie) de cônes convexes en dimension finie. Avec BOYLE, [26], il l'a étendu en 1985 aux convexes généraux dans un espace de Hilbert quelconque. Cela a été fait pour résoudre des problèmes de type moindres carrés apparaissant en Statistiques. Cet algorithme a été redécouvert indépendamment par HAN [70] en 1988 dans un contexte de dualisation d'un problème d'optimisation dans un espace euclidien. Il lui a donné le nom de méthode de *projections successives*. De là viennent les deux noms (projections successives et Boyle-Dykstra)

qui coexistent dans la littérature pour cette méthode. Cette approche par dualité a conduit à une belle justification (par GAFFKE et MATHAR [63]) de la convergence de l'algorithme .

En 1994, BORWEIN et BAUSCHKE [18] ont proposé une superbe analyse de cette méthode de projections alternées dans le cas de deux convexes. Ce travail fait suite par ailleurs à une analyse similaire sur la méthode de von Neumann (voir [17]). De plus, BAUSCHKE et LEWIS ont étendu cet algorithme à un autre type de projections : les projections de Bregman [23]. Le résultat le plus important du point de vue de notre travail est le suivant :

Théorème 2.3.1 ([18]) *Soient $\mathbb{H}$ un espace de Hilbert, A, B deux convexes fermés de $\mathbb{H}$ et x un point de $\mathbb{H}$.*

On définit les suites de Dykstra de la même manière qu'en (2.12).

Alors

$$b_n - a_n,\ b_n - a_{n+1} \to v, \tag{2.13}$$

où $v = \mathcal{P}_{\overline{B-A}}(0)$ et $\|v\| = d(A,B)$.

En particulier,

$$\|b_n - a_n\|,\ \|b_n - a_{n+1}\| \to d(A,B), \tag{2.14}$$

et

$$\frac{a_n}{n},\ \frac{b_n}{n} \to 0,\ \frac{a_n}{n} \to v,\ \frac{b_n}{n} \to -v. \tag{2.15}$$

De plus,

(i) *si $d(A,B)$ n'est pas atteinte, alors*

$$\|a_n\|,\ \|b_n\| \to +\infty. \tag{2.16}$$

(ii) *si $d(A,B)$ est atteinte, alors*

$$a_n \to \mathcal{P}_E(x),\ b_n \to \mathcal{P}_F(x), \tag{2.17}$$

où

$$E = \{a \in A : d(a,B) = d(A,B)\},\ F = \{b \in A : d(b,A) = d(A,B)\}$$

sont des convexes non vides tels que $E + v = F$.

□

Pour la preuve de ce Théorème, l'article [18] de BAUSCHKE et BORWEIN constitue une source très intéressante. La démonstration y est basée essentiellement sur les propriétés du produit scalaire d'un espace de Hilbert et la caractérisation (2.2) pour les projections.

On peut remarquer qu'en fait le cadre de ce théorème dépasse celui de convexes d'intersection non vide. On peut en déduire les deux résultats suivants :

(1) Si $A \cap B \neq \emptyset$, alors on remarque que :

$$0 \in b - a \subset \overline{B - A} \Rightarrow v = \mathcal{P}_{\overline{B-A}}(0) = 0,$$

et

$d(A, B) = 0 \ \Rightarrow E = F = A \cap B$ (où E et F sont définis dans le Théorème) .

Par suite,

$$\|\mathbf{b_n} - \mathbf{a_n}\|, \|\mathbf{b_n} - \mathbf{a_{n+1}}\| \to \mathbf{0} \text{ et } \mathbf{a_n}, \mathbf{b_n} \to \mathcal{P}_{\mathbf{A}\cap\mathbf{B}}(\mathbf{x}). \tag{2.18}$$

Ces deux résultats sont intéressants pour nous puisque d'une part, le second justifie l'usage d'un algorithme de Boyle-Dykstra pour la recherche du projeté sur une intersection de convexes ; d'autre part, le premier aide, quant à lui, à la mise en œuvre d'un test d'arrêt efficace lors de l'implémentation numérique de l'algorithme.

(2) Si $A \cap B = \emptyset$, l'algorithme peut permettre de tester si la distance entre les deux convexes n'est pas atteinte (dans ce cas, les suites principales (a_n) et (b_n) divergent) et si elle l'est, la suite (a_n) converge vers le point de A le plus proche à la fois de x ($= b_0$) et de B ; et réciproquement pour (b_n). A la limite, on récupère donc la distance entre les deux convexes.

Lorsque l'on a plus de deux convexes, l'algorithme de Boyle-Dykstra se généralise de manière naturelle en faisant des projections cycliques. Lorsque leur intersection est non vide, les principales conclusions (2.18) du Théorème 2.3.1 restent valables. On pourra consulter à ce propos [26] pour une preuve directe et [18] où on se ramène au Théorème 2.3.1 en réécrivant une intersection finie dans $\mathbb{H}$ comme une intersection de deux convexes dans $\mathbb{H}^n$ suivant l'idée de PIERRA [98].

Signalons que lorsque l'intersection finie est vide, on ne peut rien dire, contrairement au cas de deux convexes comme ci-dessus. Le comportement de l'algorithme de Boyle-Dykstra dans ce cas (au moins trois convexes) reste un problème ouvert. Le lecteur intéressé pourra trouver dans [16] une liste récente de problèmes ouverts concernant les méthodes de projections.

De même, BORWEIN et BAUSCHKE [18] proposent une série très intéressante de remarques sur l'algorithme de Boyle-Dykstra, et celui de von Neumann d'ailleurs (voir [17]), notamment sur les vitesses de convergence et les situations adaptées à son application.

Pour terminer, remarquons que le schéma de Boyle-Dykstra constitue une généralisation directe de celui de von Neumann (c'est pourquoi nous avons choisi de présenter les deux méthodes l'une après l'autre). Ceci est facile à voir en se référant encore à la Proposition 2.1.3 de la Section 2. En effet, lorsque A et B sont des sous-espaces, $\mathcal{P}_A$ et $\mathcal{P}_B$ sont linéaires et on a ainsi :

$$\forall n, \ a_{n+1} = \mathcal{P}_A(b_n + p_n) = \mathcal{P}_A(b_n) + \mathcal{P}_A(p_n) = \mathcal{P}_A(b_n),$$

car $p_n \in \mathcal{N}(a_n, A) = A^\perp \Rightarrow \mathcal{P}_A(p_n) = 0$. De même pour b_{n+1}.

Le calcul des p_n et q_n est inutile dans ce cas, et l'algorithme se ramène à celui de von Neumann. Ce fait est remarqué par DYKSTRA [52] pour des sous-espaces

vectoriels, GAFFKE et MATHAR [63] pour des sous-espaces affines. En pratique, compte tenu de cette remarque, lorsque l'un des convexes A ou B est un sous-espace, *il est inutile de calculer la composante normale qui lui correspond*.

2.4 Interprétation et vitesse de convergence

Jusqu'à nos jours, l'algorithme de Boyle-Dykstra demeure en quelque sorte un "mystère" pour les spécialistes de l'Analyse convexe. En effet, à ce jour, personne n'est parvenu à expliquer d'où provient l'idée de calculer à chaque itération les vecteurs normaux p_n et q_n à A et B respectivement. Cette intuition lumineuse demeure pour l'instant inexpliquée. Quelques tentatives d'explication existent cependant (voir par exemple [63]). Une piste possible pour interpréter l'algorithme de Boyle-Dyskstra consisterait à la relier à une des méthodes classiques d'optimisation convexe, puisqu'après tout, c'est un tel problème qui est résolu. Dans ce sens, on peut avancer sans grand risque d'erreur que cet algorithme ne devrait pas être trop éloigné de la méthode de sous-gradient classique de l'Analyse convexe.

En effet, à chaque étape de l'algorithme, on calcule un sous-gradient de la fonction d_A ou d_B ($-p_n$ et $-q_n$ respectivement), et l'itéré courant est mis à jour dans une direction de descente (p_n et q_n respectivement) en prenant un pas égal à 1. C'est exactement la démarche d'une méthode de sous-gradient avec comme nette différence qu'ici la fonction à minimiser est $d_{A\cap B}$. Tout se passe comme si on appliquait un algorithme de sous-gradient à une itération alternativement à des problèmes convexes dont les fonctions objectifs sont alternativement d_A et d_B.

Un des avantages que l'on aurait eu à rapprocher la méthode de Boyle-Dykstra d'une méthode d'optimisation convexe est que cela nous aurait donné facilement une idée de sa vitesse de convergence. Toutefois, on dispose des caractéristiques suivantes de convergence dues à BAUSCHKE et BORWEIN [18] :

- l'algorithme de Dykstra peut être "lent" : cela dépend de "l'angle" entre les deux convexes A et B. Il sera probablement difficile d'en faire une analyse de convergence simple, parce qu'on peut montrer que celle-ci dépend du point de départ (b_0) par exemple. Toutefois, il permet d'obtenir des projetés via une convergence en norme.
- Par contre, l'algorithme de Von Neumann est très facile à mettre en œuvre et est probablement plus rapide que celui de Dykstra. Malheureusement, on ne peut obtenir pour lui que de la convergence faible dans le cas général.

On vérifie en pratique qu'on ne peut obtenir au mieux qu'une convergence **linéaire**, et que cette convergence n'est obtenue que lorsqu'on a que des sous-espaces.

Chapitre 3

Approximation par matrices bistochastiques

Dans ce chapitre, nous étudions notre premier problème d'approximation matricielle : l'approximation par matrices bistochastiques. Nous introduisons pour commencer la notion de matrice bistochastique. Puis, nous aborderons le problème d'approximation par matrices bistochastiques. Après nous être assurés de l'existence d'une (unique) solution, nous proposons deux algorithmes de natures différentes pour le résoudre.

3.1 Le polytope $\mathbb{B}_n$ des matrices bistochastiques

3.1.1 Définitions et caractérisations

Soit $M = (a_{ij})_{i,j}$ une matrice carrée d'ordre n ($n \in \mathbb{N}^*$).

Définition 3.1.1 *M est appelée matrice bistochastique si on a :*

1. $a_{ij} \geq 0, \qquad i = 1, 2, \ldots, n,\ j = 1, 2, \ldots, n$;
2. $\sum_{i=1}^{n} a_{ij} = 1, \qquad j = 1, 2, \ldots, n$;
3. $\sum_{j=1}^{n} a_{ij} = 1, \qquad i = 1, 2, \ldots, n.$

Pour $n \in \mathbb{N}^*$ fixé, nous noterons $\mathbb{B}_n$ l'ensemble des matrices bistochastiques.

On peut aussi caractériser les matrices bistochastiques d'une autre manière. Rappelons que e désigne le vecteur de $\mathbb{R}^n$ dont toutes les composantes sont égales à 1.

Définition 3.1.2 *La matrice $M = (a_{ij})_{i,j}$ est une matrice bistochastique si et seulement si :*

1. *$M \geq 0$ au sens des composantes (c'est-à-dire toutes les composantes sont positives),*

2. $Me = e$,
3. $M^T e = e$.

Proposition 3.1.1 *L'ensemble $\mathbb{B}_n$ est convexe et compact.*

□

La justification de cette proposition est immédiate. D'une part, l'ensemble $\mathbb{B}_n$ est défini à partir de l'inégalité $u(M) \geq 0$ et des égalités $v(M) = 0,\ w(M) = 0$ sur les fonctions affines

$$u : M \mapsto M \quad v : M \mapsto Me - e \quad \text{et } w : M \mapsto M^T e - e.$$

Il est donc convexe, et fermé puisqu'il n'y a pas d'inégalités strictes. D'autre part, compte tenu de sa définition, toute matrice bistochastique a toutes ses composantes comprises entre 0 et 1. Il en vient que l'ensemble $\mathbb{B}_n$ est borné en plus d'être fermé : il est donc compact.

En identifiant $\mathcal{M}_n(\mathbb{R})$ à $\mathbb{R}^{n^2}$, les égalités définissant $\mathbb{B}_n$ s'écrivent respectivement :

1. $x_i \geq 0, \qquad i = 1, 2, \ldots, n^2$;
2. $\sum_{i=1}^{n} x_{jn+i} = 1, \qquad j = 0, 1, \ldots, n-1$;
3. $\sum_{i=o}^{n-1} x_{j+in} = 1, \qquad j = 1, 2, \ldots, n.$

On en déduit

Proposition 3.1.2

$$\mathbb{B}_n = \{x \in \mathbb{R}^{n^2} \,|\, \mathcal{A}x = b,\ \ x \geq 0\}, \tag{3.1}$$

où $\mathcal{A} \in \mathcal{M}_{2n,n^2}$ est définie sous la forme blocs suivante :

$$\mathcal{A} = \begin{pmatrix} 1_n & 0 & \cdots & 0 \\ 0 & 1_n & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & \cdots & \cdots & 1_n \\ I_n & \cdots & \cdots & I_n \end{pmatrix}, \tag{3.2}$$

1_n et I_n ayant été définis précédemment et $b \in \mathbb{R}^{2n}$ tel que :

$$b = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \tag{3.3}$$

□

La proposition 3.1.2 montre que $\mathbb{B}_n$ s'identifie à un polyèdre convexe fermé. Ceci est une autre justification possible de la proposition 3.1.1.

3.1.2 Points extrémaux

Nous nous intéressons aux points extrémaux du convexe $\mathbb{B}_n$. Il est connu que ces points particuliers d'un convexe présentent un grand intérêt, notamment du point de vue de l'Optimisation. Rappelons (voir Définition 1.3) qu'un point extrémal d'un convexe est un point qui ne peut s'exprimer comme combinaison convexe d'autres points du même convexe. Une propriété importante de ces points extrémaux est la suivante.

Proposition 3.1.3 (H. MINKOWSKI [77]) *Tout ensemble convexe compact est l'enveloppe convexe fermée de ses points extrémaux.*

□

En d'autres termes, dans un convexe compact, tout point s'écrit comme combinaison convexe de points extrémaux.

a) Cas $n = 2$

Lorsque $n = 2$, on peut facilement montrer (voir [76]) que les matrices de $\mathbb{B}_2$ sont celles qui peuvent se mettre sous la forme

$$M = \begin{pmatrix} a & 1-a \\ 1-a & a \end{pmatrix}, \text{ avec } a \in [0,1].$$

On peut donc écrire pour tout M appartenant à $\mathbb{B}_2$,

$$M = aI_2 + (1-a)P_1, \text{ avec } I_2 \text{ matrice identité et } P_1 \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \tag{3.4}$$

L'ensemble $\mathbb{B}_2$ est donc simplement le segment d'extrémités I_2 et P_1, qui en sont par conséquent les points extrémaux. Notons au passage la forme particulière (en 0-1) de ces points extrémaux, forme que nous retrouverons dans les paragraphes suivants et à laquelle on pouvait s'attendre en remarquant que, par définition, une matrice bistochastique a toutes ses composantes comprises entre 0 et 1.

b) Cas n quelconque

Pour déterminer les points extrémaux de $\mathbb{B}_n$, nous utiliserons la deuxième caractérisation des matrices bistochastiques présentée ci-dessus (voir (3.1),(3.2), (3.3)).

Le résultat principal sur lequel notre travail sera basé est le suivant :

Théorème 3.1.4 *Soit P un polyèdre convexe dans $\mathbb{R}^n$.*

Si P est de la forme

$$P = \{x \mid Dx = \delta, x \geq 0\},$$

avec D une matrice $m \times n$ et δ un vecteur donnés, alors les propositions suivantes sont équivalentes :

1. *$\overline{x}$ élément non nul de P est point extrémal de P ;*

2. *les colonnes de D correspondant aux composantes non nulles de $\overline{x}$ sont linéairement indépendantes.*

□

Démonstration

Ecrivons la matrice D sous la forme :

$$D = [d_1, \ldots, d_n],$$

où les d_i désignent les colonnes de D.

$(1 \Rightarrow 2)$:

Considérons un point extrémal non nul $\overline{x}$ de P.

Soit k le nombre de composantes de $\overline{x}$ non nulles. On a : $k \geq 1$. Sans perte de généralité, quitte à permuter des colonnes de D, nous pouvons toujours supposer que :

$$\overline{x} = (\overline{x}_1, \ldots, \overline{x}_k, 0, \ldots, 0), \quad k \leq n.$$

Nous devons alors montrer que les vecteurs $d_1, \ldots, d_k$ sont linéairement indépendants. Supposons par l'absurde que tel n'est pas le cas.

Alors, il existe des réels $\lambda_1, \ldots, \lambda_k$ non tous nuls tels que :

$$\sum_{i=1}^{k} \lambda_i d_i = 0. \tag{3.5}$$

On pose :

$$\lambda = (\lambda_1, \ldots, \lambda_k, 0, \ldots, 0) \in \mathbb{R}^n.$$

Alors $\lambda \neq 0$, car les réels $\lambda_1, \ldots, \lambda_k$ ne sont pas tous nuls.

Posons :

$$x^+ = \overline{x} + \lambda \quad et \quad x^- = \overline{x} - \lambda.$$

Au passage, on peut remarquer que la relation (3.5) reste valide si on la multiplie par un réel α non nul. On peut trouver alors un α non nul tel que $x_i + \alpha\lambda_i \geq 0$ et $x_i - \alpha\lambda_i \geq 0, \quad \forall i$. Ainsi, à un facteur multiplicatif près, on peut dire que λ est tel que :

$$x + \lambda \geq 0 \text{ et } x - \lambda \geq 0.$$

On a alors :

– $x^+ \neq x^-$ car $\lambda \neq 0$

– $x^+ \in P$;

en effet, on a :

$Dx^+ = b$, car $D\overline{x} = b$ et $D\lambda = \sum_i \lambda_i d_i = 0$, et $x^+ \geq 0$.

– De même, $x^- \in P$.

Alors,

$$\overline{x} = (1/2)(x^+ + x^-) \quad avec \;\; x^+ \in P, \;\; x^- \in P, \;\; x^+ \neq x^-.$$

Comme $\overline{x}$ est point extrémal,

$$\overline{x} = (1/2)(x^+ + x^-) \;\Rightarrow\; \overline{x} = x^+ = x^- \;\Rightarrow\; \lambda = 0.$$

On obtient donc une contradiction.

On a donc $1 \Rightarrow 2$.

$(2 \Rightarrow 1)$:

On considère de nouveau un point $\overline{x} = (\overline{x}_1, \ldots, \overline{x}_k, 0, \ldots, 0), \qquad k \leq n$ de P. On se place dans l'hypothèse où les vecteurs $d_1, \ldots, d_k$ sont linéairement indépendants. Nous devons montrer qu'alors $\overline{x}$ est point extrémal.

Supposons que $\overline{x}$ ne l'est pas.

Alors, il existe $y, z \in P, \;\; y \neq z$ et $t \in]0,1[$ tels que :

$$\overline{x} = (1-t)y + tz.$$

Alors, pour tout $i, \quad \overline{x}_i = (1-t)y_i + tz_i, \;\; \text{avec} \;\; y_i \geq 0, \; z_i \geq 0.$

Par suite : $y_{k+1} = \ldots = y_n = 0$ et $z_{k+1} = \ldots = z_n = 0$.

Les k-uplets $(\overline{x}_1, \ldots, \overline{x}_k), \quad (y_1, \ldots, y_k) \quad et \quad (z_1, \ldots, z_k)$ sont solutions du système linéaire :

$$\sum_{i=1}^{k} d_i w_i = b. \tag{3.6}$$

Comme les vecteurs $d_1, \ldots, d_k$ sont supposés linéairement indépendants, on a unicité des solutions de (3.6), soit :

$$x = y = z,$$

qui conduit à une contradiction.

Le théorème est donc démontré ■

Remarques :

1. Pour compléter le théorème, il faut noter que :

   $$\text{si} \;\; \overline{x} = 0 \in P, \; \text{alors} \; \overline{x} \; \text{est point extrémal de} \; P.$$

   En effet, supposons qu'il existe $y \neq z \in \;\; P, \;\; t \in]0,1[$ tel que :

   $$0 = (1-t)x + ty,$$

   soit $0 = (1-t)x_i + ty_i, \;\; \forall i.$

   Comme $z_i \geq 0 \;\; et \;\; y_i \geq 0$, on en déduit : $y_i = z_i = 0$, soit :

   $$\overline{x} = y = z.$$

2. Pour un polyèdre

   $$P = \{x \mid Dx = b, \; x \geq 0\},$$

   de nombreux résultats existent qui permettent de déterminer les points extrémaux de P lorsque D est de *rang maximal*. On peut par exemple se référer à [97].

   Le théorème 3.1.4 est en quelque sorte une généralisation de ces résultats, puisqu'aucune condition particulière de rang n'est requise pour la matrice D.

Dans un premier temps, essayons de déterminer le rang de la matrice $\mathcal{A}$ de (3.2). Puisque $\mathcal{A} \in \mathcal{M}_{2n,n^2}$, on a : $rg(\mathcal{A}) \leq 2n$. Plus précisément, on peut dire que :

$$n \leq rg(\mathcal{A}) \leq 2n - 1. \tag{3.7}$$

En effet, on remarquera que les n dernières lignes de $\mathcal{A}$ (et les n premières aussi) sont linéairement indépendantes. On en déduit que $rg(\mathcal{A}) \geq n$.

D'autre part, si nous notons L_i la $ième$ ligne de $\mathcal{A}$, on a :

$$\sum_{i=1}^{n} L_i = \sum_{i=n+1}^{2n} L_i = 1_{n^2},$$

donc

$$\sum_{i=1}^{n} L_i - \sum_{i=n+1}^{2n} L_i = 0.$$

Par suite, il existe une combinaison linéaire nulle des $2n$ lignes de $\mathcal{A}$ avec des coefficients non tous nuls. On en déduit que ces lignes ne sont pas linéairement indépendantes. D'où $rg(\mathcal{A}) < 2n$. En fait, on a :

Proposition 3.1.5

$$rg(\mathcal{A}) = 2n - 1$$

□

Démonstration :

Comme $rg(\mathcal{A}) \leq 2n - 1$ (voir 3.7), il suffit de montrer que les $2n - 1$ premières lignes de $\mathcal{A}$ sont linéairement indépendantes. Pour cela, considérons une combinaison linéaire nulle de ces lignes de $\mathcal{A}$:

$$\sum_{i=1}^{2n-1} \alpha_i L_i = 0, \quad \alpha_i \in \mathbb{R}, \quad \forall i.$$

Ecrivons les n premières colonnes de la matrice formée par ces $2n - 1$ lignes, soit les n premières colonnes de $\mathcal{A}$:

$$\begin{pmatrix} 1 & \cdots & \cdots & 1 \\ 0 & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots \\ 1 & 0 & \cdots & 0 \\ & \ddots & \vdots & \vdots \\ 0 & \cdots & 1 & 0 \end{pmatrix}$$

On a ainsi :

$$\begin{cases} \alpha_1 + \alpha_j = 0, \quad j = n+1, \ldots, 2n-1 \\ \qquad\qquad et \quad \alpha_1 = 0. \end{cases}$$

D'une manière générale, en considérant successivement, de même que ci-dessus, les colonnes suivantes par groupes de n, on obtient en fait :

$$\forall\, i = 1, \ldots, n, \quad \begin{cases} \alpha_i + \alpha_j = 0, \quad j = n+1, \ldots, 2n-1 \\ et \quad \alpha_i = 0. \end{cases}$$

D'où, $\alpha_i = 0, \quad \forall i = 1, \ldots, 2n-1$.
La proposition est donc démontrée. ■

Essayons maintenant de déterminer les points extrémaux de $\mathbb{B}_n$. Nous allons d'abord faire deux remarques d'ordre général sur les matrices bistochastiques. Soit $M = (a_{ij})_{i,j}$ une matrice bistochastique.

1. On a : $\forall\ (i,j), \ \ 0 \leq a_{ij} \leq 1$,

2. Si l'une des composantes de M vaut 1, alors les autres composantes de la ligne et de la colonne auxquelles elle appartient sont toutes égales à 0.

Soit donc S une matrice bistochastique, supposons qu'elle est un *point extrémal* de $\mathbb{B}_n$. Alors, d'après le Théorème 3.1.4, les colonnes de $\mathcal{A}$ (voir (3.2) correspondant aux composantes non nulles de S doivent être linéairement indépendantes. On en déduit :

$\rightarrow$ S **a au maximum** $2n-1$ **composantes non nulles.** En effet, si tel n'est pas le cas, d'après la Proposition 3.1.4, les colonnes de $\mathcal{A}$ correspondant aux composantes non nulles de S sont linéairement indépendantes. Il existerait alors un système d'au moins $2n$ colonnes de $\mathcal{A}$ linéairement indépendantes, ce qui est en contradiction avec la Proposition 3.1.5.

$\rightarrow$ S **a au moins une ligne composée d'un seul élément non nul.** Sinon, toutes les lignes de S ont au moins 2 éléments non nuls, ce qui porterait le nombre d'éléments non nuls de S à au moins $2n$. Contradiction.

En fait, on peut montrer :

Proposition 3.1.6 *Soit S un point extrémal de $\mathbb{B}_n$.*
Toutes les lignes de S ont une et une seule composante non nulle (qui vaut alors 1). □

Démonstration :
On procède par récurrence sur n.
Pour n=1 : c'est immédiat.
Supposons que la proposition est vraie pour tout $k \leq n$, et montrons qu'elle l'est pour $n+1$.
Soit donc S une matrice bistochastique carrée d'ordre $n+1$, i.e. $S \in \mathbb{B}_{n+1}$.

D'après les remarques faites ci-dessus, S a au moins une ligne ayant comme unique composante non nulle 1. S peut alors s'écrire sous la forme bloc suivante :

$$S = \begin{pmatrix} S_1 & 0 & S_2 \\ 0 & 1 & 0 \\ S_3 & 0 & S_4 \end{pmatrix},$$

les sous-matrices S_1, S_2, S_3, S_4 ayant les dimensions adéquates. Considérons la matrice S' carrée d'ordre n définie par :

$$S' = \begin{pmatrix} S_1 & S_2 \\ S_3 & S_4 \end{pmatrix}.$$

Cette matrice S' est une matrice bistochastique d'ordre n, de manière évidente.

De plus, S' est un point extrémal de $\mathbb{B}_n$.

En effet, si tel n'est pas le cas, il existe une combinaison convexe d'éléments M'_i de $\mathbb{B}_n$ telle que :

$$S' = \sum_i \beta_i M'_i, \quad 0 \leq \beta_i \leq 1 \ \ \forall i, \ \ \sum_i \beta_i = 1.$$

En partitionnant chaque M'_i de la même manière que S' :

$$M'_i = \begin{pmatrix} M_{i1} & M_{i2} \\ M_{i3} & M_{i4} \end{pmatrix},$$

on peut construire des matrices carrées M_i d'ordre $n+1$:

$$M_i = \begin{pmatrix} M_{i1} & 0 & M_{i2} \\ 0 & 1 & 0 \\ m_{i3} & 0 & M_{i4} \end{pmatrix},$$

qui sont bistochastiques et telles que :

$$S = \sum_i \beta_i M_i, \quad 0 \leq \beta_i \leq 1 \ \ \forall i, \ \ \sum_i \beta_i = 1,$$

ce qui est absurde, compte tenu de la définition d'un point extrémal.

S' étant un point extrémal de $\mathbb{B}_n$, on a, d'après l'hypothèse de récurrence, que toutes ses lignes ont une et une seule composante non nulle, 1. Par suite, toutes les lignes de S ont comme unique composante non nulle 1. La Proposition 3.1.6 est ainsi prouvée. ■

Définition 3.1.3 (Matrice de permutation [78]) *Soit P une matrice carrée d'ordre n.*

On dit que P est une matrice de permutation si toutes ses lignes et toutes ses colonnes ont chacune exactement une composante égale à 1, toutes les autres étant égales à 0.

Ainsi, on a :

Proposition 3.1.7 *Une matrice bistochastique dont toutes les lignes ont une unique composante non nulle (égale alors à 1) est une matrice de permutation.* □

La Proposition 3.1.6 apparaît alors comme exprimant un résultat plus ancien concernant les matrices bistochastiques.

Théorème 3.1.8 (BIRKHOFF, 1946 [78]) *Une matrice bistochastique S est un point extrémal de $\mathbb{B}_n$ si, et seulement si, S est une matrice de permutation.* □

Démonstration :

Les Propositions 3.1.6 et 3.1.7 expriment que tout point extrémal de $\mathbb{B}_n$ est une matrice de permutation.

Réciproquement, toute matrice de permutation est un point extrémal de $\mathbb{B}_n$. En effet, si P est une matrice de permutation, chacune de ses lignes possède exactement une composante non nulle. Les colonnes de la matrice $\mathcal{A}$ correspondantes forment une matrice de la forme par blocs :

$$\begin{pmatrix} I_n \\ C \end{pmatrix},$$

où C est une sous-matrice carrée d'ordre n. Cette dernière matrice est de manière évidente de rang n : il suffit d'en considérer les n premières lignes. On en déduit que ses colonnes sont linéairement indépendantes. D'après le Théorème 3.1.4, P est alors un point extrémal de $\mathbb{B}_n$ ■

Le Théorème de Birkhoff (ou de Birkhoff-Von Neumann suivant les auteurs [38]) est un résultat très connu en Analyse convexe. De fait, de nombreuses démonstrations en existent. D'une manière générale, celles-ci peuvent être classées en deux groupes.

Les démonstrations dites *combinatoires* qui consistent en général à exhiber, pour une matrice bistochastique quelconque, une combinaison convexe de matrices de permutation qui lui est égale. Le plus souvent, elles présentent un algorithme itératif qui permet de déterminer une telle combinaison. On peut se référer pour cela à [38],[90].

La deuxième classe de preuves est celle des démonstrations *géométriques*. La preuve que nous avons introduite ci-dessus entre justement dans cette catégorie. Ces preuves (voir [78], [90]) utilisent toutes comme résultat central le fait qu'une matrice bistochastique, point extrémal de $\mathbb{B}_n$, a au plus $2n - 1$ composantes non nulles. Les différences proviennent essentiellement de la manière dont ce résultat central est justifié.

Notre preuve est, à notre avis, assez originale parce que, d'une part, elle utilise une expression explicite de la matrice $\mathcal{A}$ définissant le polyèdre des matrices bistochastiques et que d'autre part, elle fait apparaître le Théorème de Birkhoff comme étant un corollaire d'un résultat de programmation linéaire : le Théorème 3.1.4.

3.2 Approximation par matrices bistochastiques

Le problème d'approximation par des matrices bistochastiques s'exprime comme suit :

$$(P)\begin{cases} \text{Soit} \quad M \in \mathcal{M}_n(\mathbb{R}). \\ \text{Trouver} \quad \overline{\overline{M}} \in \mathbb{B}_n \ \text{ tel que :} \\ \|M - \overline{\overline{M}}\| = \inf\{\|M - S\|, \quad S \in \mathbb{B}_n\}. \end{cases}$$

3.2.1 Motivations

Avant de continuer, nous allons préciser les motivations de notre étude du problème d'approximation par matrices bistochastiques. Ces matrices apparaissent dans différentes théories mathématiques, notamment en théorie des probabilités, en théorie de la majorisation (voir [90]). Il y a eu énormément de travaux mathématiques concernant les matrices bistochastiques, concernant notamment leur géométrie et la conjecture de van Der Waerden. Cette conjecture, aujourd'hui démontrée par FALIKMAN [55], EGORYCHEV [53] au début des années 80, stipulait que la valeur minimale du *permanent* des matrices sur l'ensemble des matrices bistochastiques est $\frac{n!}{n^n}$ et est atteinte pour la matrice dont toutes les composantes valent $\frac{1}{n}$. Il s'agit de la matrice J_n que nous définissons ci-après. Pour plus d'informations sur les matrices bistochastiques et sur la structure de $\mathbb{B}_n$, nous conseillons la lecture de [30],[31],[32], [33], [67], [68], [89]. D'un point de vue pratique, les matrices bistochastiques sont utilisées dans différents domaines : Recherche opérationnelle [24], en Physique [47], en Théorie des graphes [25] et aussi en Mécanique quantique [87]. Dans toutes ces situations, les matrices bistochastiques considérées, par exemple lorsqu'elles sont obtenues au moyen d'une boîte noire, peuvent avoir perdu toutes ou une partie des propriétés qui en font une matrice bistochastique. Dans ce cas, une solution serait de la remplacer par la matrice bistochastique la plus proche d'elle. Ceci est une motivation classique.

Une motivation moins basique est que le problème d'approximation par matrice bistochastique apparait naturellement dans la résolution de certains types de problèmes en mathématiques. C'est par exemple le cas dans le problème d'*agrégation de préférences* que nous allons étudier dans un prochain paragraphe.

3.2.2 Premiers résultats

L'ensemble $\mathbb{B}_n$ est convexe et compact (voir Proposition 3.1.1) de $\mathcal{M}_n(\mathbb{R})$. Il a aussi la particularité d'être contenu dans un sous-espace affine de $\mathcal{M}_n(\mathbb{R})$ et donc est d'intérieur vide.

Compte tenu de ces remarques, une première réponse au problème d'approximation (P) est donnée par le Théorème de projection (voir Théorème 2.1.1).

On a :

Proposition 3.2.1 *Soit $M \in \mathcal{M}_n(\mathbb{R})$.*

Il existe une et une seule matrice bistochastique $\overline{\overline{M}}$ telle que :

$$\|M - \overline{\overline{M}}\| = \inf\{\|M - S\|, \quad S \in \mathbb{B}_n\}.$$

La matrice $\overline{\overline{M}}$ est caractérisée par :

$$\begin{cases} \overline{\overline{M}} \in \mathbb{B}_n ; \\ \langle\langle M - \overline{\overline{M}}, S - \overline{\overline{M}} \rangle\rangle \leq 0, \quad \forall\ S \in \mathbb{B}_n . \end{cases} \tag{3.8}$$

□

D'après le Théorème de Birkhoff (Théorème 3.1.8) et la proposition 3.1.3, la caractérisation (3.8) est équivalente à la suivante :

$$\begin{cases} \overline{\overline{M}} \in \mathbb{B}_n ; \\ \langle\langle M - \overline{\overline{M}}, P - \overline{\overline{M}} \rangle\rangle \leq 0, \quad \text{pour toute matrice } P \text{ de permutation.} \end{cases} \tag{3.9}$$

En effet, il suffit de remarquer que :

1. Pour tout $S \in \mathbb{B}_n$, il existe $(\alpha_i)_i$ tel que :

$$0 \leq \alpha_i \leq 1, \quad \sum_i \alpha_i = 1 \text{ et } S = \sum_i \alpha_i P_i,$$

avec P_i matrice de permutation, pour tout i.

2. Pour $(\alpha_i)_i$ tel que $0 \leq \alpha_i \leq 1$ et $\sum_i \alpha_i = 1$,

$$\langle\langle M - \overline{\overline{M}}, \sum_i \alpha_i P_i - \overline{\overline{M}} \rangle\rangle = \sum_i \alpha_i \langle\langle M - \overline{\overline{M}}, P_i - \overline{\overline{M}} \rangle\rangle.$$

La caractérisation (3.9) peut se reformuler sous la forme :

$$\begin{cases} \overline{\overline{M}} \in \mathbb{B}_n , \\ tr((M - \overline{\overline{M}})^T (P - \overline{\overline{M}})) \leq 0, \quad \text{pour toute matrice } P \text{ de permutation.} \end{cases} \tag{3.10}$$

Pour trouver $\overline{\overline{M}}$ en utilisant la caractérisation (3.10), on est amené à résoudre un système d'équations ou inéquations, comportant en particulier $n!$ inéquations. Il est facile d'en conclure que cette caractérisation a toutes les chances de ne pas nous permettre de calculer *"explicitement"* $\overline{\overline{M}}$. Et ceci, même pour des petites valeurs de n. En effet, pour $n = 2$, le problème se ramène à (voir (3.4)) :

trouver $a \in [0, 1]$ tel que

$$\overline{\overline{M}} = \begin{pmatrix} a & 1-a \\ 1-a & a \end{pmatrix} \text{ et } tr((M - \overline{\overline{M}})^T (P - \overline{\overline{M}})) \leq 0 \text{ pour } P = I_2, P_1 \tag{3.11}$$

qui n'est pas forcément "facile" à résoudre. Nous reviendrons sur ce problème pour $n = 2$ un peu plus loin pour en donner une solution "explicite".

Manifestement en tout cas, l'approche directe semble ne pas pouvoir nous conduire à la solution du problème (P). Nous devons donc nous résoudre à considérer une approche numérique.

3.2.3 Optimisation quadratique

La première idée de résolution numérique de notre problème d'approximation par matrices bistochastique consiste à exploiter l'isomorphisme entre $\mathcal{M}_n(\mathbb{R})$ et $\mathbb{R}^{n^2}$ que nous avons explicité à la section précédente (Section 3.1). Le problème peut alors se réecrire comme suit : trouver $\overline{\overline{m}} \in \mathbb{R}^{n^2}$ tel que

$$\begin{array}{rcll} \frac{1}{2}\|m - \overline{\overline{m}}\|_2^2 & = & \min & \frac{1}{2}\|m - s\|_2^2 \\ & & \text{tq.} & \mathcal{A}s = b, \\ & & & s \geq 0, \quad s \in \mathbb{R}^{n^2}, \end{array} \tag{3.12}$$

où m est une vecteur quelconque donné de $\mathbb{R}^{n^2}$, $\|\cdot\|_2$ désigne la norme euclidienne classique de $\mathbb{R}^{n^2}$, et où $\mathcal{A}$ et b sont tels que définis à la Proposition 3.1.2.

Ecrit sous cette forme, notre problème d'approximation apparaît comme un problème d'optimisation quadratique, en particulier, un problème de moindres carrés, dans $\mathbb{R}^{n^2}$. Pour le résoudre, on pourrait donc utiliser l'un des nombreux algorithmes d'optimisation quadratique qui existent, comme par exemple, les algorithmes de type contraintes actives, ou des algorithmes spécialisés pour les problèmes de moindres carrés linéaires.

De tels tests ont été effectués où le problème a été résolu en utilisant des routines spécialisées du logiciel *Matlab*, notamment *quadprog* (version mise à jour de l'ancienne routine *qp*) qui est un algorithme de type contraintes actives pour la résolution de problèmes quadratiques (de taille moyenne) et *lsqlin* qui est un algorithme spécialisé aux problèmes de moindres carrés linéaires. Ces deux routines sont des composantes de la boite à outils d'optimisation de *Matlab*. Il a été observé, suite à ces test que les temps de calculs pour obtenir la solution devenaient rapidement prohibitifs. En effet, pour des matrices aléatoires de tailles $n = 10$, on a des temps moyens de calculs de l'ordre de 3.5 secondes. Ce temps moyen devient supérieur à 5 minutes (350 secondes, soit une multiplication par un facteur 100 !) lorsque l'on double la valeur de n ($n = 20$).

Il apparaît assez rapidement que l'utilisation de l'optimisation quadratique ne peut pas nous permettre une résolution efficace et rapide de notre problème (noter que nous nous proposer de résoudre des problèmes pour des valeurs de n de l'ordre de quelques centaines, voire du millier). Comme nous le prédisions au premier chapitre, ceci est dû au fait que nous nous ramenons à travailler dans un espace de dimension n^2, nettement plus grand que celui à n dimension où le problème est posé, dont la dimension croit exponentiellement lorsque n augmente. Pour une résolution efficace, il nous faut donc des algorithmes adaptés à la structure matricielle des données du problèmes. Aussi, allons-nous nous rabattre sur une solution itérative, qui passe par les méthodes de projections que nous avons introduites au chapitre précédent.

3.3 Approximation par projection alternées

Pour utiliser un algorithme de projections alternées en vue de résoudre notre problème, il nous faut écrire $\mathbb{B}_n$ comme une intersection de convexes. Il est facile

de voir que

$$\mathbb{B}_n = \Lambda^+ \cap \mathcal{LC}1,$$

où

$$\Lambda^+ = \{M \in \mathcal{M}_n(\mathbb{R}) \mid M \geq 0\}$$

et

$$\mathcal{LC}1 = \{M \in \mathcal{M}_n(\mathbb{R}) \mid Me = e,\ M^T e = e\}.$$

On remarque aussi, facilement, que Λ^+ et $\mathcal{LC}1$ sont des ensembles convexes ; le premier étant un cône et le second un sous-espace affine. Cette écriture de $\mathbb{B}_n$ en tant qu'intersection de convexes, nous permettra d'appliquer une méthode de type Boyle-Dykstra à la résolution de notre problème d'approximation. La mise en œuvre de cette méthode nécessite la connaissance des projections respectivement sur Λ^+ et $\mathcal{LC}1$.

3.3.1 Projection sur Λ^+

On rappelle que pour un réel a, on note

$$a^+ = \max(a, 0).$$

Pour une matrice $M = (a_{ij})$ de E, on appelle $M^+ = (m_{ij})$ la matrice dont toutes les composantes sont définies par :

$$m_{ij} = a_{ij}^+, \ \ \forall i, j.$$

On a vu (voir Proposition 2.1.4 au chapître 2) que la projection sur Λ^+ peut s'écrire :

$$\forall M \in \mathcal{M}_n(\mathbb{R}), \mathcal{P}_{\Lambda^+}(M) = M^+.$$

3.3.2 Projection sur $\mathcal{LC}1$

Soit M une matrice carrée d'ordre n.

Définition 3.3.1 *M est dite **bistochastique généralisée** ou **lc1** si elle vérifie :*

1. $\sum_{i=1}^n a_{ij} = 1, \qquad j = 1, \ldots, n$;
2. $\sum_{j=1}^n a_{ij} = 1, \qquad i = 1, \ldots, n.$

On voit que les matrices bistochastiques sont en fait des matrices **lc1** satisfaisant en plus des contraintes de positivité sur les composantes. De fait, une matrice bistochastique est **lc1**, la réciproque étant fausse.

Il est facile de voir que les matrices bistochastiques généralisées forment le sous-espace affine $\mathcal{LC}1$ que nous avons introduit précédemment

Considérons donc le problème d'approximation par les matrices bistochastiques généralisées. On est toujours placé dans l'espace de Hilbert $(E = \mathcal{M}_n(\mathbb{R}), \langle\langle \cdot, \cdot \rangle\rangle)$.

Proposition 3.3.1 *$\mathcal{LC}1$ est un sous-espace affine, donc convexe et fermé de $\mathcal{M}_n(\mathbb{R})$*

□

La justification de la proposition est claire ■

Le problème d'approximation s'exprime alors de la manière. Soit $M \in \mathcal{M}_n(\mathbb{R})$;

$$\begin{array}{l} \text{trouver } \overline{M} \in \mathcal{LC}1 \text{ tel que :} \\ \|M - \overline{M}\| = \inf\{\|M - B\|,\ B \in \mathcal{LC}1\}. \end{array} \tag{3.13}$$

La réponse à ce problème est alors donnée par le corollaire du Théorème de projection (voir Théorème 2.2). On obtient :

Proposition 3.3.2 *Soit $M \in \mathcal{M}_n(\mathbb{R})$.*

*Il existe une et une seule matrice **lc1** $\overline{M}$ telle que :*

$$\|M - \overline{M}\| = \inf\{\|M - B\|,\ B \in \mathcal{LC}1\}.$$

La matrice $\overline{M}$ est caractérisée par :

– $\overline{M} \in \mathcal{LC}1$,

– $M - \overline{M} \in \mathcal{LC}1^{\perp}$.

où $\mathcal{LC}1^{\perp}$ désigne le sous-espace orthogonal dans $\mathcal{M}_n(\mathbb{R})$ de $\mathcal{LC}1$. □

Démonstration

Comme $\mathcal{LC}1$ est un sous-espace affine de $\mathcal{M}_n(\mathbb{R})$, il existe un sous-espace vectoriel de $\mathcal{M}_n(\mathbb{R})$, V, dit direction de $\mathcal{LC}1$ et une matrice B_0 de $\mathcal{LC}1$ tels que :

$$\mathcal{LC}1 = B_0 + V.$$

Fixons B_0.

D'autre part, il existe $M' \in \mathcal{M}_n(\mathbb{R})$ tel que : $M = B_0 + M'$ (car $\mathcal{M}_n(\mathbb{R})$ est aussi bien un espace vectoriel qu'un espace affine).

Alors, le problème d'approximation se réécrit : trouver $\overline{M}' \in \mathcal{M}_n(\mathbb{R})$ tel que

$$\|M' - \overline{M}'\| = \inf\{\|M' - B'\|,\ B' \in V\}, \tag{3.14}$$

où $\overline{M} = B_0 + \overline{M}'$.

Comme V est un sous-espace vectoriel de $\mathcal{M}_n(\mathbb{R})$, le corollaire du Théorème de projection nous dit qu'il existe une et une seule matrice $\overline{M}'$ solution de (3.14). Donc, $\overline{M}$ existe et est unique.

D'autre part, $\overline{M}'$ est caractérisée par :

$$\forall B' \in V, \quad \langle\langle M' - \overline{M}', B'\rangle\rangle = 0, \tag{3.15}$$

soit : $M' - \overline{M}' \in V^{\perp}$.

Cependant, $M' - \overline{M}' = M - \overline{M}$ et $\mathcal{LC}1^{\perp} = V^{\perp}$. D'où

$$(3.15) \Leftrightarrow M - \overline{M} \in V^{\perp} = \mathcal{LC}1^{\perp}.$$

Ceci termine la preuve du théorème.

■

Remarque :

La caractérisation

$$M - \overline{M} \in V^{\perp} = \mathcal{LC}1^{\perp}$$

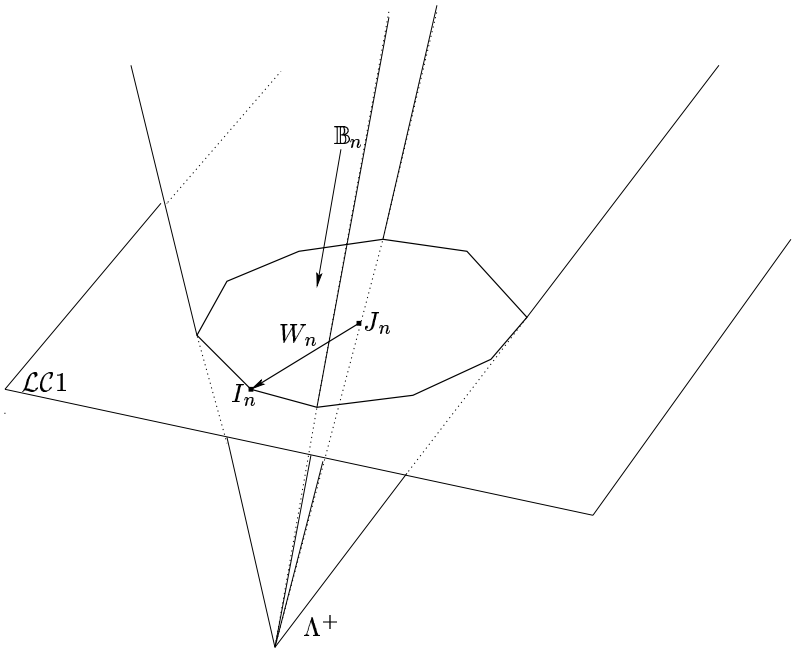


FIG. 3.1 – Visualisation 3-D de $\mathbb{B}_n$

peut être exprimée sous la forme : il existe une constante k telle que

$$\langle\langle M - \overline{M}, B\rangle\rangle = k, \quad \forall B \in \mathcal{LC}1. \tag{3.16}$$

Nous disposons donc d'une caractérisation de $\overline{M}$, nous allons l'utiliser pour en trouver une *forme explicite*.

Tout d'abord, on introduit les matrices suivantes :

– $J_n = (J_{ij})_{i,j}$ telle que $\forall i, j \;\; J_{ij} = 1/n$.

– $W_n = I_n - J_n$.

On a la configuration illustrée par la figure 3.3.2.

Faisons quelques remarques sur les matrices J_n et W_n.

♦ J_n est une matrice **lc1** (et même bistochastique, tout simplement). C'est la seule dont toutes les composantes sont égales. Elle joue le rôle de "centre" dans $\mathbb{B}_n$.

♦ J_n est "*idempotente*" i.e. $J_n^2 = J_n$.

En effet,

Posons : $J_n^2 = (c_{ij})_{i,j}$. Alors :

$$c_{ij} = \sum_k J_{ik} J_{kj} = \sum_{k=1}^{n} 1/n^2 = n \cdot (1/n^2) = 1/n$$

♦ W_n est "*idempotente*". Ceci est une conséquence du point précédent.

♦ J_n est "*absorbante*" dans l'ensemble des matrices bistochastiques généralisées ; i.e.

$$\forall B \in \mathcal{LC}1, \qquad J_n B = B J_n = J_n.$$

En effet, si $J_nB = (c_{ij})_{i,j}$, on a :

$$c_{ij} = \sum_k J_{ik}b_{kj} = 1/n \sum_k b_{kj} = 1/n, \quad \text{car} \quad \sum_k b_{kj} = 1.$$

De même pour BJ_n.

Notons que ces matrices J_n et W_n ne sont pas inconnues aux lecteurs habitués aux problèmes d'approximation. Les mêmes matrices apparaissent dans différentes autres situations en mathématiques, notamment lorsque l'on étudie le problème d'approximation par des matrices distances euclidiennes (voir [1], [3], [4]).

Essayons maintenant de trouver $\overline{M}$ à partir de la caractérisation de la Proposition 3.3.2. Nous cherchons une matrice bistochastique généralisée $\overline{M}$ telle que : $M - \overline{M} \in \mathcal{LC}1^{\perp}$.

Posons :

$$V = \{x \mid \mathcal{A}x = 0\} = \{M \in E \mid Me = 0,\ M^Te = 0\}.$$

V est un sous-espace vectoriel de $\mathcal{M}_n(\mathbb{R})$. C'est le noyau de la matrice $\mathcal{A}$ ($\mathcal{A}$ est définie en (3.2)). D'autre part, V est la direction du sous-espace affine $\mathcal{LC}1$. Donc :

$$\mathcal{LC}1^{\perp} = V^{\perp}.$$

Dans un premier temps, nous allons essayer de déterminer $V^{\perp}$. Puis, à partir de là, nous allons expliciter $\overline{M}$ en utilisant la caractérisation :

$$\overline{M} \in \mathcal{LC}1 \quad et \quad M - \overline{M} \in V^{\perp}.$$

Considérons l'application l suivante :

$$l : \mathcal{M}_n(\mathbb{R}) \to \mathbb{R}^n \times \mathbb{R}^n, \quad M \mapsto (Me, M^Te).$$

C'est une application linéaire, de manière évidente. De plus on a :

$$\mathcal{LC}1 = \{B \in \mathcal{M}_n(\mathbb{R}) : l(B) = (e,e)\};\ \ V = \{B \in \mathcal{M}_n(\mathbb{R}) : l(B) = (0,0)\},$$

soit $V = ker(l)$. On a alors :

$$V^{\perp} = (ker(l))^{\perp} = im(l^T).$$

Déterminons alors l^T. On a $l^T : \mathbb{R}^n \times \mathbb{R}^n \to \mathcal{M}_n(\mathbb{R})$ tel que :

$$\forall (u,v) \in \mathbb{R}^n \times \mathbb{R}^n,\ \forall M \in \mathcal{M}_n(\mathbb{R}) \quad \langle\langle l^T(u,v), M\rangle\rangle = \langle (u,v), l(M)\rangle_{n\times n}$$

où $\langle .,. \rangle_{n\times n}$ désigne le produit scalaire de $\mathbb{R}^n \times \mathbb{R}^n$ défini par :

$$\langle (u,v), (u',v')\rangle_{n\times n} = \langle u, u'\rangle + \langle v, v'\rangle,$$

$\langle .,. \rangle$ étant le produit scalaire usuel de $\mathbb{R}^n$.

Par suite, pour tous (u,v), pour tout M :

$$\begin{aligned}
\langle\langle l^T(u,v), M\rangle\rangle &= \langle (u,v), (Me, M^Te)\rangle_{n\times n}, \\
&= \langle u, Me\rangle + \langle v, M^Te\rangle, \\
&= \langle\langle ue^T, M\rangle\rangle + \langle\langle ve^T, M^T\rangle\rangle, \\
&= \langle\langle ue^T, M\rangle\rangle + \langle\langle ev^T, M\rangle\rangle, \\
&= \langle\langle ue^T + ev^T, M\rangle\rangle.
\end{aligned}$$

D'où,

$$\forall (u,v) \in \mathbb{R}^n \times \mathbb{R}^n, \;\; l^T(u,v) = ue^T + ev^T.$$

Proposition 3.3.3 *On a :*

$$V^{\perp} = \{ue^T + ev^T; u \in \mathbb{R}^n, v \in \mathbb{R}^n\} \square$$

Le problème de projection se réexprime alors comme suit :
Trouver $\overline{M} \in \mathcal{LC}1$, $u, v \in \mathbb{R}^n$ tel que :

$$\left\{ \begin{array}{rcl} \overline{M}e & = & e, \\ \overline{M}^T e & = & e, \\ M - \overline{M} & = & ue^T + ev^T. \end{array} \right. \tag{3.17}$$

Ainsi,

$$M - \overline{M} = ue^T + ev^T \Rightarrow \overline{M} = M - ue^T - ev^T. \tag{3.18}$$

Cette dernière relation injectée dans la première équation de (3.17) conduit à :

$$\overline{M}e = e \;\; \Rightarrow \;\; e = Me - ue^Te - ev^Te \tag{3.19}$$

$$\Rightarrow \;\; e = Me - nu - ev^Te. \tag{3.20}$$

De même, avec la seconde équation de (3.17), on obtient :

$$e = M^Te - eu^Te - nv. \tag{3.21}$$

De (3.21), on déduit :

$$v = \frac{1}{n}M^Te - \frac{1}{n}eu^Te - \frac{1}{n}e.$$

D'où,

$$\begin{aligned} (3.20) \;\; &\Rightarrow \;\; e = Me - nu - \frac{1}{n}e(M^Te - eu^Te - e)^Te \\ &\Rightarrow \;\; e = Me - nu - \frac{1}{n}e(e^TM - e^Tue^T - e^T)e \\ &\Rightarrow \;\; e = Me - nu - \frac{1}{n}ee^TMe + ee^Tu + e \\ &\Rightarrow \;\; e = (I_n - \frac{1}{n}ee^T)Me - (nI_n - ee^T)u + e, \end{aligned}$$

soit :

$$(nI_n - ee^T)u = Me - J_nMe. \tag{3.22}$$

En procédant de la même manière pour v, on obtient :

$$(nI_n - ee^T)v = M^Te - J_nM^Te. \tag{3.23}$$

Les vecteurs u et v sont donc solutions de systèmes linéaires qui ne diffèrent que par leurs seconds membres. Notons E_n la matrice des systèmes (3.22) et (3.23). Plus précisément, on a :

$$E_n = \begin{pmatrix} n-1 & -1 & \dots & -1 \\ -1 & n-1 & \dots & -1 \\ \vdots & \ddots & \ddots & \vdots \\ -1 & \dots & -1 & n-1 \end{pmatrix}.$$

Proposition 3.3.4 *La matrice E_n est de rang $n-1$. Son noyau est l'espace de dimension 1 engendré par le vecteur e.* □

Démonstration

Tout d'abord, on peut remarquer que E_n ne peut être de rang n. En effet, la somme de toutes les lignes donne le vecteur-ligne dont les composantes sont nulles. Donc,

$$rg(E_n) \leq n-1.$$

Rappelons qu'on ne change pas le rang d'une matrice en ajoutant à une ligne (respectivement une colonne) une combinaison linéaire des autres lignes (respectivement colonnes). Ainsi, E_n est de même rang que :

$$\begin{pmatrix} n-1 & -1 & \dots & -1 \\ -n & n & \dots & 0 \\ \vdots & 0 & \ddots & \vdots \\ -n & \dots & 0 & n \end{pmatrix},$$

la seconde matrice est obtenue à partir de E_n en ajoutant aux $n-1$ dernières lignes l'opposée de la première. Il est évident de voir que celle-ci est de rang $n-1$, puisqu'elle est de rang au plus égal à $n-1$ et qu'en plus, elle contient une sous-matrice carrée d'ordre $n-1$: les $n-1$ dernières lignes et colonnes de la matrice forment la matrice nI_{n-1}. Par suite, $rg(E_n) = n-1$. Donc, le noyau de E_n est de dimension 1. En remarquant que :

$$E_n e = (nI_n - ee^T)e = nI_n e - e(e^T e) = ne - ne = 0$$

on termine aisément la démonstration de la proposition.

■

Puisque E_n est de rang $n-1$ et de noyau, $ker(E_n)$, connu, pour résoudre les systèmes (3.22) et (3.23), il nous suffit maintenant d'en connaître pour chacun une solution particulière. Pour le système (3.22), on voit que :

$$\begin{aligned} E_n(\frac{1}{n}Me) &= (nI_n - ee^T)\frac{1}{n}Me, \\ &= (I_n - \frac{1}{n}ee^T)Me, \\ &= Me - \frac{1}{n}ee^T Me, \\ &= Me - J_n Me. \end{aligned}$$

Le vecteur $\frac{1}{n}Me$ est donc une solution particulière de (3.22). L'ensemble de ces solutions est :

$$S_{(3.22)} = \{\frac{1}{n}Me + ke, k \in \mathbb{R}\}.$$

De même, on détermine l'ensemble des solutions de (3.23) :

$$S_{(3.23)} = \{\frac{1}{n}M^T e + k'e, k \in \mathbb{R}\}.$$

A ce stade, nous savons donc que les vecteurs u et v que nous recherchons s'écrivent :

$$u = \frac{1}{n}Me + ke \text{ et } v = \frac{1}{n}M^T e + k'e,$$

pour un k et un k' tous deux réels.
En réinjectant ces informations dans (3.20), soit

$$e = Me - n(\frac{1}{n}Me + ke) - e(\frac{1}{n}M^T e + k'e)^T e,$$

on obtient :

$$(k + k')e = -\frac{1}{n}(I_n - J_n M)e \text{ ou } k + k' - \frac{1}{n^2}e^T(I_n - J_n M)e.$$

Donc u et v sont déterminés par :

$$\left\{ \begin{array}{rcl} u & = & \frac{1}{n}Me + ke, \\ v & = & \frac{1}{n}M^T e + k'e, \\ (k + k')e & = & -\frac{1}{n}(I_n - J_n M)e. \end{array} \right. \tag{3.24}$$

Alors, à partir de (3.18) en utilisant (3.24), on obtient :

$$\overline{M} = W_n M W_n + J_n. \tag{3.25}$$

Réciproquement, on a bien :

Proposition 3.3.5 *$\overline{M}$ vérifie la relation de caractérisation de la Proposition 3.3.2*
□

Démonstration
♦ $\overline{M} \in \mathcal{LC}1$.
En effet, soit $e^T = (1, \ldots, 1)$, $e \in \mathbb{R}^n$. Comme $J_n \in \mathcal{LC}1$ et $J_n^T = J_n$, on a :

$$J_n e = e \text{ et } J_n^T e = e.$$

On en déduit que :

$$W_n e = (I_n - J_n)e = e - e = 0 \text{ et } W_n^T e = (I_n - J_n)e = e - e = 0.$$

D'où :

$$\overline{M}e = e \text{ et } \overline{M}^T e = e.$$

On en déduit le résultat.
♦ $M - \overline{M} \in \mathcal{LC}1^{\perp}$.
En effet, compte tenu de la remarque ci-dessus, nous allons utiliser la caractérisation (3.16).
Soit $B \in \mathcal{LC}1$. On doit montrer que : $\langle\langle M - \overline{M}, B\rangle\rangle = cte$. Par définition,

$$\langle\langle M - \overline{M}, B\rangle\rangle = tr((M - \overline{M})^T B) = tr(B^T(M - \overline{M})).$$

On a :

$$\overline{M} = W_n M W_n + J_n \quad \Longrightarrow \quad M - \overline{M}M - W_n M W_n - J_n,$$

$$\begin{aligned} B^T(M-\overline{M}) &= B^TM - B^TW_nMW_n - B^TJ_n, \\ &= B^TM - B^TW_nMW_n - J_n, \ \text{car} \ B \in \mathcal{LC}1 \Rightarrow B^T \in \mathcal{LC}1. \end{aligned}$$

D'où :

$$\langle\langle M-\overline{M}, B\rangle\rangle = tr(B^TM) - tr(B^TW_nMW_n) - 1, \quad \text{car} \quad tr(J_n) = 1.$$

Or on a :

$$\begin{aligned} W_nMW_n &= (I_n - J_n)(M - MJ_n), \\ &= M - MJ_n - J_nM + J_nMJ_n; \\ B^T(W_nMW_n) &= B^TM - B^TMJ_n - B^TJ_nM + B^TJ_nMJ_n, \\ &= B^TM - B^TMJ_n - J_nM + J_nMJ_n, \quad \text{car} \ B^TJ_n = J_n. \end{aligned}$$

On en déduit :

$$\begin{aligned} tr(B^T(W_nMW_n)) &= tr(B^TM) - tr(B^TMJ_n) - tr(J_nM) + tr(J_nMJ_n), \\ &= tr(B^TM) - tr(MJ_nB^T) - tr(J_nM) + tr((J_n)^2M), \\ &= tr(B^TM) - tr(MJ_n) - tr(J_nM) + tr(J_nM), \ \text{car} \ J_n, B^T \in \mathcal{LC}1, \\ &= tr(B^TM) - tr(MJ_n). \end{aligned}$$

Ainsi,

$$\begin{aligned} \langle\langle M-\overline{M}, B\rangle\rangle &= tr(B^TM) - tr(B^TM) + tr(MJ_n) - 1, \\ &= tr(MJ_n) - 1. \end{aligned}$$

Les matrices J_n et M étant fixées, $tr(MJ_n) - 1$ est une constante. Par suite, on a :

$$\langle\langle M-\overline{M}, B\rangle\rangle = cte, \ \ \forall B \in \mathcal{LC}1.$$

D'où le résultat.

La proposition est ainsi prouvée. ■

Ainsi, on peut dire que

$$\forall \mathbf{M} \in \mathcal{M}_{\mathbf{n}}(\mathbb{R}), \quad \mathcal{P}_{\mathcal{LC}\mathbf{1}}(\mathbf{M}) = \mathbf{W_nMW_n} + \mathbf{J_n}. \tag{3.26}$$

Nous obtenons un résultat qui a été trouvé de deux manières différentes par R. N. KHOURY [80] et GLUNT *et al*. [65]. KHOURY a utilisé une approche purement géométrique (en fait algébrique) tandis que GLUNT *et al.* se sont placés dans un contexte d'optimisation convexe et attachés à la résolution du système de Karush-Kuhn-Tucker correspondant au problème d'optimisation.

3.3.3 Algorithme

Nous avons proposé l'algorithme structuré comme suit :

Algorithme 3.3.1

Initialisation $B^0 = M$
$Q^0 = 0$
Précision ε

Itération
$A^{k+1} = W_n B^k W_n + J_n \quad [= \mathcal{P}_{\mathcal{LC}1}(B^k)]$
$B^{k+1} = (A^{k+1} + Q^k)^+ \quad [= \mathcal{P}_{\mathcal{C}^+}(A^{k+1})]$
$Q^{k+1} = (A^{k+1} + Q^k) - (A^{k+1} + Q^k)^+$

Test d'arrêt *si* $\|A^{k+1} - B^{k+1}\|_F < \varepsilon$ *Stop*
sinon retour à Itération

où M est la matrice que l'on cherche à approcher par une matrice bistochastique.

Cet algorithme est tout simplement une adaptation de l'algorithme (3.3.1) à notre cas. Nous l'avons écrit en tenant compte du fait que l'un de nos convexes est un sous-espace, et qu'il est donc inutile d'en calculer les composantes normales à chaque itération.

Le test d'arrêt est basé sur le fait qu'on doit avoir $\lim_{k\to+\infty} \|A^{k+1} - B^{k+1}\|_F = 0$ (voir Théorème 2.3.1).

3.3.4 Quelques remarques

Dans toute cette partie, nous notons, pour une matrice M donnée de E,

$$\overline{M} = \mathcal{P}_{\mathcal{LC}1}(M) \text{ et } \overline{\overline{M}} = \mathcal{P}_{\mathbb{B}_n}(M).$$

Nous pouvons dès à présent dire un certain nombre de choses sur notre problème d'approximation par des matrices bistochastiques. Compte tenu de la géométrie de $\mathbb{B}_n$, nous allons le considérer comme étant la composée du problème d'approximation sur l'ensemble des matrices **lc1** et, à l'intérieur de ce sous-espace affine du problème d'approximation sur l'orthant positif (cf. figure 5 ci-après).

Puisque sur l'espace $\mathcal{LC}1$, les contraintes $\overline{M}e = e$ et $\overline{M}^T e = e$ sont déjà satisfaites, il reste en fait à s'assurer que $\overline{M}$ a toutes ses composantes positives.

On va distinguer alors les deux situations suivantes :

1. $\overline{M} \geq 0$,
2. $\overline{M} \ngeq 0$.

Cas où $\overline{M} \geq 0$.

Tout d'abord, reprenons le cas $\mathbf{n = 2}$. Il est facile de déduire de l'étude précédente de $\mathbb{B}_2$ que $\mathcal{LC}1$ est la droite (dimension 1) passant par I_2 et P_1 (qui sont définies en (3.4)). Le problème se ramène alors à celui de projeter sur le segment $[I_2; P_1]$, quand l'on sait projeter sur la droite (I_2P_1) sous-jacente. Ainsi,

Proposition 3.3.6 *Si $n = 2$, on a :*

$$\overline{\overline{M}} = \mathcal{P}_{\mathcal{LC}1}(M) = \begin{cases} \overline{M} = W_2 M W_2 + J_2 & \textit{si} \quad \overline{M} \geq 0, \\ I_2 & \textit{si} \quad \overline{M} \ngeq 0 \textit{ et } \|I_2 - \overline{M}\|_F < \|P - \overline{M}\|_F, \\ P_1 & \textit{si} \quad \overline{M} \ngeq 0 \textit{ et } \|I_2 - \overline{M}\|_F > \|P - \overline{M}\|_F. \end{cases}$$

□

La preuve est évidente. Pour $n = 2$, la projection sur $\mathcal{LC}1$ est donc explicite. Et, pour n quelconque, on a une *forme explicite pour certaines matrices*.

En effet, on a la proposition suivante :

Proposition 3.3.7 *Si* $\overline{M} \geq 0$ *alors* $\overline{\overline{M}} = \overline{M}$. □

La preuve est immédiate.

L'hypothèse $\overline{M} \geq 0$ est tout à fait plausible, puisque, par exemple, on vérifie bien que :

$$M = 0 \Rightarrow \overline{M} = J_n \geq 0 \Rightarrow \overline{\overline{M}} = \overline{M}.$$

Rappelons que pour i, j entiers compris respectivement entre 1 et n, on définit les matrices E_{ij} de la base canonique de $\mathcal{M}_n(\mathbb{R})$ par :

$$E_{ij} = (e_{kl})_{k,l} \text{ et } e_{kl} = \begin{cases} 1 & \text{si} \quad (k,l) = (i,j), \\ 0 & \text{sinon.} \end{cases}$$

On a alors :

Proposition 3.3.8 *Si* $M = 0_{\mathcal{M}_n(\mathbb{R})}$ *ou* $M = E_{ij}, \;\; i = 1, \ldots, n, \;\; j = 1, \ldots, n$, *alors*

$$\overline{\overline{M}} = \overline{M} = W_n M W_n + J_n.$$

□

Démonstration

Le cas $M = 0_{\mathcal{M}_n(\mathbb{R})}$ a déjà été évoqué. Pour $M = E_{ij}$, il nous suffit de montrer que pour i, j fixés,

$$\overline{M} = W E_{ij} W + J_n \geq 0.$$

Nous allons tout simplement calculer explicitement les composantes de $\overline{M}$ et vérifier qu'elles sont toutes positives.

On pose : $D = W E_{ij} W = (d_{kl})_{k,l}$.

Par définition, on a :

$$\begin{aligned}
d_{kl} &= \sum_{u=1}^{n} \sum_{v=1}^{n} w_{ku} e_{uv} w_{vl}, \\
&= \sum_{v} w_{kk} e_{kv} w_{vl} + \sum_{u \neq k} \sum_{v} w_{ku} e_{uv} w_{vl}, \\
&= w_{kk} e_{kl} w_{ll} + \sum_{v \neq l} w_{kk} e_{kv} w_{vl} + \sum_{u \neq k} w_{ku} e_{ul} w_{ll} + \sum_{u \neq k} \sum_{v \neq l} w_{ku} e_{uv} w_{vl}, \\
&= (1 - \frac{1}{n})^2 e_{kl} + (1 - \frac{1}{n})(-\frac{1}{n}) \sum_{v \neq l} e_{kv} + (1 - \frac{1}{n})(-\frac{1}{n}) \sum_{u \neq k} e_{ul} + \frac{1}{n^2} \sum_{u \neq k} \sum_{v \neq l} e_{uv}, \\
&= (1 - \frac{1}{n})^2 e_{kl} + \left(\frac{1}{n} - 1\right) \left(\frac{1}{n} (\sum_{v \neq l} e_{kv} + \sum_{u \neq k} e_{ul} \right) + \frac{1}{n^2} \sum_{u \neq k} \sum_{v \neq l} e_{uv}.
\end{aligned}$$

On en déduit :

– si $(k,l)=(i,j)$ alors $d_{ij}=(1-\frac{1}{n})^2$;

– si $k=i,\ \ l\neq j$, $d_{il}=\frac{1}{n}(\frac{1}{n}-1)=\frac{1}{n^2}-\frac{1}{n}$;

– si $k\neq i,\ \ l=j$, $d_{kj}=\frac{1}{n}(\frac{1}{n}-1)=\frac{1}{n^2}-\frac{1}{n}$;

– si $k\neq i,\ \ l\neq j$, $d_{kl}=\frac{1}{n^2}$.

Comme J_n a toutes ses composantes égales à $\frac{1}{n}$, on a : pour $M=E_{ij}$, $\overline{M}=(\overline{e}_{kl})_{k,l}$ tel que :

$$\left\{\begin{array}{lcll}\overline{e}_{ij} & = & (1-\frac{1}{n})^2+\frac{1}{n}, & \\ \overline{e}_{il} & = & \frac{1}{n^2} & l\neq j,\\ \overline{e}_{kj} & = & \frac{1}{n^2} & k\neq i,\\ \overline{e}_{kl} & = & \frac{1}{n^2}+\frac{1}{n} & k\neq i \quad l\neq j.\end{array}\right.$$

Il va de soi qu'on a :

$$\overline{E}_{ij}\geq 0.$$

D'où le résultat. ■

Signalons qu'au passage, nous avons montré que pour $M=(m_{ij})$, on a $\overline{M}=(\overline{m}_{ij})_{i,j}$ avec :

$$\overline{m}_{ij}=(1-\frac{1}{n})^2m_{ij}+\frac{1}{n}(\frac{1}{n}-1)(\sum_{k\neq i}m_{kj}+\sum_{l\neq j}m_{kl})+\frac{1}{n^2}\sum_{k\neq i}\sum_{l\neq j}m_{kl}+\frac{1}{n}.$$

Pour aller plus loin, nous allons essayer de caractériser les matrices M de $\mathcal{M}_n(\mathbb{R})$ qui sont telles que les matrices **lc1** les plus proches d'elles sont en même temps les matrices bistochastiques les plus proches.

Proposition 3.3.9 **(1)** *Soit $M=(a_{ij})\in E$ tel que $\sum_{i,j}a_{ij}\leq 1$.*

Alors,

$$\mathcal{P}_{\mathbb{B}_n}(M)=\mathcal{P}_{\mathcal{LC}1}(M)=W_nMW_n+J_n.$$

(2) *Soit $M\in E$ tel que M puisse s'écrire $M=B+ue^T+ev^T$ avec $B\in\mathbb{B}_n$, $(u,v)\in\mathbb{R}^n\times\mathbb{R}^n$.*

Alors,

$$\mathcal{P}_{\mathbb{B}_n}(M)=\mathcal{P}_{\mathcal{LC}1}(M)=W_nMW_n+J_n.$$

□

Démonstration

La justification du **(2)** est facile. Elle découle directement de la caractérisation (3.17) et de la Proposition 3.3.7.

En ce qui concerne le **(1)**, le résultat découle directement du lemme suivant dû à E. H. ZARANTONELLO [118] :

Lemme 3.3.1 ([118]) *Si P est opérateur de projection dans un Hilbert (par exemple $\mathcal{P}_C$), alors :*

$$\left\|P(\sum_1^k\alpha_ix_i)-\sum_1^k\alpha_iP(x_i)\right\|^2\leq\frac{1}{2}\sum_{i,j=1}^k\alpha_i\alpha_j\langle P(x_i)-P(x_j),(I-P)(x_i)-(I-P)(x_j)\rangle,\tag{3.27}$$

pour toutes familles finies $\{x_i\}_i$ *de vecteurs et* $\{\alpha_i\}_i$ *de réels positifs tels que* $\sum_1^k \alpha_i = 1$.

□

Pour prouver **(1)**, il suffit d'appliquer (3.27) à la décomposition :

$$M = \sum_{ij} a_{ij} E_{ij} + (1 - \sum_{ij} a_{ij}) 0_{\mathcal{M}_n(\mathbb{R})}.$$

■

Cas où $\overline{M} \not\geq 0$

Nous nous intéressons au cas où la matrice $\overline{M} \in \mathcal{LC}1$ la plus proche de M n'est pas bistochastique.

Notre hypothèse de travail est donc :

$$(H) \quad \exists \ (i_0, j_0) \quad \text{tel que} \ \ \overline{m}_{i_0 j_0} < 0.$$

Notre idée est de voir si nous pouvons déduire dans ce cas un résultat intéressant qui puisse nous permettre d'obtenir, dans le cas n quelconque, une expression analogue à la Proposition 3.3.6 et qui soit, bien sur, facilement utilisable.

Pour commencer, nous allons nous intéresser plus précisément à la structure du polytope convexe des matrices bistochastiques $\mathbb{B}_n$. Rappellons que $\mathbb{B}_n$ est l'enveloppe convexe de l'ensemble des matrices de permutations (cf. Théorème 3.1.8).

Proposition 3.3.10 *Soit* $P_i, \ i = 1, \ldots, n!$ *les matrices de permutations d'ordre* n. *On a les propriétés suivantes :*

1. $J_n \in \mathcal{LC}1^{\perp}$;
2. $\|J_n - P_i\|^2 = n - 1, \quad \forall i;$
3. $\|P_i\|^2 = n, \quad \forall i.$

□

Le preuve de ces 3 points est immédiate.

Cette proposition est assez intéressante : elle fait apparaître une structure assez régulière pour $\mathbb{B}_n$.

1. La matrice J_n semble jouer un rôle central dans le polytope $\mathbb{B}_n$, rôle que l'on subodorait puisqu'elle est la seule matrice de $\mathbb{B}_n$ dont toutes les composantes sont égales.
2. Le polytope $\mathbb{B}_n$ est entièrement contenu dans une sphère centrée en J_n et passant par tous les points extrémaux le définissant.

Or, on peut comprendre une projection de la manière suivante : on trace une collection de sphères centrées au point que l'on veut projeter et dont on augmente progressivement le rayon jusqu'à ce qu'on obtienne une sphère tangente à une facette du convexe. le point de contact étant le projeté recherché.

Compte tenu des différentes remarques ci-dessus, il nous apparaît judicieux d'introduire le point suivant de $\mathbb{B}_n$.

♦ **Définition de $\widehat{\mathbf{M}}$.**

Considérons dans $\mathcal{LC}1$ le segment d'extrémités J_n et $\overline{M}$ contenu dans $\mathcal{LC}1$. Puisque $\overline{M} \not\geq 0$, ce segment rencontre la frontière de $\mathbb{B}_n$. Nous notons $\widehat{M}$ cette intersection.

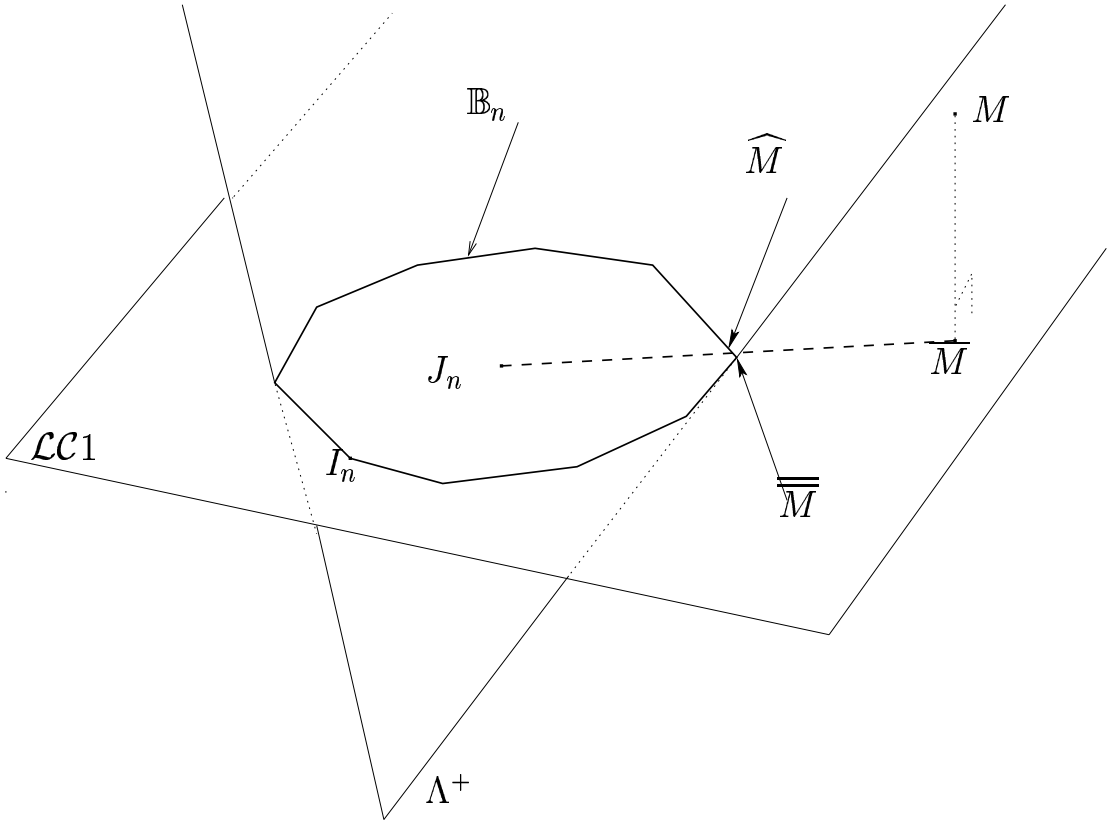


FIG. 3.2 – Illustration de la définition de $\widehat{M}$

• Calcul de $\widehat{M}$.

On a : $\widehat{M} = [J_n, \overline{M}] \cap Fr(\mathbb{B}_n)$.

Comme $\widehat{M} \in [J_n, \overline{M}]$, il existe $\widehat{t} \in [0,1]$ tel que $\widehat{M} = J_n + \widehat{t}(\overline{M} - J_n)$.

Pour trouver $\widehat{M}$, il nous suffit de connaître $\widehat{t}$. Pour cela, il nous suffit de faire une recherche linéaire sur t en partant de J_n dans la direction $\overline{M} - J_n$ tout en gardant positives toutes les composantes des matrices $M_t = J_n + t(\overline{M} - J_n), t \in [0,1]$. La valeur optimale obtenue correspond à $\widehat{M}$.

Plus précisément, $\widehat{t}$ est valeur optimale du problème d'optimisation suivant :

$$(RL) \begin{cases} \max \ t \\ \text{tel que} \\ J_n + t(\overline{M} - J_n) \geq 0 \\ t \in [0,1]. \end{cases}$$

Notons : $D = \overline{M} - J_n = (d_{ij})_{i,j}$.

Alors on montre facilement que :

$$\widehat{t} = -\frac{1}{n d_{i_0 j_0}} \text{ avec } d_{i_0 j_0} = \min\{d_{ij} \mid d_{ij} < 0\}.$$

Ainsi, connaissant $\overline{M}$, il est facile de connaître $\widehat{t}$ donc $\widehat{M}$. Nous faisons alors la conjecture suivante :

Conjecture : *$\overline{\overline{M}}$ et $\widehat{M}$ sont sur la même facette de $\mathbb{B}_n$.*

Si cette conjecture est avérée, l'idée est de se ramener à travailler simplement sur cette facette de $\mathbb{B}_n$, que l'on peut identifier par exemple en exhibant, grâce à l'algorithme de Birkhoff (voir [90]), la combinaison convexe de matrices de permutations qui est égale à M. On pourrait alors en déduire un algorithme exact en calcul, et qui convergerait en un nombre fini (au maximum n) d'itérations pour calculer $\overline{\overline{M}}$. Hélas, tout ceci reste encore à l'état de conjecture et n'a pas été testé numériquement.

3.3.5 Tests numériques

Nous avons appliqué l'algorithme de Boyle-Dykstra ci-dessus (Algorithme 3.3.1) à la résolution du problème d'approximation par des matrices bistochastiques, compte tenu du fait que $\mathbb{B}_n$ est l'intersection du sous-espace $\mathcal{LC}1$ et du cône Λ^+.

Nous avons testé l'algorithme pour différentes matrices. Nous avons obtenu les résultats exprimés par les figures suivantes.

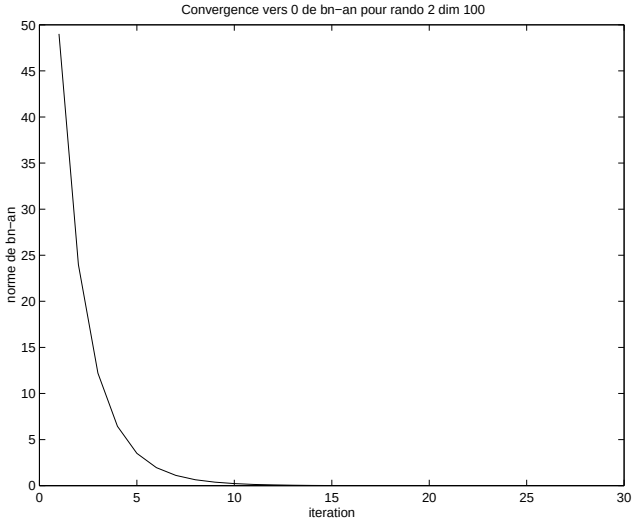


FIG. 3.3 – Convergence de $\|B^k - A^k\|$ pour matrice rando, $n = 100$

La première figure, figure 3.3, représente la courbe de convergence de $\|A^k - B^k\|_F$ vers 0 pour une matrice M de dimension 100 dont les composantes sont générées aléatoirement et dont chaque composante est comprise entre 0 et 1. Ce choix est dicté par le fait que les applications auxquelles nous nous sommes intéressés conduisent à des matrices à approximer de ce type. Nous avons fait la même chose avec une matrice de Hilbert de même dimension (100). Nous obtenons la figure 3.4,. Rappelons que les matrices de Hilbert sont définies par :

$$H = (h_{ij}) \in \mathcal{M}_n(\mathbb{R}) \text{ tel que } h_{ij} = \frac{1}{i+j-1}.$$

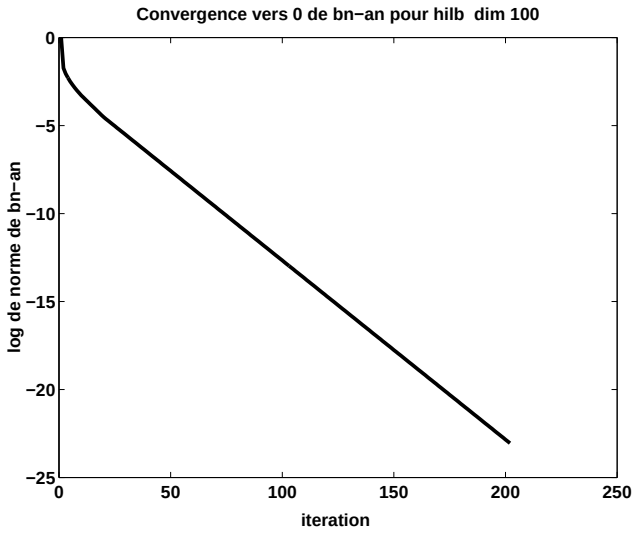


FIG. 3.4 – Convergence de $\ln \|B^k - A^k\|$ pour matrice Hilbert, $n = 100$

Puis, nous avons étudié le comportement de l'algorithme par rapport à la taille de la matrice que l'on veut approcher. Pour des matrices générées aléatoirement, on obtient la figure 3.5 et pour les matrices de Hilbert la figure 3.6.

Les tests numériques que nous présentons ont été réalisé à partir d'un terminal X connectée à un serveur biprocesseur fonctionnant sous Linux et disposant de deux processeurs Penthium III cadencés à 550 Mhz et d'une mémoire vive (RAM) de 512 Mo.

Il apparaît, au vu des exemples que nous avons traités, que l'algorithme converge assez bien, et que le nombre d'itérations n'explose pas lorsqu'on augmente la taille de la matrice traitée. En ce qui concerne les temps de calculs, pour les exemples que nous présentons, il est de l'ordre de la minute. Dès que la taille des matrices dépasse la centaine, l'algorithme prend plus de temps. Mais ceci est finalement peu significatif puisqu'on peut améliorer le temps de calcul en améliorant le calcul d'un produit matriciel que nous effectuons à chaque étape pour la projection sur $\mathcal{LC}1$, ceci compte tenu de la particularité des matrices J_n et W_n. Les résultats que nous avons présentés sont obtenus en faisant un calcul matriciel classique (sans exploiter la structure particulière de J_n et W_n) sous Matlab. Nous en avons tenu compte par

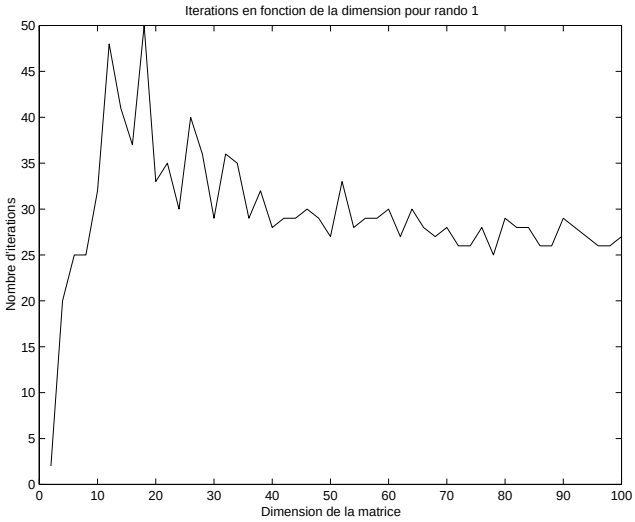


FIG. 3.5 – Nombre d'itérations en fonction de la taille de matrices générées aléatoirement

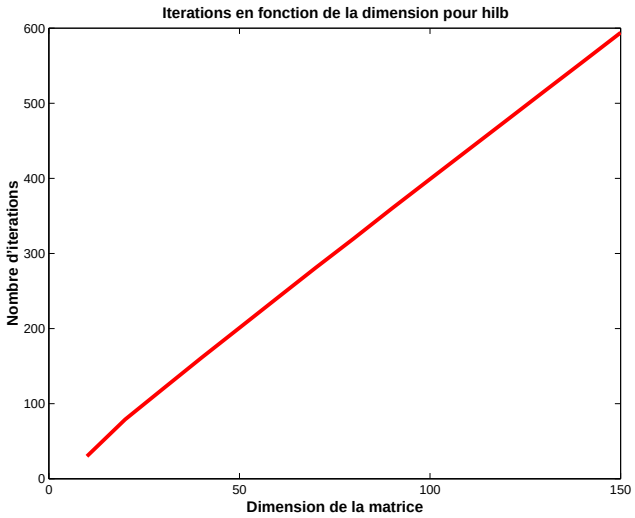


FIG. 3.6 – Nombre d'itérations en fonction de la taille de la matrice de Hilbert

contre pour les tests ci-après qui portent sur des matrices de taille supérieure à 100. De plus, il est possible qu'avec un autre langage, on gagne aussi en temps de calcul.

Nous terminons avec une remarque sur le comportement de l'algorithme pour les matrices creuses. Malheureusement, il semble que l'approximation par matrices ne conserve pas dans l'absolu le caractère creux de la matrice de départ. Ceci est probablement dû au double produit matriciel effectué à chaque projection sur $\mathcal{LC}1$. Il est facile d'anticiper ce résulat, compte tenu de la Proposition 3.3.8 sur la projection des matrices de la base canonique. On peut visualiser cela numériquement : à partir de la matrice E_{11} de dimension 4, la matrice solution

$$\frac{1}{16}\begin{pmatrix} 13 & 1 & 1 & 1 \\ 1 & 5 & 5 & 5 \\ 1 & 5 & 5 & 5 \\ 1 & 5 & 5 & 5 \end{pmatrix}.$$

qui, contrairement à E_{11}, est dense.

Pour illustrer un peu plus cela, nous avons fait des tests pour différentes tailles et différentes densités de matrices. Nous désigons par densité la proportion de composantes non nulles de la matrice. Nous nous intéressons au nombre d'éléments non nuls dans la matrice solution. Nous avons représentés dans les figures 3.7, 3.8 et 3.9 ci-après l'évolution du nombre de composantes non nulles dans la solution que nous obtenons en fonction de la densité de la matrice à approcher A pour des matrices de taille 50, 100, et 150.

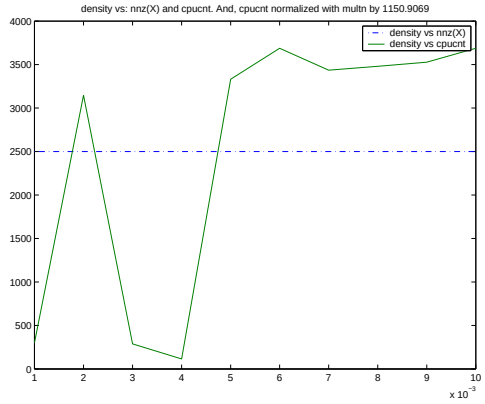


FIG. 3.7 – Temps de calcul et nombre de termes non nuls en fonction de la densité de A pour $n = 50$

Ces remarques confirment notre remarque précédente concernant l'absence de corrélation entre la densité de la matrice à approcher et son approximation bistochastique. On remarque sur les graphiques que les matrices approchées obtenues sont systématiquement pleines, malgré le fait que A était creuse.

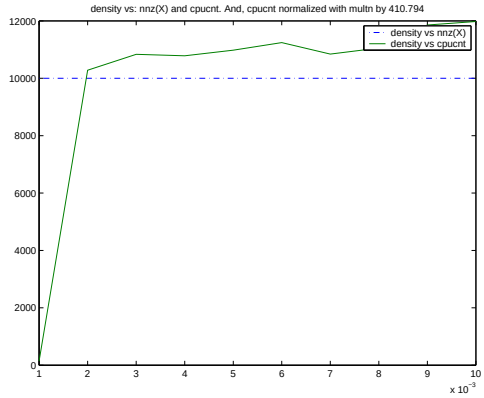


FIG. 3.8 – Temps de calcul et nombre de termes non nuls en fonction de la densité de A pour $n = 100$

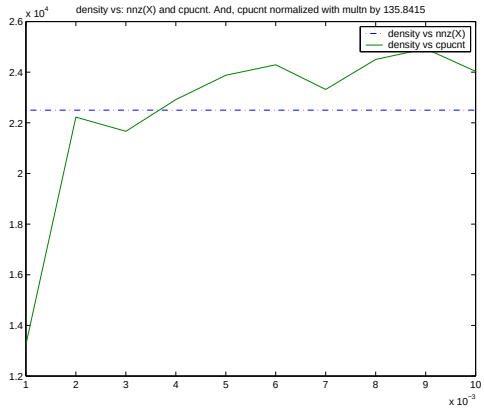


FIG. 3.9 – Temps de calcul et nombre de termes non nuls en fonction de la densité de A pour $n = 150$

3.4 Approximation par algorithme dual

Parallèlement à nos propres travaux consistant en la mise en œuvre de méthodes numériques de résolution du problème d'approximation par matrices bistochastiques en utilisant les projections alternées, d'autres approches de résolution ont été introduites pour ce type de problèmes. Ainsi, dans [88], J. MALICK propose un algorithme de résolution qui utilise la dualité lagrangienne, et qui s'applique à n'importe quel problème d'approximation linéaire conique. Pour des raisons d'unité et de présentation pédagogique, nous présentons ci-dessous l'approche de J. MALICK.

3.4.1 Principe de l'algorithme dual

Rappelons que nous cherchons à résoudre le problème suivant :

$$\begin{array}{rcll} \frac{1}{2}\|A-\overline{X}\|^2 & = & \frac{1}{2}\min & \|A-X\|^2 \\ & & \text{tq.} & \mathcal{A}X=b \\ & & & X\in\mathcal{K} \end{array} \tag{3.28}$$

On commence par une étape de dualisation partielle des contraintes du problème.

Dualité lagrangienne

Sur le problème 3.28, on applique un procédé de relaxation lagrangienne qui dualise uniquement les contraintes affines. Pour des rappels sur les procédés de relaxation lagrangienne, on pourra se referer à [106].

On forme donc la fonction lagrangienne (partielle),

$$L(X,y)=\frac{1}{2}\min\|A-X\|^2-\langle y,\mathcal{A}X-b\rangle,$$

où $y\in\mathbb{R}^m$.

On définit la fonction duale

$$\mathcal{T}(y)=\min_{X\in\mathcal{K}}L(X,y),$$

qui fournit pour chaque valeur de y une borne inférieure de la valeur optimale du problème 3.28. De manière classique, la meilleure de ces bornes est obtenue en résolvant le problème

$$\begin{array}{ll} \sup & \mathcal{T}(y) \\ \text{tq.} & y\in R^m \end{array} \tag{3.29}$$

qui est appelé *problème dual* par opposition au problème 3.28 appelé *problème primal*. On a alors les résultats suivants :

Théorème 3.4.1 *Dans la définition de la fonction duale de $\mathcal{T}(y)=\min_{X\in\mathcal{K}}L(X,y)$:*

1. *la valeur minimale est atteinte pour*

$$X_y=\mathcal{P}_{\mathcal{K}}(A+\mathcal{A}^*y).$$

2. Pour tout $y \in \mathbb{R}^m$, on a :

$$\mathcal{T}(y) = -\frac{1}{2}\|\mathcal{P}_{\mathcal{K}^\circ}(A + \mathcal{A}^* y)\|^2 + \frac{1}{2}\|A\|^2 + \langle y, b\rangle.$$

□

Pour la preuve de ces résultats, on pourra se referer à l'article de Malick [88].

Propriétés de la fonction duale $\mathcal{T}$ et algorithme

On a le théorème suivant (voir [88]) :

Théorème 3.4.2 ([88]) *La fonction duale $\mathcal{T}$ satisfait aux propositions suivantes :*
(i) $\mathcal{T}$ est concave.

(ii) $\mathcal{T}$ est différentiable, et pour tout y dans $\mathbb{R}^m$,

$$\nabla\mathcal{T}(y) = -\mathcal{A}\left[\mathcal{P}_{\mathcal{K}}(A + \mathcal{A}^* y)\right] + b.$$

(iii) $\nabla\mathcal{T}$ est lipschitzienne.
Par suite, $\nabla\mathcal{T}$ est différentiable presque partout.

□

Compte tenu du théorème ci-dessus, le problème dual que l'on a obtenu après relaxation lagrangienne partielle est un problème de maximisation sans contraintes d'une fonction concave, presque partout deux fois différentiable et pour laquelle on dispose d'une forme explicite du gradient. Par suite, le problème dual peut être facilement résolu en utilisant un algorithme d'optimisation convexe sans contraintes (voir [96]). Il est particulièrement adapté à l'usage d'un algorithme de type quasi-Newton.

Puisque c'est le dual qui est résolu et que le gradient dépend aussi des variables du problème primal, nous avons besoin de construire une solution primale à partir d'une solution duale. Pour cela, on a :

Proposition 3.4.3 *Soit y une solution duale. Alors,*

$$X = \mathcal{P}_{\mathcal{K}}(A + \mathcal{A}^* y)$$

est une solution primale □

On montre (voir [88]) au passage qu'il n'y a pas de saut de dualité, c'est-à-dire que la valeur optimale du problème primal coincide avec celle du problème dual. On en déduit l'algorithme suivant :

Algorithme 3.4.1 (Algorithme conique dual) *On part d'une donnée initiale y_0.*
Pour $k = 0, 1, 2, \ldots$
– calculer $X^k = \mathcal{P}_{\mathcal{K}}(A + \mathcal{A}^* y^k)$,
– calculer $\nabla\mathcal{T}(y^k) = -\mathcal{A}X^k + b$,
– calculer $\mathcal{T}(y^k) = \frac{1}{2}\|X^k\|^2 + \langle y^k, b\rangle$,
– faire la mise à jour $y^{k+1} \leftarrow y^k$ par une formule de BFGS,
jusqu'à convergence.

3.4.2 Application à $\mathbb{B}_n$

Nous avons appliqué l'algorithme conique dual de J. MALICK que nous venons de présenter au problème d'approximation par matrices bistochastiques, et nous l'avons comparé à notre algorithme par projections alternées.

Ici on a :

$$A = M, \quad m = 2n, \quad b = [e\ e]'.$$

On considère $\mathbb{R}^m$ sous la forme $\mathbb{R}^n \times \mathbb{R}^n$. Ainsi, $y \in \mathbb{R}^m$ sera écrit sous la forme partitionnée $y = [y_1\ y_2]'$. L'opérateur $\mathcal{A}$ s'identifie à l'opérateur linéaire l que nous avons introduit au paragraphe 3.3.3 (voir justifications de la proposition 3.3.3). On a ainsi :

$$\forall y = [y_1\ y_2]' \in \mathbb{R}^n \times \mathbb{R}^n, \ \ \mathcal{A}^*(y) = y_1 e^T + e y_2^T.$$

Et, l'algorithme s'écrit ici :

Algorithme 3.4.2 (Algorithme conique dual) *On part d'une donnée initiale y_0.*

Pour $k = 0, 1, 2, \ldots$

– *calculer* $X^k = \left[M + y_1^k e^T + e(y_2^k)^T\right]^+$,

– *calculer*

$$\nabla \mathcal{T}(y^k) = \begin{bmatrix} -X^k e + e \\ -(X^k)^T e + e \end{bmatrix},$$

– *calculer* $\mathcal{T}(y^k) = \frac{1}{2}\|X^k\|^2 + \langle y_1^k, e\rangle + \langle y_2^k, e\rangle$,

– *faire la mise à jour $y^{k+1} \leftarrow y^k$ par une formule de BFGS,*

jusqu'á convergence.

Les résultats sont présentés ci-après. Nous avons utilisé l'algorithme de quasi-Newton, *fminunc*, qui est distribué avec Matlab.

Sur la figure 3.10, la courbe en trait simple représente l'évolution du temps de calculs de la solution par l'approche duale en fonction de la dimension de la matrice A. Les temps de calcul de l'algorithme de projections alternées en fonction de la dimension de A sont représentés par la courbe en gras. Enfin, on peut distinguer une courbe en pointillés qui se confond presque avec l'axe des abcisses. Elle représente l'erreur relative en norme de Frobënius entre la solution obtenues par projections alternées et celle obtenue par l'autre approche. Idéalement, cette erreur devrait être nulle. Le fait que la courbe semble se confondre avec l'axe des abcisses est de ce point de vue intéressant. Mais, on peut remarquer en regardant de plus près, que ces normes sont en moyenne de l'ordre de 10^{-2}. Cette moyenne pourrait être améliorée en jouant sur le test d'arrêt de l'algorithme de quasi-Newton utilisé. Pour nos tests, nous avons pris comme tolérance sur la solution la même valeur 10^{-10}.

3.4.3 Approche par points fixes

A partir de l'approche par dualité que nous avons présentée précédemment, on peut décliner une nouvelle approche de résolution de notre problème d'approximation. Cette approche, très récente, est due à BAUSCHKE, KRUK et WOLKOWICZ [22].

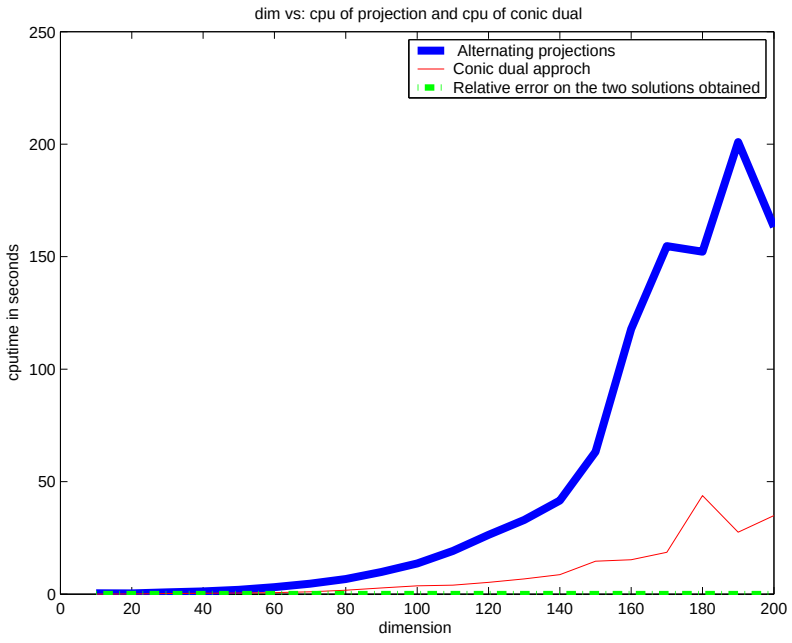


FIG. 3.10 – Comparaison de l'approche duale et des projections alternées

Rappelons que nous avons montré à l'étape précédente que la solution optimale $\overline{X}$ à notre problème est la solution primale associée à la solution optimale $\bar{y}$ du problème dual. Il vient que

Proposition 3.4.4

$$\overline{X} = \mathcal{P}_{\mathcal{K}}(A + \mathcal{A}^* y),$$

avec

$$\mathcal{A}\left[\mathcal{P}_{\mathcal{K}}(A + \mathcal{A}^* y)\right] = b \tag{3.30}$$

□

Quitte à le normaliser (au sens strict) et à modifier b, on peut toujours supposer que $\mathcal{A}$ est tel que

$$\||\mathcal{A}|\| < 1.$$

Moyennant cette hypothèse, $\mathcal{A}$ devient un opérateur contractant. Et, grâce aux propriétés de ces opérateurs, on peut réécrire la condition d'optimalité 3.30 sous la forme d'une condition de points fixes sur un opérateur contractant (dans sa terminologie française [1]). Résoudre le problème d'approximation se ramène alors à chercher un point fixe d'un opérateur (non linéaire) contractant. Nous conseillons [15] pour la définition des opérateurs contractants (au sens anglo-saxon), et des références sur la Théorie des points fixes pour les opérateurs contractants.

Les travaux utilisant cette approche étant encore en cours, nous ne nous étendrons pas plus sur cette partie. Nous renvoyons le lecteur aux travaux (futurs) de BAUSCHKE, KRUK et WOLKOWICZ.

[1] Constante de Lipschitz égale à 1. Dans la terminologie anglaise, une contraction est un opérateur lipschitzien de constante *strictement* comprise entre 0 et 1. Lorsque $k = 1$, on parle de "*nonexpansive operator*".

3.5 Application : Problèmes d'agrégations de préférences

3.5.1 Introduction

Certains problèmes de décision qui se posent en pratique ne peuvent être considérés en ne tenant compte que d'un seul point de vue. On peut citer en exemple les cas d'une société qui doit choisir entre plusieurs projets en tenant compte de différents critères : profit, durée, état du marché, risque, etc. ou celui d'électeurs qui doivent choisir entre différents candidats. Ces situations conduisent à des problèmes dits *d'agrégation de préférences*.

De nombreuses approches existent pour ce problème. Nous proposons ici une modélisation qui permet de représenter les préférences par des matrices dont toutes les composantes sont 0 ou 1. Ces préférences sont agrégées en utilisant une procédure d'agrégation par pondérations. Nous retrouvons ainsi la formulation proposée par Blin [24] en 1976 quand nous considérons les mêmes hypothèses que lui sur les préférences. Celles-ci imposaient aux préférences d'être des relations d'ordre strict et de porter sur la totalité des candidats. Cela lui permettait d'agréger les préférences exprimées en une matrice qui, compte tenu des hypothèses sur les préférences, est *bistochastique*. On ramenait alors le problème à celui de chercher la matrice de permutation la plus proche de cette matrice bistochastique. Cela revient à se placer dans un ensemble convexe compact, le polytope des matrices bistochastiques, et à chercher le point extrémal du convexe le plus proche d'un point donné de cet ensemble. Nous nous sommes donnés dans [108] des hypothèses moins restrictives. Dans un premier temps, cela fait perdre le caractère bistochastique de la matrice agrégeant les préférences. Nous récupérons cette propriété en effectuant une approximation de cette matrice par une matrice bistochastique, en utilisant un algorithme que nous avons mis au point. Cela nous permet de retrouver le même type problème que celui considéré par Blin, qui finalement se ramène à un problème de programmation linéaire ou à un problème de mariages dans un graphe bipartite pondéré (*weighted bipartite matching problem*, en anglais).

3.5.2 Présentation des problèmes d'agrégation de préférences

On considère un ensemble $M = \{1, 2, \ldots, m\}$ de m "votants" qui sont les individus appelés à donner leurs avis, donc à exprimer des préférences sur un ensemble $X = \{1, 2, \ldots, n\}$ de n "objets" que nous appellerons également éléments ou candidats dans la suite. Ces objets peuvent être des candidats à une élection, différents projets d'investissements d'une société, etc. Le votant i $(i = 1, \ldots, m)$ exprime une préférence que nous notons P_i sur l'ensemble des n objets. Cela correspond en général à faire un classement de ces n objets. On souhaite alors *agréger les préférences individuelles exprimées P_i en une préférence collective $\overline{P}$* représentant du mieux possible l'opinion collective. On définit alors :

Définition 3.5.1 *On appelle* **problème d'agrégation de préférences** *le problème*

suivant :

$$(\mathcal{P}) \quad \left\{ \begin{array}{l} \textit{Construire la préférence } \overline{P} \\ \textit{qui soit la plus proche possible} \\ \textit{des } m \textit{ préférences individuelles } P_i \textit{ exprimées.} \end{array} \right. \tag{3.31}$$

Une fois décrit formellement ce problème, se posent immédiatement deux questions :

1. comment (sous quelles formes) représenter les préférences ?
2. suivant quelles procédures ou règles agrège-t-on ces préférences ?

Il va de soi qu'à chaque réponse à ces questions correspond une modélisation et une manière de résoudre ces problèmes. Ces modélisations ont comme point commun qu'elles conduisent en général à un *problème d'optimisation*.

D'une manière générale, les préférences sont représentées par des relations binaires (donc parfois par des graphes) ayant un certain nombre de propriétés exprimant la préférence, l'indifférence et/ou l'incompatibilité entre les "éléments" (voir Monjardet [91], et surtout Vincke [112]). Nous prendrons dans la suite une représentation matricielle pour ces préférences.

La classification des procédures d'agrégation les plus utilisées n'est pas forcément aisée (voir [111], [112]). On peut considérer sommairement deux classes. Une première comprend les méthodes qui consistent à remplacer les différents critères (constitués ici par les différentes préférences exprimées) par un critère unique englobant du mieux possible ces critères. La méthode d'agrégation par pondérations que nous utilisons ici en fait partie. La seconde classe est celle des méthodes (voir [91]) qui consistent à chercher un ordre de préférence recueillant le nombre maximum de suffrages sur toutes les préférences par paires qu'il exprime. On dit que cette règle cherche à maximiser les accords ou minimiser les désaccords entre les différentes préférences exprimées. En ce qui concerne cette règle d'agrégation, on peut se référer à l'article de Monjardet [91] où l'auteur étudie les différentes formulations de problèmes qui correspondent à cette règle qui remonterait à Condorcet en 1789. Pour plus d'informations, nous conseillons au lecteur intéressé de consulter les articles [12], [13], [37], [103], [104], [117], par exemple.

L'objet de ce travail est de proposer une *généralisation de la procédure d'agrégation de Blin* [24]. Toutefois, il nous faut préciser que cette procédure n'est pas très développée en Théorie des choix collectifs. Il n'existerait notamment pas d'axiomatisation de cette procédure. L'étude de la pertinence de cette procédure, la recherche d'une axiomatisation lui correspondant et des éventuels points communs qu'elle possèderait avec d'autres procédures existantes comme le classement par points (voir [104], [117]) sont autant de points importants auxquels il faudrait consacrer son attention. De même, un travail similaire sur la procédure par approximation par matrices bistochastiques que nous présentons ci-après est nécessaire. Mais ceci dépasse le cadre de ce travail, nous n'aborderons donc pas ces thèmes.

3.5.3 Une approche matricielle

Nous proposons maintenant une modélisation du problème d'agrégation de préférences (3.31) dans laquelle les préférences sont représentées par des matrices à composantes $0-1$ qui seront agrégées par pondérations.

À chaque préférence nous associons la matrice P définie par : pour $i = 1, 2, \ldots, n$, et $j = 1, 2, \ldots, n$,

$$P_{ij} = \begin{cases} 1 & \text{si l'élément } i \text{ est classé en } j\text{ème position,} \\ 0 & \text{sinon.} \end{cases} \tag{3.32}$$

Ainsi, les préférences seront réprésentées par des matrices $n \times n$ à composantes 0 et 1 dont les lignes comportent au *maximum une composante non nulle* qui vaut alors 1. En effet, compte tenu des hypothèses que nous avons prises sur les préférences, une matrice P représentant une préférence peut avoir :

- une ligne entièrement nulle : il y a donc incompatibilité, le candidat (ou l'élément) correspondant à la ligne n'est pas classé ;
- une colonne comportant plusieurs 1 : il y a indifférence, on a des candidats ex aequo ;
- une colonne comportant un unique 1 : il y a préférence stricte.

Par exemple, pour un ensemble ordonné $\{a, b, c, d, e\}$ de 5 candidats, la matrice

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

représente la préférence

a premier,
b deuxième,
c pas classé,
d premier ex aequo,
e troisième.

Ces préférences vont être agrégées par pondérations. Cela consiste à attribuer un poids à chaque préférence et à faire la moyenne de ces préférences ainsi pondérées. On se ramène alors à chercher la préférence la plus "proche" de cette somme pondérée.

Définition 3.5.2 *Soit $P_i,\ i = 1, 2, \ldots, m$, m préférences sur un ensemble de candidats X de cardinal n. Soit $\{w_i\}_{i=1,2,\ldots,m}$ une famille de poids positifs tels que $\sum_{i=1}^{m} w_i = 1$.*

On dit que le problème d'agrégation de préférences (3.31) est **agrégé par pondérations** *lorsqu'on le ramène au problème d'approximation suivant*

$$\begin{array}{l} \textit{Trouver la préférence (stricte) } \overline{P} \\ \textit{la plus "proche" (dans un sens à préciser) de } \sum_{i=1}^{m} w_i P_i. \end{array} \tag{3.33}$$

La technique d'agrégation par pondérations, encore appelée méthode de la moyenne pondérée semble être une des premières idées d'agrégation qui ait été proposée (voir [112], [111]). Elle avait l'avantage de ramener le problème à celui de la résolution d'un problème d'optimisation monocritère pour lequel on dispose d'algorithmes de résolutions performants. Elle est néanmoins quelque peu abandonnée ces dernières années parce qu'elle correspond en quelque sorte à un lissage des critères. Et qui dit lissage, dit forcément perte d'informations spécifiques qui peuvent s'avérer importantes. D'autre part, elle n'est manifestement pas adaptée si on a, comme c'est souvent le cas, des critères de nature fondamentalement différentes : des critères qualitatifs et quantitatifs. Néanmoins, nous pensons qu'elle fournit une première solution souvent intéressante dans l'analyse du problème et qui peut servir de point de départ aux autres méthodes proposées (qui sont souvent de nature combinatoire).

Si nous revenons à notre cadre de travail, chaque préférence exprimée est représentée par une matrice P_i. On cherche une préférence stricte $\overline{P}$ qui reflète l'opinion générale, elle est représentée par une matrice de permutation. Le problème d'agrégation de préférences par pondérations (3.33) se ramène au problème d'approximation matricielle suivant :

$$\left\{ \begin{array}{rcll} \| \sum_{i=1}^{m} w_i P_i - \overline{P} \| & = & \min & \| \sum_{i=1}^{m} w_i P_i - P \| \\ & & \text{tel que} & P \text{ matrice de permutation} \end{array} \right. , \tag{3.34}$$

où le fait d'être plus proche, évoqué plus haut en (3.33), est compris au sens de la norme $\| \cdot \|$.

On retrouve sous une forme plus générale une formulation proposée par Blin pour un problème d'agrégation de préférences avec certaines hypothèses sur les préférences, notamment :

- les préférences portent sur tous les éléments : tous doivent être classés ;
- les préférences sont strictes : l'incompatibilité et l'indifférence ne sont pas autorisées.

Sous ces hypothèses, il est facile de voir que *les préférences (strictes) exprimées sont représentées par des matrices de permutation*. Alors, la matrice moyennes pondérées $\sum_{i=1}^{m} w_i P_i$ de ces matrices de permutations est une matrice bistochastique, puisqu'elle apparaît en fait comme une combinaison convexe de matrices de permutation (voir section 2), car $\sum_{i=1}^{m} w_i = 1$ et $w_i > 0$ pour tout i.

Prenons en particulier des poids tous égaux, c'est à dire,

$$\forall\ i = 1, 2, \ldots, m \qquad w_i = \frac{1}{m}.$$

La moyenne pondérée des préférences vaut alors

$$\frac{1}{m} \sum_{i=1}^{m} P_i.$$

Notons :

$$\Pi = \sum_{i=1}^{m} P_i \text{ et } \widetilde{\Pi} = \frac{1}{m} \sum_{i=1}^{m} P_i.$$

Il est facile de voir que pour $l = 1, 2, \ldots, n$, $k = 1, 2, \ldots, n$,

$\Pi_{lk} =$ nombre de fois où le candidat l est classé en $k$ème position.

On retrouve ainsi avec Π la matrice définie par Blin [24] de la manière évoquée ci-dessus (nombre de fois où un candidat est classé dans une position) et dénommé **matrice d'agrément** du problème. Dans ce cas, $\widetilde{\Pi}$ est appelée *normalisée* de la matrice d'agrément.

On se ramène alors à chercher la matrice de permutation la plus proche de la matrice bistochastique $\widetilde{\Pi}$. Cette formulation est celle proposée par Blin. Cet auteur l'appelle méthode de projection sur les sommets (*vertex projection method*, en anglais).

Revenons au cas général. Par analogie, (et abus), avec Blin, nous allons appeler **matrice d'agrément** la moyenne pondérée $\sum_{i=1}^{m} w_i P_i$ des préférences, et la noter Π.

Les hypothèses considérées par Blin avaient le défaut de ne pas prendre en compte des situations qui se produisent souvent en pratique, entre autres :

- erreurs dans les classements, perte de données ;
- possibilité d'avoir des ex aequo, des "objets" non classés (exprimant par exemple de l'incompatiblité, de l'indifférence, etc ...) ;
- possibilité que le nombre de candidats soit connu seulement *a posteriori*, comme nous le verrons dans un exemple plus tard.

Nous nous proposons ici d'affaiblir les hypothèses faites par Blin sur les préférences, de manière à prendre en compte ces situations.

En ce qui concerne le problème (3.34), notons tout d'abord qu'il admet des solutions optimales. En effet, on effectue une minimisation sur un ensemble **fini** de solutions réalisables. L'optimum existe donc et est atteint. Par contre, l'unicité de la solution n'est pas acquise. En fait, comme nous le verrons plus loin, cela est induit par le fait qu'un programme linéaire n'a pas forcément une solution optimale unique.

Pour la résolution du problème (3.34), nous proposons un schéma en deux phases. Cette séparation en deux est motivée entre autres par le désir de résoudre le problème en utilisant des outils déjà existants. Une fois construite la matrice d'agrément Π,

Phase 1 : on recherche la matrice bistochastique $\Pi_{\mathbb{B}_n}$ la plus proche de Π en utilisant l'algorithme de projections alternées évoqué en section 2,

Phase 2 : on met en œuvre la méthode de projection sur les sommets ("vertex projection method") de Blin [24] pour rechercher la matrice de permutation la plus proche de $\Pi_{\mathbb{B}_n}$.

3.5.4 Quelques exemples

Nous avons appliqué le schéma de résolution par étapes suivant :

1 On construit la matrice d'agrément par moyenne pondérées. On obtient une matrice Π^{norm} à composantes comprises entre 0 et 1, mais qui n'est pas bistochastique ;

2 *On calcule la matrice bistochastique la plus proche de* Π *en utilisant l'algorithme défini en section 2. On obtient la matrice* $\Pi_{\mathbb{B}_n}$ *bistochastique.*

3 On résout le problème min $d(P, \Pi_{\mathbb{B}_n})$, P matrice de permutation, où d est la distance induite par la norme de Fröbenius.

Nous avons considéré, dans tous les tests numériques que nous présentons ci-après, des poids tous égaux (à $\frac{1}{m}$).

a) Résolution de l'étape 3

Nous revenons sur l'étape 3 où on cherche la matrice de permutation la plus proche d'une matrice bistochastique. On cherche à résoudre le problème d'approximation :

$$\left\{ \begin{array}{rcll} \|\Pi_{\mathbb{B}_n} - \overline{P}\| & = & \min & \|\Pi_{\mathbb{B}_n} - P\| \\ & & \text{tel que} & P \text{ matrice de permutation.} \end{array} \right. \tag{3.35}$$

C'est un problème d'optimisation convexe en variables $0-1$. Pour le résoudre, on a deux stratégies.

Programmation linéaire

En nous souvenant du développement du carré de la norme dans un espace de Hilbert, la fonction-objectif du problème (3.35) s'écrit :

$$\|P - \Pi_{\mathbb{B}_n}\|^2 = \|P\|^2 - 2\langle\langle P, \Pi_{\mathbb{B}_n}\rangle\rangle + \|\Pi_{\mathbb{B}_n}\|^2 \tag{3.36}$$

Or, comme P est une matrice de permutation, on a :

$$\|P\|^2 = n, \quad \text{pour toute matrice } P \text{ de permutation.}$$

Minimiser la quantité $\|P - \Pi_{\mathbb{B}_n}\|$ revient donc (quitte à considerer le carré de la norme) à **maximiser** le produit scalaire : $\langle\langle P, \Pi_{\mathbb{B}_n}\rangle\rangle$. On se ramène ainsi à une fonction-objectif linéaire.

D'autre part, l'ensemble des points réalisables du problème, est l'ensemble des matrices de permutations. C'est donc l'ensemble des points extrémaux du polytope convexe des matrices bistochastiques. Or, optimiser un critère linéaire sur l'ensemble des points extrémaux d'un polytope peut se ramener à optimiser le même critère sur le polytope tout entier, puiqu'on sait (voir [97]) qu'il existe un point extrémal solution d'un tel problème. Il suffit donc par exemple de le résoudre en utilisant la méthode du simplexe qui se termine toujours en un point extrémal.

Ainsi, l'étape 3 revient à résoudre le problème de programmation linéaire en variables $0-1$:

$$\left\{ \begin{array}{rcll} \langle\langle \Pi_{\mathbb{B}_n}, \overline{P}\rangle\rangle & = & \max & \langle\langle \Pi_{\mathbb{B}_n}, P\rangle\rangle \\ & & \text{tel que} & P \in \mathbb{B}_n, P \text{ de permutation,} \end{array} \right. \tag{3.37}$$

que l'on résout (ou plutôt sa *relaxation continue*) par la méthode du simplexe de manière à en obtenir une solution extrémale, c'est-à-dire une matrice de permutation.

Optimisation combinatoire

En pratique, pour résoudre le problème linéaire (3.37), on résout sa relaxation continue qui est le même problème dans lequel on a relaxé la contrainte stipulant que P doit être à composantes entières (0 et 1). Le fait d'utiliser la méthode du simplexe permet cela. Si l'on ne fait pas cette relaxation, notons P_{ij} les composantes de la matrice P et Π_{ij} celles de $\Pi_{\mathbb{B}_n}$. Alors le problème (3.37) s'écrit :

$$\left\{\begin{array}{rl} \max & \sum_{i,j=1}^{n} \Pi_{ij} P_{ij} \\ \text{tel que} & \sum_{j=1}^{n} P_{ij} = 1, \quad \forall i \\ & \sum_{i=1}^{n} P_{ij} = 1 \quad \forall j \\ & P_{ij} \geq 0, \quad \forall i, j \\ & P_{ij} = 0 \text{ ou } 1, \quad \forall i, j. \end{array}\right. \tag{3.38}$$

On reconnaît ici un exemple du "problème de mariages dans un graphe bipartite pondéré", *weighted bipartite matching problem* en anglais, (voir [97]). On est donc ramené à un problème d'optimisation dans un graphe, qui dans un certain sens, peut être vu comme un problème d'affectation de tâches (*assignment problem*, en anglais).

On peut donc mettre en œuvre, pour résoudre (3.38), des méthodes d'optimisation combinatoire existantes, de complexité polynomiale. Nous avons implémenté une de ces méthodes, notamment la méthode dite hongroise (*Hungarian method*, en anglais : voir [97]) pour les problèmes d'affectation. Cette méthode devrait produire un résultat plus exact (notamment pour trouver les composantes entières 0 et 1), et il a été prouvé qu'elle résout le problème exactement en $O(n^3)$ opérations arithmétiques.

b) Tests numériques

Nous avons testé l'algorithme sur différentes gammes de tests. Nous en présentons ici deux. Dans tous ces exemples, nous avons pris des poids tous égaux à $\frac{1}{m}$. L'étape d'approximation par matrices bisochastiques est résolue en utilisant l'algorithme de projections alternées. De plus, dans tous les tests présentés ci-après, l'étape 3 a été résolue par programmation linéaire. Nous avons utilisé pour le premier exemple deux codes de programmation linéaire. Le premier est le code *linprog* qui fait partie de la distribution classique de Matlab. Le second, dû à H. WOLKOWICZ[2], est un code basé sur la méthode du simplexe programmé sous Matlab. Nous nous sommes contentés de *linprog* pour le second.

Exemple avec perte de données

Nous avons considéré comme première situation, celle où des pertes d'informations sur les données auraient eu lieu. Dans tous les cas où il manquait des informations dans les préférences exprimées, nous avons supposé que ce manque exprimait une incompatibilité.

[2]Code disponible à l'url http ://orion.math.uwaterloo.ca/~hwolkowi

Nous avons considéré l'ensemble $X = \{a, b, c, d, e\}$ de $n = 5$ candidats, pour lequel les $m = 6$ préférences suivantes sont exprimées :

$$P_1 \begin{cases} a & \text{premier,} \\ b & \text{quatrième,} \\ c & \text{troisième,} \\ d & \text{pas classé,} \\ e & \text{pas classé.} \end{cases}, \quad P_2 \begin{cases} a & \text{premier,} \\ b & \text{quatrième,} \\ c & \text{deuxième,} \\ d & \text{troisième,} \\ e & \text{cinquième.} \end{cases}, \quad P_3 \begin{cases} a & \text{deuxième,} \\ b & \text{quatrième,} \\ c & \text{premier,} \\ d & \text{troisième,} \\ e & \text{cinquième.} \end{cases},$$

$$p_4 \begin{cases} a & \text{pas classé,} \\ b & \text{premier,} \\ c & \text{pas classé,} \\ d & \text{quatrième} \\ e & \text{cinquième.} \end{cases}, \quad p_5 \begin{cases} a & \text{troisième} \\ b & \text{deuxième,} \\ c & \text{cinquième,} \\ d & \text{quatrième,} \\ e & \text{premier.} \end{cases}, \quad p_6 \begin{cases} a & \text{troisième,} \\ b & \text{pas classé,} \\ c & \text{deuxième,} \\ d & \text{cinquième,} \\ e & \text{pas classé.} \end{cases}$$

On obtient la matrice d'agrément suivante :

$$\Pi = \frac{1}{6} \begin{pmatrix} 2 & 1 & 2 & 0 & 0 \\ 1 & 1 & 0 & 3 & 0 \\ 1 & 2 & 1 & 0 & 1 \\ 0 & 0 & 2 & 2 & 1 \\ 1 & 0 & 0 & 0 & 3 \end{pmatrix}.$$

La matrice bistochastique obtenue avec un critère d'arrêt $\varepsilon = 10^{-20}$ après approximation est :

$$\overline{\Pi}_{\mathbb{B}_n} = \frac{1}{6} \begin{pmatrix} .3600 & .2267 & .3600 & .0267 & .0267 \\ .1933 & .2267 & .0267 & .5267 & .0267 \\ .1933 & .3933 & .1933 & .0267 & .1933 \\ .0267 & .0600 & .3600 & .3600 & .1933 \\ .2267 & .0933 & .0600 & .0600 & .5600 \end{pmatrix}.$$

La matrice de permutation optimale obtenue est alors :

$$\overline{P} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Ceci nous donne comme classement agrégé :

$$\overline{P} \begin{cases} a & \text{premier,} \\ b & \text{quatrième,} \\ c & \text{deuxième,} \\ d & \text{troisième,} \\ e & \text{cinquième.} \end{cases}$$

Signalons que nous avons construit cet exemple en modifiant un exemple proposé par Blin. L'ordre agrégé que nous avons obtenu ici est le même que celui obtenu par Blin qui avait, lui, des préférences portant sur tous les candidats à chaque fois. Cette remarque, quoique surprenante, n'est aucunement significative : on peut obtenir une toute autre solution optimale. Ceci montre bien qu'il n'y a pas unicité des solutions.

Exemple avec nombre de candidats connu a posteriori

Nous proposons maintenant un exemple dans lequel le nombre de candidats m n'est pas défini à l'avance. Cet exemple est tiré d'un magazine de football *Onze Mondial*[3], ce qui est une illustration, selon nous, du fait que les mathématiques peuvent s'appliquer dans presque tous les domaines de la vie, même les plus insoupçonnés.

La situation est la suivante : après une journée de championnat de football, on demande à un collège de 11 journalistes (qui représentent donc les votants) de désigner (classer) chacun exactement 11 joueurs qu'ils considèrent (dans l'ordre) comme les meilleurs. On cherche à partir de ces onzes classements exprimés à établir le classement général des onze meilleurs joueurs de la journée.

Ainsi, on est devant un problème dans lequel on ne connaît pas a priori le nombre de candidats sur lesquels les préférences seront exprimées. Ce nombre sera connu seulement une fois les préférences exprimées. On sait seulement qu'il va varier entre 11 et 121. De par cette nature, ce type de problème ne peut pas vérifier les hypothèses de Blin. Cela justifie a posteriori les motivations de notre travail. Dans l'exemple ci-après, le nombre de candidats est finalement $n = 38$.

Pour représenter graphiquement les matrices, nous traçons le graphe 3D de la fonction définie par

$$(i, j) \mapsto M_{ij}$$

On obtient une matrice d'agrément représentée par la Figure 3.11.

La matrice de permutation que nous obtenons est illutrée par la Figure 3.12.

Concernant cette dernière figure, nous aurions dû visualiser 38 pics uniquement, tout le reste de la surface étant plat. La différence que nous observons est due au critère d'arrêt que nous avons utilisé. Toutefois, elle est suffisante pour nous, puisque notre but est d'obtenir un classement des onzes premiers.

Nous avons comparé le classement que nous avons obtenus avec celui obtenu dans le journal. Celui-ci a été établi en utilisant la fonction de choix social de Borda (voir [104], [117]). Ceci consiste à attribuer un joueur 11 points à chaque fois qu'il est classé premier, 10 points s'il est second, et ainsi de suite. Le classement est effectué après cumul des points obtenus par chaque jour, de celui qui en a le plus (classé premier) à celui qui en a le moins. Seuls les onze premiers du classement sont pris en compte. Dans les résultats nous avons obtenus, nous avons 6 joueurs classés aux mêmes positions que dans le classement obtenu par Borda.

[3]Disponible dans tous les kiosques à journaux. L'exemple que nous proposons se trouve dans le numéro de décembre 2001.

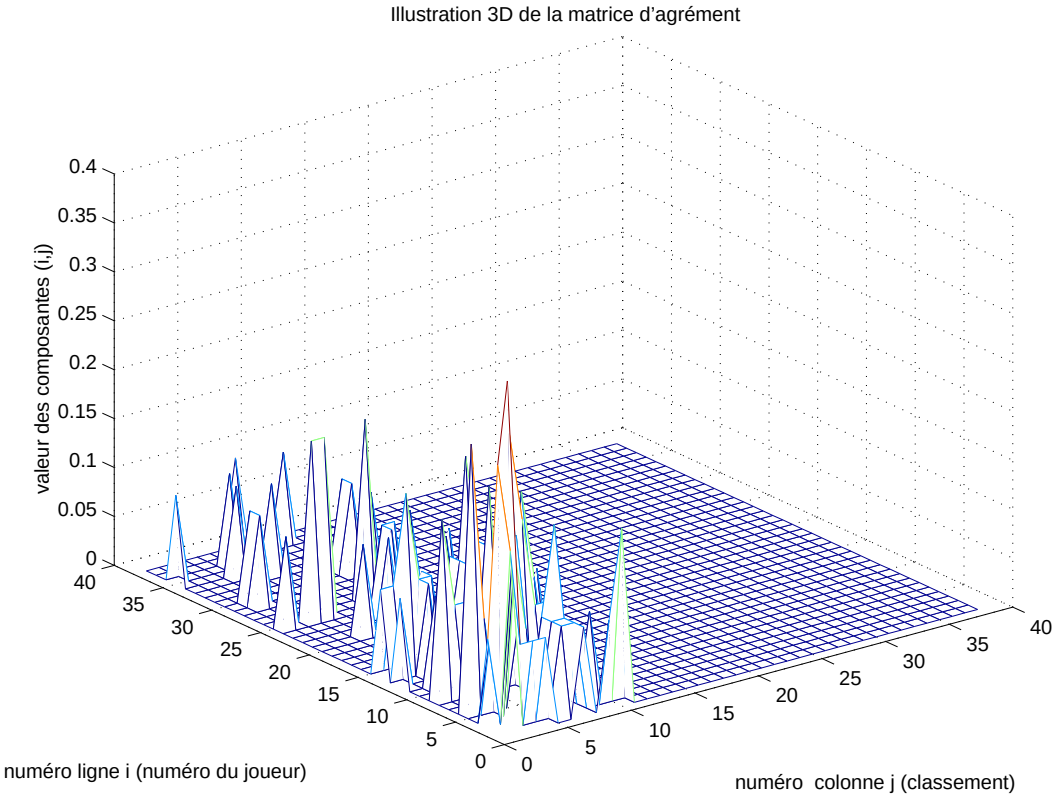


FIG. 3.11 – Illustration 3D de la matrice d'agrément

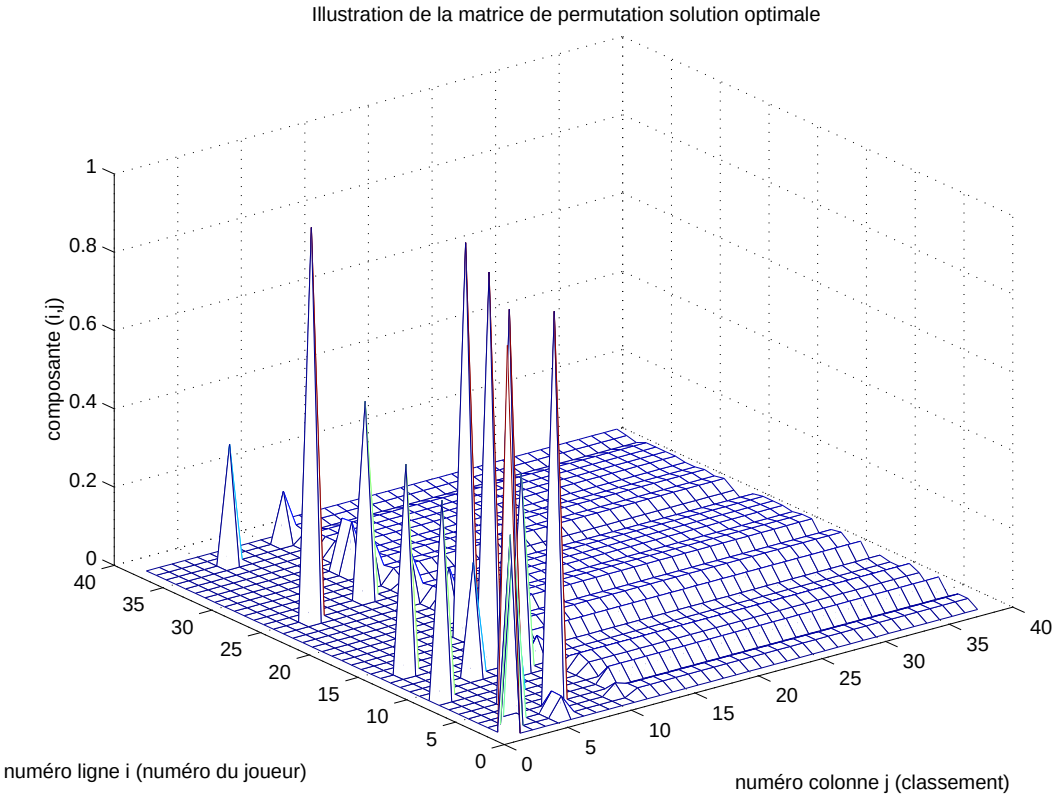


FIG. 3.12 – Illustration 3D de la matrice de permutation optimale obtenue

Nous avons étudié précédemment le problème classique d'agrégation de préférences. D'une part, à partir d'une modélisation matricielle des préférences, nous avons proposé une formulation mathématique dont nous avons montré qu'elle généralise la formulation qu'avait proposée Blin [24] sous certaines hypothèses que nous affaiblissons donc au passage. D'autre part, nous proposons un schéma de résolution de notre formulation dans lequel nous utilisons une application du problème d'approximation par des matrices bistochastiques. Cela nous permet de terminer la résolution par celle d'un programme linéaire.

Une suite naturelle de ce travail consisterait, dans un premier temps, à continuer la mise en œuvre numérique des algorithmes d'optimisation combinatoire que nous avons évoqués comme autre possibilité de terminer la résolution que nous avons proposée. Nous souhaitons aussi pouvoir tester ce schéma sur des problèmes concrets issus de la pratique. Une perspective plus générale consiste à aborder l'axiomatisation de la procédure de Blin, à étudier la pertinence de la procédure d'approximation par matrices bistochastiques que nous avons présentée, et surtout à établir les liens qui peuvent exister entre ces procédures et d'autres qui existent en Théorie des choix collectifs.

3.6 Conclusion

Nous venons d'étudier le problème d'approximation par des matrices bistochastiques. Il ressort de cette étude que pour une matrice donnée M, il existe une et une seule matrice la plus proche de M. Cette matrice possède une caractérisation qui, malheureusement, ne peut permettre d'obtenir une formule "explicite" de cette matrice bistochastique, sauf dans certains cas particuliers que nous avons étudiés. Cela étant, nous avons proposé différentes mises en œuvre algorithmiques qui permettent de calculer cette approximation. Nous avons appliqué ces algorithmes à la résolution de problèmes d'agrégation de préférences. Nous avons ainsi pu proposer une généralisation à la procédure d'agrégation proposée par Blin [24].

L'algorithme par projections alternées présente l'avantage d'être élégant et simple à programmer. Il suffit de décomposer le convexe $\mathbb{B}_n$ des matrices bistochastiques sous la forme d'une intersection de convexes et de savoir explicitement projeter sur ces convexes. L'algorithme conique dual peut lui aussi être considéré comme "simple" puisque la partie difficile en termes de programmation peut être évitée en utilisant des codes d'optimisation convexe sans contraintes préexistants. A priori, il devrait être plus efficace que l'algorithme de projections puisqu'on dispose pour lui explicitement des informations du premier ordre (gradient) et d'au moins une partie des informations du second ordre (la hessienne existe presque partout, etc...) tandis que l'approche par projections est plutôt une méthode de type sous-gradients. Nous l'avons constaté sur les différents tests que nous avons effectués avec le code *fminunc* de Matlab. Toutefois, cette différence de performance est très liée à la nature du code d'optimisation convexe sans contraintes utilisé.

On peut dire, en résumé que nous avons abordé, jusquà présent notre problème d'approximation linéaire conique, soit d'un point de vue totalement primal (projection alternées), soit d'un point de vue totalement dual (approche conique duale).

Il existe la possibilité d'aborder le problème d'un point de vue mixte primal dual. Cette approche est possible, notamment au travers des algorithmes de type points intérieurs que nous introduisons au prochain chapitre.

Chapitre 4

Optimisation sous contraintes de semi-définie positivité

Dans ce chapitre, nous présentons les problèmes dits d'optimisation sous contraintes de semi-définie positivité, encore appelés problèmes d'optimisation SDP ou problème SDP. Cette dernière appelation est une conséquence de la terminologie anglaise *Semi Definite Programming*. L'étude de ce genre de problèmes a connu un fantastique regain d'intérêt depuis les années 90, entres autres parce que l'on a disposé depuis d'algorithmes efficaces permettant de les résoudre : les algorithmes de points intérieurs.

4.1 Problèmes d'optimisation sous contraintes de semi-définie positivité

Les problèmes d'optimisation sous contraintes de semi-définie positivité apparaissent comme une généralisation des problèmes de programmation linéaire. Nous ferons donc très souvent le parallèle entre ces deux types de problèmes. Pour de plus amples détails, nous conseillons aux lecteurs intéressés le récent *Handbook of semidefinite programming* [115]

4.1.1 Définition

Dans toute la suite de ce chapitre, nous nous supposerons, sauf indication contraire, placés dans l'espace euclidien $(\mathcal{S}_n, \langle \cdot, \cdot \rangle)$ muni du produit scalaire

$$\langle A, B \rangle = tr(AB).$$

Définition 4.1.1 *On appelle problème d'optimisation sous contraintes de semi-définie positivité le problème suivant :*

$$\begin{array}{ll} \min & f(X) \\ \textit{t.q.} & \langle A_i, X \rangle = b_i, \quad \forall i = 1, \ldots, m, \\ & X \succeq 0, \end{array} \tag{4.1}$$

où X est une matrice symétrique, f est une fonction convexe de X. Le vecteur $b = (b_1, \ldots, b_n)^T$ de $\mathbb{R}^m$ et les matrices symétriques $(A_i)_{i=1,\ldots,m}$ sont des paramètres donnés du problème.

Un problème SDP est donc un problème d'optimisation convexe.

La définition que nous avons donnée ci-dessus n'est pas vraiment la définition habituelle qui est donnée pour les problèmes SDP. Dans celles-ci, la fonction-objectif est une fonction affine :

$$f(X) = \langle C, X \rangle \tag{4.2}$$

où C est une matrice symétrique donnée. Nous avons pris le parti de donner plutôt la définition 4.1.1 sous une forme plus générale pour bien faire le lien avec les problèmes d'approximation matricielle qui apparaissent directement sous la forme (4.1). En effet, ces problèmes sont en général de la forme (4.1) avec comme fonction-objectif la fonction

$$f(X) = \frac{1}{2}\|A - X\|^2 \tag{4.3}$$

Ceci étant, dans toute la suite, lorsque nous parlerons de problème SDP, nous considérerons sauf indication contraire le problème (4.1) avec la fonction-objectif linéaire (4.2). On peut en effet souvent ramener le problème (4.1) à un problème linéaire (ce sera le cas pour nous), par passage à l'epigraphe notamment, comme nous allons le voir au prochain chapitre.

On peut remarquer le lien entre un programme linéaire et un problème d'optimisation linéaire sous contraintes de semi-définie positivité. Ce dernier problème est en fait une généralisation des programmes linéaires. Il suffit pour le voir de se restreindre à ne considérer que des matrices *diagonales* dans le problème ((4.1)-(4.2).)

Dans la définition 4.1.1, on peut remplacer les contraintes $\langle A_i, X \rangle = b_i,\ \forall i = 1, \ldots, m$ par la contrainte multidimensionnelle unique :

$$\mathcal{A}X = b$$

où $\mathcal{A} : \mathcal{S}_n \to \mathbb{R}^m$ est l'opérateur linéaire défini par

$$\mathcal{A}X = (\langle A_i, X \rangle)_{i=1,\ldots,m.}$$

Les problèmes SDP, ainsi que leur généralisation aux fonction-objectifs convexes, sont des cas particuliers de problèmes plus généraux de la forme :

$$\begin{array}{ll} \min & f(x) \\ \text{t.q.} & g(x) \succeq_{\mathcal{K}} 0, \end{array} \tag{4.4}$$

où $\mathcal{K}$ est un cône convexe fermé, et f et g sont des fonctions appropriées. La relation d'ordre $\succeq_{\mathcal{K}}$ est la même que celle définie au premier chapitre. Ces problèmes sont appelés **problèmes d'optimisation conique** (*cone programming problems*), et ont notamment été étudiés par SHAPIRO [102].

4.1.2 Motivations et Historique

Nous faisons un petit aparté sur les motivations de l'étude de ces problèmes SDP, qui n'est devenue que très récemment un axe de recherche mathématique à part entière.

Avant les années 90, lorsque l'on cherchait à modéliser des situations pratiques réelles, ou que l'on cherchait à approximer numériquement des problèmes compliqués, on utilisait presque sytématiquement les modèles linéaires. Ceci est dû au fait que l'on disposait depuis les années 40 d'algorithmes efficaces de résolution dans les cas linéaires. IL s'agit notamment de l'algorithme du simplexe [97] qui avait l'avantage d'être robuste et de converger en un nombre finis d'itérations, même si on sait qu'il n'avait pas une complexité polynomiale. Puis, grâce entre autres aux travaux de KARMARKAR [79] dans les années 80, sont apparues les méthodes de points intérieurs qui se sont avérées être plus efficaces que le simplexe : ils permettent de résoudre des problèmes de plus grande taille, en un nombre d'itérations indépendant de la dimension du problème, ils sont très rapides, et ont une complexité polynomiale.

Depuis les années 90, grâce notamment aux travaux fondateurs de ALIZADEH [5], NEMIROVSKI, NESTEROV [94] en 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 *mutatis mutandis* aux problèmes SDP. Une des conséquences est que l'on a ainsi pu résoudre par exemple des approximations quadratiques (modèles quadratiques) de problèmes complexes aussi efficacement qu'on le faisait pour les approximations linéaires.

Il a résulté de tout cela un grand nombre de domaines dans lesquels les problèmes SDP ont trouvé des applications. Compte tenu du nombre et de la variété de ces domaines d'applications, il nous est impossible d'en faire ici une liste exhaustive. De plus, de nombreux écrits existent qui répertorient d'une manière que nous ne saurions égaler ici, les différents champs d'applications de l'optimisation SDP. Nous citerons quand même comme champ d'applications :

L'optimisation combinatoire [115], [114] Les relaxations SDP sont utilisées en lieu et place de la relaxation linéaire (ou continue) pour obtenir de bonnes bornes pour les problèmes d'optimisation en variables entières. Contrairement à la relaxation linéaire qui consiste à résoudre le problème en "oubliant" les contraintes d'intégrités (celles qui imposent aux variables d'avoir des valeurs entières), la relaxation SDP consiste exprimer ces contraintes d'intégrité sous la forme de contraintes quadratiques qui sont dualisés. En utilisant notamment le concept de *contraintes cachées* en optimisation quadratique (voir ci-après), on se ramène à un problème dual SDP dont la résolution fournit une borne pour la valeur optimale du problème. Cette borne SDP est en général au moins aussi bonne que celle obtenue par relaxation linéaire, et elle peut être très souvent substantiellement meilleure.

L'optimisation non linéaire (non convexe) Jusqu'à ces dernières années, une des

manières les plus efficaces de résoudre des problèmes non convexes d'optimisation était d'appliquer la programmation quadratique successive (PQS). Celle-ci consitait à résoudre itérativement une suite de problèmes quadratiques convexes (faciles à résoudre) qui sont des approximations du problème de départ obtenues en prenant notamment les développements de Taylor de la fonction-objectif (à l'ordre 2) et des contraintes (ordre 1) dans un voisinage du point courant. La même idée a été reprise pour construire itérativement des suites de problèmes SDP obtenus grâce aux développements de Taylor, aux méthodes de région de confiance, ou aux méthodes de Lagrangien augmenté. On pourra se referer aux travaux de WOLKOWICZ *et al.* (voir [61]), à ceux de APKARIAN, FARES, NOLL (voir [56],[57], [58]) pour des problèmes venant de la commande robuste en Automatique, entre autres.

On pourra se referer à [115] pour plus d'informations sur d'autres applications des problèmes SDP.

4.1.3 Etude des problèmes SDP

Nous commençons par quelques remarques sur la géométrie des ensembles réalisables des problèmes SDP.

a) Géométrie de l'optimisation SDP

Nous désignons par ensemble réalisable d'un problème d'optimisation l'ensemble des points qui satisfont aux contraintes du problème. Les points pour lesquels la valeur optimale du problème est atteinte forment l'ensemble optimal du problème.

Les ensembles réalisables des problèmes de programmation linéaire sont en général des polyèdres ou polytopes convexes. Une grande partie du succès de la programmation linéaire provient des propriétés géométriques de ces polyèdres (ou polytopes). La plupart de ces propriétés s'étendent aux ensembles réalisables des problèmes SDP, même si ceux-ci sont de nature parfois spectaculairement différentes, notamment en termes de leur frontière. Ceci est dû entre autre aux propriétés algébriques, et en termes d'Analyse convexe, des matrices carrées symétriques réelles, du cône des matrices semi-définie positives, etc. Pour de plus amples informations sur ces différents points, nous conseillons les articles du handbook [115].

b) Dualité et Optimalité

De la même manière que pour les programmes linéaires, les problèmes SDP sont en général abordés sous l'angle de la dualité. Rappelons que nous nous intéressons au problème

$$\text{(PSDP)} \qquad \begin{array}{ll} \min & \langle C, X \rangle \\ \text{t.q.} & \mathcal{A}X = b, \\ & X \succeq 0. \end{array} \tag{4.5}$$

On applique un schéma de dualité classique (voir [77]) au problème (PSDP). On associe à la contrainte $\mathcal{A}X = b$ la variable duale $y \in \mathbb{R}^m$. On forme alors la

fonction lagrangienne

$$\begin{aligned} L(X,y) &= \langle C, X\rangle + y^T(b - \mathcal{A}X) && (4.6)\\ &= y^T b + \langle C, X\rangle - y^T\mathcal{A}X && (4.7)\\ &= y^T b + \langle C, X\rangle - \langle \mathcal{A}^* y, X\rangle && (4.8)\\ &= y^T b + \langle C - \mathcal{A}^* y, X\rangle. && (4.9) \end{aligned}$$

On en déduit la fonction duale

$$\mathcal{T}(y) = \min_{X\succeq 0} \left(y^T b + \langle C - \mathcal{A}^* y, X\rangle \right). \tag{4.10}$$

Ce problème n'a de solution que si $C - \mathcal{A}^* y \succeq 0$. En effet, si tel n'est pas le cas, il est possible de trouver un $X \succeq 0$ tel que la quantité $\langle C - \mathcal{A}^* y, X\rangle$ soit aussi négative que l'on veut. Le minimum ne peut alors être que $-\infty$. Cette contrainte $C - \mathcal{A}^* y \succeq 0$ est en fait une contrainte inhérente au problème de minimisation (4.10) qui n'apparaît pas explicitement. On parle alors de *contraintes cachées*.

En introduisant la nouvelle variable duale $Z = C - \mathcal{A}^* y$, la fonction duale devient

$$\mathcal{T}(y) = y^T b + \min_{X\succeq 0} \langle Z, X\rangle, \tag{4.11}$$

avec

$$\min_{X\succeq 0} \langle Z, X\rangle = \begin{cases} 0 & \text{si } Z \succeq 0, \\ -\infty & \text{sinon.} \end{cases}$$

On peut alors montrer que le problème dual s'écrit :

$$\text{(DSDP)} \qquad \begin{array}{ll} \max & b^T y \\ \text{t.q.} & \mathcal{A}^* y + Z = C, \\ & Z \succeq 0. \end{array} \tag{4.12}$$

Notons que puisque $\mathcal{A}X = (\langle A_i, X\rangle)_i$, on a :

$$\mathcal{A}^* y = \sum_{i=1}^{m} y_i A_i. \tag{4.13}$$

On voit alors que le problème dual (DSDP) est exactement équivalent au problème suivant :

$$\begin{array}{ll} \max & b^T y \\ \text{t.q.} & A_0 + \sum_{i=1}^m y_i A_i \succeq 0 \end{array} \tag{4.14}$$

qui est la forme sous laquelle étaient originellement présentés les problèmes SDP (voir [110])

Les résultats de dualité faible de programmation linéaire s'étendent aux problèmes SDP. Notons p^* la valeur optimale du problème primal (4.5), et d^* celle de (4.12).

Proposition 4.1.1 *On a :*

$$p^* \geq d^*. \tag{4.15}$$

A priori, on a un saut de dualité non nul $p^* - d^*$ entre les problèmes (4.5) et (4.12), contrairement à la programmation linéaire où il n'y a pratiquement jamais de saut de dualité. De manière analogue à la programmation linéaire, on montre que si les contraintes du problème primal (4.5) et du dual (4.12) sont qualifiées au sens de Slater, c'est à dire que les ensembles réalisables leur correspondant sont d'intérieurs non vides, alors il n'y a pas de saut de dualité et les optima sont atteints pour chaque problème. Plus précisément, on montre

Théorème 4.1.2 *On suppose que les contraintes des problèmes* (4.5) *et* (4.12) *sont qualifiées au sens de Slater.*

Alors, on a $p^* = d^*$ *et les valeurs optimales des problèmes* (4.5) *et* (4.12) *sont atteintes pour les variables primales-duales* X, y, Z *vérifiant :*

$$\begin{array}{rl} \mathcal{A}X - b = 0 & \textit{(réalisabilité primale)} \\ \mathcal{A}^* y + Z - C = 0 & \textit{(réalisabilité duale)} \\ ZX = 0 & \textit{(conditions des écarts complémentaires)} \\ X \succeq 0, Z \succeq 0. & \end{array} \tag{4.16}$$

Les conditions d'optimalité ci-dessus sont d'un grand intérêt, notamment comme nous allons le voir ci-après, pour la conception d'algorithmes de points intérieurs en vue de la résolution des problèmes SDP.

Il est à noter que même lorsque les contraintes ne sont pas qualifiées au sens de Slater, on peut obtenir des résultats similaires d'optimalité et de dualité forte en se ramenant à travailler sur les cônes minimaux de $\mathcal{S}_n^+$ (voir [8]). De même, on pourra se referer aux travaux de Shapiro pour l'obtention des conditions d'optimalités du premier et second ordre, déduit de ceux obtenus pour des problèmes généraux d'optimisation conique.

4.1.4 Quelques remarques

Nous allons à présent évoquer différents points ayant un rapport avec les problèmes SDP.

a) Dégénerescence et Complémentarité

Nous avons jusqu'ìci présenté les problèmes SDP en insistant sur les analogies avec la programmation linéaire. Ces analogies tiennent en grande partie au fait qu'il s'agit dans les deux cas de problèmes d'optimisation conique. Toutefois, comme on peut s'y attendre, toutes les propriétés des programmes linéaires ne s'étendent pas aux problèmes SDP. Ceci s'explique entre autres par le fait que les cônes qui interviennent dans chacun de ces problèmes ne sont pas de même nature. Les cônes considérés en programmation linéaire sont polyédraux, tandis que le cône des matrices semi-définies positives qui intervient en programmation SDP ne l'est pas. En conséquence, les notions de complémentarité stricte et de dégénérescence ne se généralisent pas immédiatement aux problèmes SDP, notammant parce que les conditions sous lesquelles on a ou non dégénerescence nécessitent l'étude de la géométrie de la SDP. On montre que la non dégénérescence implique l'unicité

de solutions pour les problèmes duaux et primaux, mais n'implique pas la complémentarité stricte. La condition de complémentarité stricte de la programmation linéaire se traduit par

$$Z + X \succ 0$$

quand on passe aux problèmes SDP. Elle intervient dans la mise en œuvre pratique des algorithmes de points intérieurs de suivi de trajectoire. Elle n'est pas toujours vérifée en programmation SDP au contraire de la programmation linéaire. Ceci est aussi dû aux propriétés du cône SDP, différentes de celles des cônes polyédraux.

b) Algorithmes et Complexité

Il est prouvé dans KARMAKAR [79] ou NESTEROV et NEMIROVSKI [94] que les problèmes d'optimisation sous contraintes de semi-définie positivité sont des problèmes d'optimisation convexe qui appartiennent à la classe des problèmes pouvant être résolus approximativement en un temps polynomial. Ce résultat de complexité est basé sur l'existence de fonctions barrières **auto-concordantes** pour le cône des matrices semi-définies positives, ainsi que l'on montré NESTEROV et NEMIROVSKI.

Se pose ensuite la question des algorithmes qui peuvent permettre cette résolution en temps polynomial. A l'heure actuelle, les plus populaires parmi ces algorithmes sont ceux dits de points intérieurs. Nous revenons à la fin de ce chapitre sur ces algorithmes. Il existe aussi des algorithmes qui consistent en l'application de méthodes de faisceaux de sous-gradients de l'analyse convexe à la résolution de problèmes SDP. Ces algorithmes tirent avantage du fait que tout problème SDP peut se réexprimer sous la forme d'un problème d'optimisation de valeurs propres. On pourra se referer pour plus de détails aux articles de HELMBERG ET RENDL, OUSTRY dans [115] Bien sûr, il existe d'autres classes d'algorithmes qui sont conçus pour les problèmes SDP. On pourra se referer à [115].

4.2 Quelques rappels d'Analyse numérique

Avant de continuer, nous allons rappeler quelques méthodes ou notions d'Analyse numérique dont nous aurons besoin dans la suite de cette thèse. Nous commencerons par les méthodes de résolution des équations non linéaires dites de Newton et de Gauss-Newton. Ensuite, nous introduirons la méthode de gradient conjugué utilisée pour la résolution d'équations linéaires pour laquelle nous nous attarderons sur la notion de *pré-conditionnement* d'un système linéaire.

4.2.1 Méthodes de types Newton

Dans ce paragraphe, nous cherchons à résoudre l'équation non linéaire (multidimensionnelle) suivante

$$F(x) = 0, \tag{4.17}$$

où $F : \mathbb{R}^m \to \mathbb{R}^p$ est supposée non linéaire (en fait non affine).

a) La méthode de Newton

La méthode de Newton provient de la linéarisation de la fonction F autour du point courant $x_0 \in \mathbb{R}^n$:

$$F(x) = F(x_0) + F'(x_0)\,(x - x_0) + o\,(\|x - x_0\|)\,.$$

Si $F'(x_0)$ est inversible, la solution de l'équation linéaire

$$F(x_0) + F'(x_0)(x - x_0) = 0$$

devient le point courant (en remplaçant x_0) et cela permet d'itérer le procédé en suivant l'algorithme ci-dessous.

Algorithme 4.2.1 (Méthode de Newton) x_0 *point initial*
ε *tolérance*
$i = 0$
tant que $\|F(x_i)\| > \varepsilon$ *faire*
    résoudre le système linéaire : $F'(x_i)\,c = -F(x_i)$
    $x_{i+1} := x_i + c$
    $i \leftarrow i + 1$
fin du tant que

Le principal avantage de la méthode de Newton (cf. [51]) est sa rapidité de convergence à proximité de la solution (la convergence est quadratique si $F'(x^*)$ n'est pas singulière).
Cette méthode a, par ailleurs, deux inconvénients majeurs. D'une part, chaque itération nécessite le calcul de $F'(x)$ et la résolution d'un système linéaire de matrice $F'(x)$, ce qui peut s'avérer très coûteux en temps de calcul (et cela d'autant plus que n est grand). D'autre part, la convergence est seulement locale : le point initial doit être assez proche de la solution pour que l'algorithme atteigne son but.

Entre autres applications, la méthode de Newton a été utlisée pour la résolution de problèmes d'optimisation (convexe) sans contraintes, différentiables. En effet, un problème

$$\min_{x \in \mathbb{R}^m} f(x) \tag{4.18}$$

avec f convexe différentiable, a comme condition nécessaire et suffisante d'optimalité

$$\nabla f(\bar{x}) = 0 \tag{4.19}$$

Pour le calcul de $\bar{x}$, on applique la méthode de Newton présentée plus haut à la résolution de l'équation d'optimalité ci-dessus (4.19). On calcule la direction de recherche en résolvant le système linéaire

$$\nabla^2 f(x_k) d = -\nabla f(x_k). \tag{4.20}$$

Cette idée de résoudre des problèmes d'optimisation en résolvant par la méthode de Newton les systèmes d'optimalité est très répandue. La plupart des algorithmes utilisent cette idée (ou une approximation) pour calculer les directions de recherche. En fait, la proprété de convergence locale de ces algorithmes est souvent un héritage de la méthode de Newton.

b) La méthode de Gauss-Newton

La méthode de Gauss-Newton consiste à résoudre, non pas directement (4.17), mais le problème d'optimisation (quadratique) sans contraintes, différentiable

$$\min_{x\in\mathbb{R}^m} \frac{1}{2}\|F(x)\|^2 = f(x) \tag{4.21}$$

dont une solution optimale est de manière évidente une solution de (4.17). En ce sens, on peut dire que la méthode de Gauss-Newton est une résolution (approximative) au sens des moindres carrés de l'équation (4.17). Elle est en général préférée à la méthode de Newton classique, lorsque la fonction F est définie de $\mathbb{R}^m \to \mathbb{R}^p$ avec $p \neq m$.

En pratique, le problème (4.21) est résolu par une version modifiée de la méthode de Newton à laquelle on rajoute souvent un étape de recherche linéaire. Dans une méthode de Newton classique, on aurait calculé la direction de recherche courante par la linéarisation (4.20). Ici on a :

$$\nabla f(x) = \nabla F(x)^T F(x), \tag{4.22}$$

et

$$\nabla^2 f(x) = \nabla F(x)^T \nabla F(x) + F(x)^T \nabla^2 F(x). \tag{4.23}$$

On peut remarquer que le terme $F(x)^T\nabla^2 F(x)$ de la hessienne devient de plus en plus petit au cours des itérations, puisqu'on cherche un x tel que $F(x) = 0$. On peut donc le négliger. C'est la clé de la méthode de Gauss-Newton.

En d'autres termes, une méthode de Gauss-Newton est un algorithme de Newton avec recherche linéaire appliqué au problème sans contraintes (4.21), où la direction de recherche est obtenue en utilisant l'approximation $\nabla^2 f(x) \approx \nabla F(x)^T \nabla F(x)$ de la Hessienne. On pourra se referer à [96], [101], [51].

4.2.2 Méthode de gradients conjugués

Dans les méthodes que nous avons rappelées précédemment, le calcul des directions de recherche nécessite à chaque fois la résolution d'un système linéaire :

$$Ax = b, \tag{4.24}$$

où la matrice A est rectangulaire dans le cas d'une méthode de Newton, et carrée symétrique dans le cas d'une méthode de Gauss-Newton.

D'une manière générale, les méthodes de résolution utilisées pour ces systèmes linéaires sont des méthodes itératives. La plupart de ces méthodes itératives s'appliquent uniquement pour les cas où la matrice A est carrée (et symétrique souvent). Dans les cas où A est rectangulaire en général, on se ramène à un système équivalent de matrice carrée et symétrique : on parle de *symétrisation* du système.

Nous présentons ci-après une des méthodes itératives les plus utlisées, en grande partie parce qu'elle est simple et peu coûteuse, qu'elle est particulièrement adaptée aux problèmes de grande taille.

a) Présentation de la méthode de gradients conjugués

La méthode de gradient conjué (G-C) est une méthode itérative de résolution de systèmes linéaires pour lesquels la matrice A est carrée, symétrique et définie positive.

Rappelons que le système linéaire (4.24) constitue la condition d'optimalité du problème de minimisation

$$\min_{x\in\mathbb{R}^m} \Phi(x) = \frac{1}{2}x^T Ax - b^T x. \tag{4.25}$$

Par suite, la méthode de G-C peut être présentée aussi comme une méthode de minimisation de fonctions quadratiques convexes. C'est cette présentation que nous adoptons.

Définition 4.2.1 (Vecteurs conjugués) *Soit $\{v_1, \ldots, v_l\}$ un ensemble de vecteurs de $\mathbb{R}^m$. On dit que cet ensemble est conjugué par rapport à la matrice symétrique définie positive A si on a :*

$$v_i^T A v_j = 0, \qquad \forall i \neq j.$$

Cette notion de conjugaison est très importante parce qu'on montre qu'on peut minimiser la fonction quadratique Φ en n itérations en minimisant successivement le long des différentes directions d'un ensemble (d'au moins n vecteurs) conjugué par rapport à A. On en déduit la méthode dite des directions conjuguées qui étant donné un $x_0 \in \mathbb{R}^m$ et un ensemble conjugué $\{v_1, \ldots, v_l\}$, engendre la suite $(x_k)_k$ définie par

$$x_{k+1} = x_k + \alpha_k v_k, \tag{4.26}$$

où α_k est le pas de plus profonde descente de la fonction Φ le long de la direction v_k.

On montre que cette suite $(x_k)_k$ converge vers une solution du système linéaire.

La méthode de *gradients conjugués* est une méthode de directions conjuguée particulière pour laquelle une nouvelle direction conjuguée v_k est calculée **uniquement** à partir de la direction précédente v_{k-1}. Différentes stratégies permettent de faire la mise à jour

$$v_k \leftarrow v_{k-1}.$$

On pourra se referer à [96], [101], [51].

Contrairement aux autres méthodes itératives qui nécessitent des factorisations (Cholesky, LU, etc.), des pivots de Gauss, etc., les calculs principaux nécessaires à une méthode de gradients conjugués consistent en produits scalaires ou produit matrice-vecteur qui interviennent dans la mise à jour $v_k \leftarrow v_{k-1}$. De ce fait, elle est particulièrement adapté aux problèmes de grande taille.

La méthode de G-C converge vers une solution du système linéaire (4.24) en un maximum de m itérations où m est la taille de la matrice A (supposée carrée). En ce qui concerne sa vitesse de convergence, on montre que la méthode de G-C converge très vite vers la solution, pour peu que l'itéré initial en soit suffisamment

prés. Mais, cette vitesse est fortement dépendante de la taille des valeurs propres de la matrice A et surtout de leur distribution spatiale. En effet, la vitesse de convergence peut être controlée par le rapport entre la plus petite et la plus grande des valeurs propres, appelé **conditionnement de** A, noté $\kappa(A)$. On pourra retenir sur ce point que plus les valeurs propres de A sont regroupées (tout en pouvant être facilement distinguées les unes des autres), plus la méthode de gradient conjuguée est efficace.

b) Pré-conditionnement

Nous venons de voir que la vitesse de convergence (et donc l'efficacité) d'une méthode de gradient conjugué dépendait de la distribution des valeur propres de la matrice du système linéaire. Il est donc possible d'accélérer une méthode de G-C en transformant le système linéaire d'origine en un système équivalent ayant une meilleure distribution de valeurs propres. Ce procédé porte le nom de **pré-conditionnement**.

L'ingrédient principal du pré-conditionnement consiste en un changement de variables :

$$\hat{x} = Cx \tag{4.27}$$

où C est une matrice inversible.

La fonction Φ du problème de minimisation (4.25) s'écrit alors :

$$\hat{\Phi}(\hat{x}) = \frac{1}{2}\hat{x}^T(C^{-T}AC^{-1})\hat{x} - (C^{-1}b)^T\hat{x}. \tag{4.28}$$

En appliquant cette fois une méthode de gradient conjugué à la minimisation de la fonction $\hat{\Phi}$, on résout le système linéaire

$$\left(C^{-T}AC^{-1}\right)\hat{x} = C^{-T}b \tag{4.29}$$

et on récupère la solution x de (4.29) par

$$\hat{x} = C^{-1}x. \tag{4.30}$$

La convergence de la méthode de gradients conjugués dépend maintenant de la distribution des valeurs propres de $C^{-T}AC^{-1}$. On peut donc choisir C de manière à avoir une distribution de valeurs propres plus adaptée à une méthode de G-C. On dit qu'on pré-conditionne le sytème linéaire (4.24). Et lorsque qu'on résout (4.29), on dit que le système (4.24) est résolu par gradients conjugués pré-conditionnés. De nombreux travaux existent qui discutent des différents choix de C et des différents critères suivant lesquels $C^{-T}AC^{-1}$ serait plus favorable à une méthode de G-C que A.

En pratique, le changement de variables (4.27) n'est pas effectué explicitement. On modifie l'algorithme de gradients conjugués classique en y introduisant des étapes de pré et post multiplication de la variable x au cours des opérations d'une itération. Nous préciserons cette manière de faire sur un cas pratique au prochain chapitre. Dans certaines présentations du préconditionnement, on n'utilise pas explicitement C, mais la matrice $M = C^TC$ qui a l'avantage d'être symétrique et

définie positive. Dans certains ouvrages ([101] par exemple), c'est cette matrice M qui est appelée pré-conditionneur au lieu de C comme nous l'avons fait ici.

En ce qui concerne le choix de C (ou de M), il n'existe pas de manière optimale de faire, qui s'adapte à tous les cas. Au contraire, un "bon" pré-conditionneur est forcément lié à la structure de A. Toutefois, on peut lister quelques propriétés que doit idéalement avoir un pré-conditionneur. Il doit entre autres être facile à stocker en mémoire, et peu coûteux à inverser (en fait, il suffit que le produit matrice-vecteur par C soit peu coûteux). Le compromis entres ces différents objectifs, souvent antagonistes, est difficile à trouver, et dépend des systèmes linéaires, et surtout de la précision avec laquelle on veut la solution.

Différents pré-conditionneurs généraux ont été proposés (voir [51], [96], [101]). Nous pouvons citer entre autres :

les pré-conditionneurs de type diagonaux qui consistent à prendre M comme étant la matrice diagonale (ou blocs-diagonale, si A est une matrice par blocs) extraite de A,

les pré-conditionneurs de type Cholesky pour lesquels on prend $C = L$ où LL^T représente une factorisation de Cholesky (classique ou incomplète) de A, ou d'une approximation de A (qui peut être la matrice M précédente).

Dans ce dernier cas, si on effectue une factorisation complète de Cholesky, on obtient $C^{-T}AC^{-1} = I$ (ou $C^{-T}AC^{-1} \approx I$), ce qui conduit à un système équivalent dont la matrice est égale au moins approximativement à la matrice identité. Il est donc particulièrement adapté à une méthode de G-C. Malgré quelques inconvénients, notamment le fait qu'il n'est pas toujours facile d'effectuer efficacement (de manière peu coûteuse) la factorisation de Cholesky, le pré-conditionneur de Cholesky (surtout celui utilisant la version incomplète de la factorisation) est un des plus utilisés en Analyse numérique.

4.3 Méthodes de points intérieurs de suivi de trajectoire

Une des méthodes les plus utilisées et les plus efficaces de résolution de problèmes SDP est la méthode de points intérieurs. Le fait qu'on ait justement prouvé que ces méthodes pouvaient permettre notamment une résolution efficace des problèmes SDP a été à la base du regain d'intérêt et de recherche pour ces problèmes. Derrière le terme *points intérieurs* se cachent différents types d'algorithmes : les algorithmes de points intérieurs non réalisables (voir [116]), les algorithmes de réduction de potentiels [115], les algorithmes de suivi de trajectoire. Ces algorithmes ont pour point commun de générer des itérés successifs qui se situent à l'intérieur des ensembles réalisables du problème primal (4.5) et/ou du problème dual (4.12) (voir [116]). L'idée d'adapter ces algorithmes, qui à l'origine servaient à résoudre des programmes linéaires, remonte aux travaux de ALIZADEH [5], NEMIROVSKI et NESTEROV [94]. Le premier a proposé des transpositions quelques fois mécaniques d'algorithmes (primaux-duaux) de points intérieurs de la programmation linéaire aux cas SDP, tandis que les deux autres proposaient une théorie unifiée des méthodes de points intérieurs pour les problèmes d'optimisation conique en

s'appuyant sur la notion fondamentale de **fonction barrière auto-concordante**. Dans la variété des méthodes de points intérieurs, nous allons présenter uniquement les méthodes dites de suivi de trajectoire, et parmi celles-ci, ce sont les versions primales-duales qui nous intéresserons. Ces méthodes constituent déjà une large classe d'algorithmes et sont celles qui sont les plus utilisées en pratique.

4.3.1 Principes généraux

Nous nous proposons de résoudre le problème :

$$\text{(PSDP)} \qquad \begin{array}{ll} \min & \langle C, X\rangle \\ \text{t.q.} & \mathcal{A}X = b, \\ & X \succeq 0. \end{array} \tag{4.31}$$

Nous introduisons la fonction barrière associée à (PSDP) suivante définie uniquement sur le cône des matrices définies positive :

$$f(X) = -\ln \det X. \tag{4.32}$$

On a alors les résultats suivants :

Proposition 4.3.1 *[92, section 10.2, p. 273]*

1. *f est différentiable et*

$$\forall X \in \mathcal{S}_n^+, \qquad \nabla f(X) = -X^{-1}. \tag{4.33}$$

2. *f est strictement convexe.*

Les résultats ci-dessus se montrent assez facilement, le premier en effectuant un développement classique de type Taylor, et le second en calculant explicitement la hessienne de f et en montrant qu'elle est définie positive.

On associe alors au problème (PSDP) le problème barrière :

$$\text{(Pbar)} \qquad \begin{array}{ll} \min & \langle C, X\rangle + \mu f(X) \\ \text{t.q.} & \mathcal{A}X = b, \\ & X \succeq 0 \end{array} \tag{4.34}$$

pour μ positif. Compte tenu de la proposition 4.3.1, (Pbar) est un problème d'optimisation convexe dont les contraintes convexes sont qualifiées au sens de Slater. Puisque ce problème est un problème convexe, les conditions d'optimalité de Karush-Kuhn-Tucker (ou de la Lagrange) sont donc nécessaires et suffisantes. Elles s'écrivent : il existe y tel que

$$\begin{array}{c} C - \mu X^{-1} - \mathcal{A}^* y = 0 \\ \mathcal{A}X = b \\ X \succeq 0, y \in \mathbb{R}^m. \end{array} \tag{4.35}$$

En introduisant comme précédemment la variable duale $Z = C - \mathcal{A}^*y$, il vient que $Z \succeq 0$ compte tenu de l'équation $C - \mathcal{A}^*y = \mu X^{-1}$. On en déduit comme conditions d'optimalité pour le problème barrière

$$\begin{array}{rcrcl} \mathcal{A}X & & & = & b, \\ \mathcal{A}^*y & + & Z & = & 0, \\ -\mu X^{-1} & + & Z & = & 0, \end{array} \tag{4.36}$$

avec $Z \succeq 0$ et $X \succeq 0$. Nous pouvons réécrire ces conditions sous la forme :

$$\begin{array}{rcrcl} \mathcal{A}X & & & = & b, \\ \mathcal{A}^*y & + & Z & = & 0, \\ ZX & & & = & \mu I_n. \end{array} \tag{4.37}$$

Sous cette dernière forme (4.37), les conditions d'optimalité du problème barrière apparaissent comme une perturbation, par l'ajout du terme μI_n à la condition des écarts complémentaires, des conditions d'optimalité des problèmes SDP (4.16). De la vient le nom de *conditions d'optimalité perturbées* que l'on donne à ces équations (4.36) ou (4.37). Cette remarque est d'autant plus importante que cette idée de perturbation de la condition des écarts complémentaires d'équations primales duales d'optimalité est intimement liée aux algorithmes de points intérieurs. On obtient les mêmes résultats si l'on introduit plutôt un problème barrière sur le problème dual (4.12).

L'autre intérêt des conditions d'optimalité perturbées est qu'elles possèdent une unique solution pour tout μ au contraire des problèmes (PSDP). De plus, quand μ tend vers 0, cette solution tend vers une solution optimale de (PSDP) (voir [92],[116].

Théorème 4.3.2 (Existence du Chemin central [115]) *On suppose que les problèmes (PSDP) et (DSDP) ont des solutions strictement réalisables (condition de Slater vérifiée).*

1. *Pour chaque valeur de $\mu > 0$, les équations d'optimalité perturbées* (4.37) *possèdent une unique solution $(X(\mu), y(\mu), Z(\mu))$.*
2. *Pour chaque valeur de μ, $X(\mu)$ est strictement réalisable pour (PSDP), et $y(\mu), Z(\mu)$ le sont pour (DSDP) avec comme saut de dualité*

$$\langle C, X(\mu)\rangle - b^T y(\mu) = \langle X(\mu), Z(\mu)\rangle = n\mu. \tag{4.38}$$

3. *L'ensemble $\{(X(\mu), y(\mu), Z(\mu)) \mid \mu > 0\}$ forme un chemin différentiable dans l'espace primal-dual.*

□

Définition 4.3.1 *L'ensemble $\{(X(\mu), y(\mu), Z(\mu)) \mid \mu > 0\}$ est appelé* **chemin central**.

La preuve des deux premiers résultats du théorème précédent est assez immédiate. La preuve de l'existence et l'unicité de $(X(\mu), y(\mu), Z(\mu))$ peut être donnée en se remémorant qu'il s'agit là de solutions primales duales du problème barrière (4.34) qui est un problème d'optimisation convexe, dont la fonction-objectif est en

plus strictement convexe. Ces variables sont strictement réalisables de manière évidente à cause de la fonction barrière, et de la conditions des écarts complémentaires perturbées.

Le dernier résultat est plus difficile à prouver, en particulier le fait que le chemin central est différentiable. En effet, pour montrer qu'un chemin est différentiable, il suffit de montrer que celui-ci est défini par une fonction (on sous-entend la fonction de plusieurs variables induite par les équations du chemin) différentiable, dont la dérivée est carrée et régulière le long du chemin. Ici, dans notre cas, les équations (4.37) sont définies de manière évidente à partir d'une fonction différentiable. Contrairement à ce qui se passe en programmation linéaire où les matrices sont diagonales, le produit ZX n'est pas symétrique dans le cas général. La fonction induite par les équations (4.37) est donc définie pour $(X, yZ) \in \mathcal{S}_n \times \mathbb{R}^m \times \mathcal{S}_n$ et à valeurs dans l'espace plus grand $\mathcal{S}_n \times \mathbb{R}^m \times \mathcal{M}_n(\mathbb{R})$. Sa différentielle (en fait sa matrice jacobienne) ne peut donc pas être carrée et régulière.

En fait, pour montrer la différentiabilité du chemin central, il faut considérer pour sa définition non pas les équations simples (4.37), mais plutôt la forme (4.36), dans laquelle la troisième équation (c'est elle qui pose problème) est bien à valeurs dans $\mathcal{S}_n$. On montre que sous cette forme, les équations sont définies à partir d'une fonction dont la différentielle est bien carrée régulière.

La forme sous laquelle sont présentées les conditions d'otimalité perturbées, et en particulier la conditions des écarts complémentaires perturbée, est donc importante pour une bonne définition du chemin central. Il en existe plusieurs qui permettent d'obtenir la différentiabilité du chemin central, et à chacune va correspondre des propriétés particulières du chemin central, et comme nous allons le voir plus tard une direction de recherche particulière dans la mise en œuvre d'algorithmes de points intérieurs.

Le chemin central d'un problème SDP est d'une importance capitale dans la mise en œuvre d'une méthode de points intérieurs de type suivi de trajectoire.

Définition 4.3.2 (Points intérieurs par suivi de trajectoire) *Une méthode de points intérieurs par suivi de trajectoire consiste à atteindre (au moins approximativement) l'ensemble des solutions optimales en progressant dans un voisinage autour du chemin central dans le sens des μ décroissant vers 0. Les directions de recherche sont obtenues en résolvant la linérisation des conditions d'optimalité perturbées (éventuellement symétrisées)* (4.37)*, et les matrices X et Z sont maintenues semi-définie positives au cours du déroulement de l'algorithme.*

Elle peut être décrite par :

Algorithme 4.3.1 **Initialisation** *on choisit $L > 1$, $\gamma \in]0, 1[$ et un voisinage associé $\mathcal{N}(\gamma)$.*

on choisit des points initiaux $(X^0, y^0, Z^0) \in \mathcal{N}(\gamma)$

on pose $\mu_0 = \frac{\langle X^0, Z^0 \rangle}{n}$.

Répéter *tant que $\mu_k > 2^{-L}\mu_0$*

1. *Calculer une direction de recherche $(\Delta X, \Delta y, \Delta Z)$.*

2. *Faire la mise à jour*

$$(X^{k+1}, y^{k+1}, Z^{k+1}) = (X^k, y^k, Z^k) + \alpha_k(\Delta X, \Delta y, \Delta Z)$$

pour un réel α_k *tel que* $(X^{k+1}, Z^{k+1}) \in \mathcal{N}(\gamma)$.

3. $\mu_{k+1} \leftarrow \frac{\langle X^{k+1}, Z^{k+1}\rangle}{n}$,

fin

Signalons avant de finir que la mise en œuvre d'un algorithme de points intérieurs nécessite des conditions supplémentaires. Par exemple, il est nécessaire qu'il y ait complémentarité stricte

$$Z + X \succ 0$$

pour le problème. On pourra se referer à [69] et [115] pour de plus amples détails sur ces points.

4.3.2 Directions de recherche de Newton

Nous nous intéressons plus précisément à présent au calcul des directions de recherche. Celles-ci sont obtenues par résolution de la linéarisation de (formes symétrisées) des équations d'optimalité (4.37). Dans la plupart des cas, celles-ci sont résolues en utilisant la méthode de Newton, de là vient le nom de *direction de recherche de Newton* que l'on donne aux différentes directions de recherche ainsi calculées.

Nous avons vu précédemment que les conditions d'optimalité d'un problème d'optimisation sous contraintes de semi-définie positivité, obtenue après introduction d'une barrière logarithmique (4.37) étaient :

$$\Phi_\mu(X, y, Z) = \begin{pmatrix} \mathcal{A}X \\ \mathcal{A}^* y + Z \\ ZX \end{pmatrix} = \begin{pmatrix} b \\ 0 \\ \mu I_n. \end{pmatrix} \tag{4.39}$$

Puisque le produit ZX n'est pas symétrique ici, la fonction Φ_μ ci-dessus est définie sur $\mathcal{S}_n \times \mathbb{R}^m \times \mathcal{S}_n$ à valeurs dans $\mathcal{S}_n \times \mathbb{R}^m \times \mathcal{M}_n(\mathbb{R})$. Nous avons également vu que pour assurer que le chemin central est différentiable, il fallait que Φ_μ soit tel que sa différentielle (sa matrice jacobienne) soit carrée et régulière. Cela nécessite entre autres que les ensembles de départ et d'arrivée de Φ_μ soient les mêmes (à un isomorphisme près). En fait, cette condition sur la matrice jacobienne de Φ_μ est aussi nécessaire pour assurer l'existence des directions de recherche puisque cette jacobienne est aussi la matrice du système linéaire dont la solution donne ces directions de recherche. Pour avoir des conditions d'optimalité pour lesquelles la fonction Φ_μ vérifie cette condition sur la jacobienne, puisque les deux premières équations sont affines, il suffit en pratique de remplacer la dernière équation

$$ZX - \mu I_n = 0 \tag{4.40}$$

par des équations équivalentes qui sont, elle, définies dans $\mathcal{S}_n$.

Ainsi par exemple, on peut remplacer (4.40) par

$$XZ + ZX = 2\mu I_n. \tag{4.41}$$

Cette équation est obtenue par symétrisation de l'équation (4.40). En résolvant les équations d'optimalité (4.37) ou (4.39) avec comme troisième équation (4.41), les directions de recherche de Newton ainsi générées portent le nom de **direction AHO**, pour ALIZADEH, HAEBERLY, OVERTON [6] qui ont été les instigateurs de cette symétrisation.

La symétrisation (4.41) apparaît comme une manière naturelle de rendre l'équation (4.40) symétrique. La direction AHO bénéficie de cet état de fait, et en pratique, elle est très efficace. Elle permet d'obtenir des solutions très précises. Mais, elle présente beaucoup d'inconvénients. D'un point de vue théorique, cette direction n'a pas la propriété intéressante d'invariance aux ajustements affines, et de nombreux résultats tels que la convergence en temps polynomial sont difficiles à obtenir. D'un point de vue pratique, la linéarisation de l'équation (4.41) donne :

$$\frac{1}{2}(\Delta XZ + Z\Delta X + \Delta ZX + X\Delta Z) = \mu I_n - \frac{1}{2}(XZ + ZX)$$

dont la résolution nécessite celle d'équations de Lyapounov comportant des matrices non symétriques et, par conséquent, l'usage des compléments de Schur. Ceci s'avère très coûteux, et limite grandement la taille des problèmes qui peuvent être traités.

Il existe de nombreuses autres directions de recherche de Newton qui sont obtenues à partir d'autres symétrisations et/ou transformations de l'équation (4.40). Elles diffèrent les unes des autres par les différentes formes de conditions d'optimalité perturbées ou de linéarisations de celle-ci, qui sont adoptées. Toutefois, elles présentent un point commun pittoresque : les acronymes variés qui les identifient et qui sont encore plus folkloriques que ceux des méthodes de quasi-Newton qui sont leur plus illustres devancières. Nous pouvons citer parmi les plus utilisées ou les plus représentatives :

la direction HRVW/KSH/M : les directions de ce type proviennent de la réécriture de (4.40) sous la forme

$$X - \mu Z^{-1} = 0 \quad \text{ou sa forme duale } Z - \mu X^{-1} = 0. \tag{4.42}$$

Elles sont dues à HELMBERG-RENDL-VANDERBEI-WOLKOWICZ [71], KOJIMA-SHINDOH-HARA [83] et MONTEIRO [93]. De nombreuses autres directions, comme celle de MONTEIRO-ZHANG (voir [115]), sont des extensions ou des généralisations de cette direction.

la direction Nesterov-Todd [95] : cette direction est obtenue à partir de la même troisième équation (4.42), mais, l'équation linéarisée est modifiée par l'introduction d'une matrice dite d'ajustement. L'équation linéarisée résolue est :

$$\Delta X + W\Delta SW = \mu Z^{-1} - X \text{ avec } W = W_{nt} = Z^{-\frac{1}{2}}(Z^{-\frac{1}{2}}XZ^{-\frac{1}{2}})^{-\frac{1}{2}}Z^{-\frac{1}{2}} \tag{4.43}$$

Il existe bien sûr de nombreuses autres directions de recherches de Newton, voir [109].

4.3.3 Exemples d'algorithmes

De nombreux algorithmes de points intérieurs primaux-duaux de suivi de trajectoire existent. La plupart utilisent les directions de recherche de type AHO, HRVW/KSH/M, NT que nous avons présentées précédemment. On peut décrire ces algorithmes sous la forme suivante :

Algorithme 4.3.2 Initialisation – *Données :* C, b.
- *Points initiaux réalisables :* X^0, Z^0, y^0.
- *Tolérance :* ε *(pour la convergence des points intérieurs).*
- $\mu_0 = \frac{\langle X^0, Z^0 \rangle}{n}$, $\sigma_0 \in]0, 1[$.

Itération *Tant que critère d'arrêt* $> \varepsilon$,
- *Calculer la direction de recherche (de Newton)* $(\Delta X, \Delta y, \Delta Z)$ *en résolvant*

$$\Phi'_{\sigma_k \mu_k}(X^k, y^k, Z^k) \begin{pmatrix} \Delta X \\ \Delta y \\ \Delta Z \end{pmatrix} = -\Phi_{\sigma_k \mu_k}(X^k, y^k, Z^k)$$

- *Faire la mise à jour*

$$(X^{k+1}, y^{k+1}, Z^{k+1}) = (X^k, y^k, Z^k) + \alpha^k (\Delta X, \Delta y, \Delta Z)$$

  pour un réel α_k *tel que* $(X^{k+1}, Z^{k+1}) \in \mathcal{N}(\gamma)$.
- *faire la mise à jour :*$\mu_{k+1} \leftarrow \frac{\langle X^{k+1}, Z^{k+1} \rangle}{n}$ *et* $\sigma_{k+1} \leftarrow \sigma_k$ *de façon à se recentrer.*

Par rapport à la précédente description des algorithmes de points intérieurs, il est apparu une différence : la présence d'un paramètre supplémentaire σ, appelé *paramètre de recentrage*. C'est un nombre réel compris entre 0 et 1. Il paramétrise en pratique le voisinage $\mathcal{N}(\gamma)$ de l'algorithme 4.3.1 : il permet de se maintenir raisonnablement près (dans un voisinage) du chemin central, tout en évitant de trop se rapprocher de la frontière du domaine réalisable. En effet, on peut noter que :

si $\sigma = 0$, on obtient une direction de recherche qui est en fait une direction de Newton sur les conditions d'optimalité (4.16) du problème SDP de départ, et non plus sur les conditions perturbées. On dit souvent qu'il s'agit de direction d'ajustement affine. Elle permet de réduire fortement le paramètre μ. Cette direction a tendance à ramener les itérés près de la frontière du domaine réalisable. On peut aussi voir qu'elle permet de prédire la région dans laquelle se trouve la solution optimale. Ceci fait qu'on l'appelle aussi *direction prédictrice*.

si $\sigma = 1$, on obtient une direction de recherche qui indique un point qui se trouve exactement sur le chemin central, puisque les équations linéarisées sont exactement les équations d'optimalité perturbées. On dit qu'il s'agit de direction de recentrage. Elle ne permet pas souvent de réduction substancielle de μ. Par contre, si l'itéré courant n'est pas au voisinage du chemin central, elle permet de se ramener dans le voisinage du chemin, donc de faire une correction de trajectoire. C'est pourquoi elle est aussi appelée *direction correctrice*.

Bien sûr, lorsque $\sigma \in]0,1[$, on a une direction qui amène dans un voisinage du chemin central plus ou moins près du bord selon que σ est plus ou moins près de 0.

La mise à jour du paramètre σ dans un algorithme de suivi de trajectoire est un compromis entre les deux objectifs contradictoires que sont : faire décroître μ vers 0, et donc prendre σ proche de 0, et rester dans un voisinage du chemin central, et prendre σ proche de 1. De plus, ce choix du paramètre de recentrage influence énormément le choix du pas α : plus on est proche du chemin cenral, moins on a la latitude de se déplacer et on ne peut faire que des petits pas. A chaque stratégie de mise à jour du paramètre σ et du pas α correspond un algorithme primal dual de points intérieurs par suivi de trajectoire. On peut noter parmi les plus connus :

l'algorithme préditeur - correcteur pur. C'est un algorithme qui consiste à faire alterner deux types différents d'étapes : des étapes prédictrices ($\sigma = 0$) qui permettent de réduire μ, et des étapes centralisatrices ($\sigma = 1$) qui consistent à se rapprocher le plus possible du chemin central. La terminologie *predicteur - correcteur* provient d'une analogie avec la théorie des équations différentielles ordinaires. Se reporter à [115], [116].

l'algorithme prédicteur-correcteur de Mehrotra [116]. L'idée est la même que ci-dessus : alterner des pas correcteurs et des pas (plus ou moins) centralisateurs. La différence ici est qu' on ne fait pas des pas de centralisation purs, mais σ est plutôt choisi dans $]0,1]$ de manière adaptative. Beaucoup d'algorithmes pratiques ou de codes de points intérieurs sont de ce type.

les algorithmes à grands et petits pas. Ce sont des algorithmes un peu plus généraux que ceux présentés ci-dessus. Au contraire de ce que pouvait laisser penser leurs noms, la différence entre ces algorithmes ne se fait pas directement sur la valeur du pas α, mais sur le type de voisinage du chemin central $\mathcal{N}(\gamma)$ dans lequel on veut que les itérés de l'algorithme se situent. Ces voisinages sont en général définis à partir de normes ou semi-normes dans l'espace primal dual (voir [115]). Sans entrer dans les détails, nous pouvons dire que pour les algorithmes à petits pas on choisit des voisinages définis à partir de la norme euclidienne, tandis que pour ceux à grands pas, celle utilisée est du type de la norme infinie. On trouvera dans à [115] des précisions sur ce point. Cette différence se traduit en pratique par différents choix des paramètres σ et α. Pour un algorithme à petits pas, on prend en général des valeurs constantes $\alpha = 1$ et $\sigma_k = \overline{\sigma} \in [0,1]$ au cours des itérations. Par exemple, l'algorithme prédicteur-correcteur pur précédent est du type petits pas. L'algorithme à grands pas au contraire est caractérisé par des stratégies adaptatives (dépendantes de l'itération courante) de mise à jour de ces paramètres σ et α.

Les algorithmes tels que présentés jusqu'à présent sont ceux qui sont les plus utilisés en pratique. Ce sont les méthodes qui marchent le mieux pour résoudre des problèmes SDP. Toutefois, ils ont en commun le fait d'être des transpositions directes d'algorithmes qui étaient appliqués en programmation linéaire. Même si cette idée est naturelle puisque les problèmes linéaires sont des problèmes SDP, et qu'elle s'avère judicieuse puisqu'elles marchent, le fait que les problèmes linéaires

soient des problèmes SDP très particuliers induit des mauvais comportements en pratique de ces algorithmes sur les problèmes SDP un peu ardus. Par exemple, les systèmes linéaires desquels proviennent les directions de recherche sont vectorisés avant dêtre résolus. Il faut donc construire la matrice du système à chaque fois. Ceci est très limitatif dès qu'on ambitionne de résoudre des problèmes de grande taille. D'autre part, les systèmes linéaires obtenus de la linéarisation des équations (4.37) sont souvent creux. Mais il est en général très difficile d'exploiter cet avantage. Il est donc nécessaire d'envisager des algorithmes de points intérieurs qui soient adaptés aux problèmes SDP, et qui tirent avantage des données et variables matricielles que nous avons.

4.4 Points intérieurs par Gauss-Newton

Nous proposons dans cette dernière partie une des premières tentatives d'adaptation des algorithmes de points intérieurs aux problèmes SDP. Il s'agit d'algorithmes pour lesquels :

- les directions de recherches sont de celles de type Gauss-Newton proposées et étudiées par KRUK et al. (voir [84]) comme alternative à celle de Newton ;
- les systèmes linéaires dont la résolution donne les directions de recherche sont résolus par gradients conjugués plutôt qu'après symétrisation par complément de Schur et autres équations de Lyapounov comme c'était le cas précédemment ;
- une étape de "crossover" est introduite en fin d'algorithme, ce qui permet de récupérer de la convergence q-quadratique asymptotique.

4.4.1 Direction de recherche de Gauss-Newton

a) Motivations

Les directions de recherche de Gauss-Newton ont été proposées comme alternatives aux directions de Newton. Le but était d'obtenir des directions de recherche qui soient aussi efficaces que celles de Newton, notamment la direction AHO et la direction HRVW/KSH/M, tout en évitant du mieux possible leurs inconvénients.

En effet, d'un point de vue pratique, nous avons vu que le calcul de directions de recherche AHO, par exemple, nécessitait en général la résolution d'équations de Liapounov, l'utlisation des compléments de Schur, etc. De plus, dans certains cas, comme la direction RVW/KSH/M, du fait de la présence de l'inverse d'une matrice dans la forme (4.42) de l'équation d'optimalité perturbées utilisée, plus on se rapproche de l'optimum, plus on se rapproche du bord du domaine réalisable, et plus la matrice du système linéaire devient près d'être singulière rendant difficile, voire parfois impossible, le calcul des directions de recherche de Newton.

Au delà de ces inconvénients qui apparaissent lors des calculs, il existe d'autres inconvénients dus à la forme des équations d'optimalité perturbées utilisées. En ef-

fet, la forme de cette équation qui est la plus simple

$$ZX - \mu I_n = 0 \tag{4.44}$$

ne peut pas être linéarisée pour obtenir des directions de recherche (jacobienne obtenue par linéarisation pas carrée). On est obligé de la symétriser, c'est-à-dire, lui trouver des formes équivalentes dont la linéarisation conduit à des jacobiennes carrées et régulières. Ce faisant, on effectue, d'un certain point de vue, un préconditionnement de l'équation (4.44). Mais, ce préconditionnement est contre - nature : on remplace une équation simple (4.44) par des équations qui sont de nature plus compliquée (4.41), (4.42), (4.43) puisqu'elle sont plus **non linéaires** que (4.44) qui est juste bilinéaire. Certains des inconvénients que l'on rencontre lors du calcul des directions de recherche de Newton proviennent d'ailleurs de ces fortes non-linéarités.

Si l'on veut éviter ces inconvénients, il apparaît naturel de travailler plutôt avec l'équation bilinéaire d'optimalité perturbée (4.44). Mais alors, la linéarisation obtenue ne peut plus être résolue par une méthode de Newton classique : c'est une équation surdéterminée puisque définie sur $\mathcal{S}_n \times \mathbb{R}^m \times \mathcal{S}_n$ à valeurs dans $\mathcal{S}_n \times \mathbb{R}^m \times \mathcal{M}_n(\mathbb{R})$. En général, en Analyse numérique, lorsqu'on est face à une telle équation non linéaire surdéterminée, la démarche classique est de la résoudre au sens des moindres carrés. A la place de la méthode de Newton, on utilise donc plutôt une méthode de Gauss-Newton, ce qui donne naissance à une nouvelle classe de direction de recherche : *les directions de Gauss-Newton (G-N).*

b) Conditions bilinéaires d'optimalité

Nous présentons dans ce qui suit une démarche pratique de calcul de la direction de Gauss-Newton. L'idée principale, qui est celle qui sous-tend ce nouveau cadre des méthodes de points intérieurs, est que pour trouver les directions de Gauss-Newton on peut se ramener à utiliser des outils classiques d'Analyse numérique plutôt que des outils tels que les compléments de Schur ou les équations de Lyapounov qui sont très particuliers. On pourra ainsi profiter de toute l'expertise qui a été développée depuis des années en Analyse numérique.

Nous choisissons donc les conditions d'optimalités perturbées sous une forme dans laquelle la troisième équation est (4.44) :

$$F_\mu = \begin{pmatrix} \mathcal{A}X - b \\ \mathcal{A}^* y + Z - C \\ ZX - \mu I_n \end{pmatrix} = 0. \tag{4.45}$$

La linéarisation de l'équation précédente nous donne ceci :

$$\begin{pmatrix} 0 & \mathcal{A}^* & -I \\ \mathcal{A} & 0 & 0 \\ X & 0 & Z \end{pmatrix} \begin{pmatrix} \Delta Z \\ \Delta y \\ \Delta X \end{pmatrix} = -F_\mu(X, y, Z). \tag{4.46}$$

Le système linéaire ci-dessus est de grande taille : la matrice est à $\begin{pmatrix} n+1 \\ 2 \end{pmatrix} \times m \times n^2$ lignes et $\begin{pmatrix} n+1 \\ 2 \end{pmatrix} \times m \times \begin{pmatrix} n+1 \\ 2 \end{pmatrix}$ colonnes. On pourrait tenter de le

résoudre directement, mais cela pourrait devenir rapidement prohibitif. Les techniques de résolution utilisées dans un algorithme de points intérieurs classique (avec direction de Newton) procèdent souvent par une étape de pré-traitement des équations linéaires (4.46). Celle-ci, héritée de la pratique en programmation linéaire, consiste en une étape d'élimination de variables dans (4.46). Par exemple, comme en programmation linéaire, on peut déduire ΔX de la dernière équation, et la réinjecter dans les deux autres. Mais, ceci a le défaut de nécessiter l'inversion de Z, conduisant à des problèmes mal posés quand on s'approche du bord. KRUK et al. [84] ont proposé un autre schéma qui consiste à éliminer d'abord ΔZ de l'équation de réalisabilté duale (la deuxième). En l'injectant dans les équations restantes, on obtient un système de taille plus réduite. Cette procédure diffère fondamentalement de la première par le fait que l'élimination ne nécessite qu'une addition de matrices au lieu d'inversions et de produits de matrices.

L'intérêt des éliminations de variables est qu'elles conduisent à des systèmes de plus petite taille, qui sont de toute façon plus rapides à résoudre. Sur un problème pratique, l'idée est d'effectuer autant d'éliminations de variables que possible. Seulement, ce faisant, on détruit une propriété très importante du système (4.46) : le caractère creux. Cette perte peut être un inconvénient à cette étape d'élimination, surtout lorsque les équations sont destinées à être résolues au sens des moindres carrés, par gradient conjugué. Dans le but de faire cette élimination de variables tout en conservant le caractère creux du système linéaire (4.46) et en évitant les autres inconvénients évoqués au paragraphe précédent, la stratégie suivante est proposée par WOLKOWICZ [114] : éliminer ΔX (respectivement ΔZ) de l'équation de réalisabilité primale (respectivement duale), et les injecter dans l'équation de complémentarité perturbée (4.44) conduisant ainsi à des conditions d'optimalité **bilinéaires**.

On rappelle que les matrices A_i définissant l'opérateur $\mathcal{A}$ sont supposées linéairement indépendantes. Il en résulte que l'opérateur $\mathcal{A}$ est de rang maximal m.

Nous noterons $\mathcal{A}^\dagger$ le pseudo-inverse de Moore-Penrose de $\mathcal{A}$. Introduisons l'opérateur suivant

$$\mathcal{N} : \mathbb{R}^{\binom{n+1}{2}-m} \to \mathcal{S}_n$$

dont l'image est le noyau de $\mathcal{A}$. Nous l'appelons "noyau" de $\mathcal{A}$. On peut montrer :

Proposition 4.4.1

$$\mathcal{A}X = b \quad \Leftrightarrow \quad X = \mathcal{A}^\dagger b + (I_n - \mathcal{A}^\dagger \mathcal{A})W \quad \textit{pour } W \in \mathcal{S}_n, \tag{4.47}$$

$$\Leftrightarrow \quad X = \mathcal{A}^\dagger b + \mathcal{N} w \quad \textit{pour } w \in \mathbb{R}^{\binom{n+1}{2}-m}. \tag{4.48}$$

Ce résultat est une conséquence des propriétés des pseudo-inverses d'opérateurs linéaires. En utilisant ce résultat, on peut procéder à une étape d'élimination de variables directement sur les équations (4.45), plutôt que sur leur linéarisation (4.46). En remplaçant Z et X par leurs valeurs dans l'équation de complémentarité perturbée (4.44), on obtient une équation bilinéaire d'optimalité de taille plus petite que (4.45).

Proposition 4.4.2 *[114] On suppose que les problèmes SDP primaux et duaux* (4.5) *et* (4.12) *ont leurs contraintes qualifiées au sens de Slater. On suppose aussi $\mathcal{A}$ de rang maximal et $\mathcal{N}$ défini comme précédemment.*

Alors, les variables primales duales (x, y) sont optimales pour les problèmes (4.5) *et* (4.12) *si et seulement si*

$$F(x, y) = (\mathcal{A}^*(y) - C)(\mathcal{A}^\dagger + \mathcal{N}x) = 0, \tag{4.49}$$

avec

$$\mathcal{A}^*(y) - C \succeq 0 \text{ et } \mathcal{A}^\dagger + \mathcal{N}x \succeq 0.$$

La proposition ci-dessus provient directement de la réexpression des résultats primaux duaux de la section 4.3, en tenant compte de l'introduction des opérateurs $\mathcal{A}^\dagger$ et $\mathcal{N}$ suivant la relation (4.48). Les équations d'optimalité perturbées (4.45) obtenues après pénalisation logarithmique, et éventuellement prétraitement, deviennent alors :

$$F_\mu(x, y) = (\mathcal{A}^*(y) - C)(\mathcal{A}^\dagger + \mathcal{N}x) - \mu I_n = 0. \tag{4.50}$$

Le théorème suivant donne une des conséquences intéressantes de la réécriture que nous venons de proposer.

Théorème 4.4.3 *[114]*

Considérons les problèmes SDP primal (4.5) *et dual* (4.12)*. On suppose que $\mathcal{A}$ est de rang maximal, $\mathcal{N}$ définit le noyau de $\mathcal{A}$ suivant* (4.48)*.*

On suppose que les solutions optimales primales duales X, y, Z des problèmes (4.5) *et* (4.12) *satisfont* **strictement** *la condition de complémentarité, c'est-à-dire $Z + X \succ 0$. Alors, la matrice du système linéaire*

$$-F_\mu(x, y) = F'(x, y) \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}, \tag{4.51}$$

$$= (\mathcal{A}^*(y) - C)\mathcal{N}(\Delta x) + \mathcal{A}^*(\Delta y)(\mathcal{A}^\dagger + \mathcal{N}x), \tag{4.52}$$

c'est-à-dire, ($F'(x, y)$, jacobienne de F en (x, y)) est de rang maximal (régulière).

Voir [84] pour une preuve de ce résultat. En tout état de cause, c'est un résultat très important puisqu'il montre qu'en procédant comme ci-dessus, on évite les problèmes mal posés et les matrices de systèmes linéaires non (ou pas assez) régulières que l'on observe dans le cas de directions de Newton. Ceci, outre le fait déjà évoqué que l'on se ramène à des problèmes de plus petite taille, plaide en faveur de l'adoption de la démarche que nous venons de présenter. A cela s'ajoute le fait que, puisque les systèmes linéaires sont résolus par gradients conjugués, l'équation d'optimalité sous une forme bilinéaire, avec une jacobienne toujours de rang maximal, est particulièrement adaptée. Toutefois, il nous faut modérer ce qui a été dit : la démarche n'est intéressante et efficace que si l'on réussit à exprimer les contraintes affines du problème SDP (4.5) au moyen d'un opérateur (linéaire) $\mathcal{A}$, dont l'adjoint et le pseudo inverse sont aisément calculables (au moins numériquement), et pour lequel, on peut facilement choisir un "bon" opérateur noyau $\mathcal{N}$. Par exemple, on montre si $\mathcal{N}$ est une isométrie, le conditionnement de la jacobienne F' obtenue à partir de l'équation bilinéaire (4.49) ou (4.50) est au moins aussi bon, sinon meilleur, que celui de la jacobienne obtenue à partir de (4.45).

4.4.2 Algorithmes de points "intérieurs-extérieurs"

Nous présentons ici le nouvel algorithme de points intérieurs proposé comme alternative à ceux que nous avions présenté à la section précédente qui utilisent des directions de Newton. Le principe est toujours celui d'un algorithme de suivi de trajectoire. Mais, contrairement aux algorithmes qui s'imposaient à la fois d'être dans un voisinage du chemin central et de se maintenir réalisables (en imposant à X et Z de demeurer définis positifs au cours de l'algorithme), nous considérons ici que seul le fait d'être dans un voisinage du chemin central est primordial. On ne maintiendra pas nécessairement la réalisabilité de X et Z.

a) Notion de "crossover"

La technique de "crossover", pour laquelle nous conservons la terminologie anglaise faute d'une traduction satisfaisante en français, est directement inspirée de l'intention de ne pas forcément privilégier la réalisabilité de X et Z au cours du déroulement de l'algorithme.

On peut remarquer que la linéarisation de l'équation d'optimalité bilinéaire (4.49) conduit à un système linéaire dont la matrice est non dégénérée (de rang maximal) tout au long de l'algorithme. Il existe donc en chaque point du chemin central et surtout *de l'optimum*, une région de convergence quadratique (cela veut dire qu'une méthode de Newton pure convergerait quadratiquement si elle était initialisée dans cette région). Ces régions contiennent également des matrices Z et X qui **ne sont pas** définie positives. Si on ne force pas Z et X à être réalisables, il est donc possible de faire des grands pas. Et il n'est pas nécessaire de forcer les matrices Z et X à rester définies positives (réalisables) au cours des itérations, comme cela se fait dans la plupart des algorithmes de points intérieurs, puisqu'on peut montrer (voir [114]) que de toute façon, on revient toujours dans le domaine réalisable.

L'idée du "crossover" est une conséquence de ce constat : dans le déroulement de l'algorithme de points intérieurs, on aboutit forcément à un moment à un itéré courant qui appartient aussi à la région de convergence quadratique de la solution optimale du problème. A partir de ce point-là, il n'est plus nécessaire de se forcer à rester réalisable ou dans un voisinage du chemin central. On fixe le paramètre de centralisation à $\sigma = 0$ et les pas à $\alpha = 1$. Cela revient en fait à appliquer directement la méthode de Newton pure à l'équation d'optimalité (non perturbée) (4.49). Cela permet de converger plus rapidement (puisque la convergence est alors superlinéaire (quadratique)), donc de récupérer asymptotiquement de la convergence quadratique pour l'algorithme de points intérieurs.

La question qui se pose alors est comment calculer exactement le voisinage de convergence quadratique d'un point donné pour une équation donnée. Cette question a donné lieu à de très nombreux travaux, et en fait, la question n'a jamais pu être tranchée de manière définitive. Il existe différents types de majorations qui permettent d'estimer cette région de convergence quadratique. Dans nos travaux, nous avons choisi ici d'utiliser les résultats de [51] pour développer une heuristique pour mettre en œuvre la technique de "crossover".

On suppose que l'on applique une méthode de Gauss-Newton à la résolution

de l'équation

$$F(s) = 0 \qquad \text{avec } s = \begin{pmatrix} x \\ y \end{pmatrix}. \tag{4.53}$$

On a le théorème classique suivant :

Théorème 4.4.4 *([51, Théorème 10.2.1]) Soit $F : \mathbb{R}^n \to \mathbb{R}^m$, et soit $f(v) = \frac{1}{2}F(v)^T F(v)$ supposée de classe $\mathcal{C}^2$ dans un ouvert D de $\mathbb{R}^n$.*

On suppose que

- *la matrice jacobienne $J(v) = F'(v)$ est lipschitzienne sur D de constante γ, avec $\|J(v)\|_2 \leq \alpha$ pour tout $v \in D$,*
- *il existe $v_* \in D$ et des réels $\lambda, \bar{\sigma} \geq 0$, tel que*
  - *$J(v_*)^T F(v_*) = 0$,*
  - *λ est la plus petite valeur propre de $J(v_*)^T J(v_*)$,*
  - *et*

$$\|(J(v) - J(v_*))^T F(v_*)\|_2 \leq \bar{\sigma}\|v - v_*\|_2, \quad \forall v \in D. \tag{4.54}$$

Si $\bar{\sigma} < \lambda$, alors pour tout $c \in]1, \frac{\lambda}{\bar{\sigma}}[$, il exsite $\varepsilon > 0$ tel que pour tout v_0 tel que $\|v - v_*\| < \varepsilon$, la suite générée par une méthode de Gauss-Newton

$$v_{k+1} = v_k - (J(v_k)^T J(v_k))^{-1} J(v_k)^T F(v_k) \tag{4.55}$$

est bien définie, converge vers v_*, et vérifie

$$\|v_{k+1} - v_*\| \leq \frac{c\bar{\sigma}}{\lambda}\|v_k - v_*\| + \frac{c\alpha\gamma}{2\lambda}\|v_k - v_*\|^2 \tag{4.56}$$

et

$$\|v_{k+1} - v_*\| \leq \frac{c\bar{\sigma} + \lambda}{2\lambda}\|v_k - v_*\| \leq \|v_k - v_*\|^2. \tag{4.57}$$

□

Ce théorème, et surtout les inégalités (4.56) et (4.57), tout en montrant la convergence quadratique lorsque la jacobienne est de rang maximal, nous permettra de déterminer la région de convergence quadratique autour d'un point. On peut déjà remarquer que, puisque nous résolvons une équation dont une solution exacte existe, la solution au sens des moindres carrés est atteinte, et par la suite, on a $\bar{\sigma} = 0$. L'inégalité (4.56) devient alors :

$$\|v_{k+1} - v_*\| \leq \frac{c\alpha\gamma}{2\lambda}\|v_k - v_*\|^2. \tag{4.58}$$

b) Exemples d'algorithmes

Un algorithme de points "intérieurs-extérieurs" est un algorithme qui suit la démarche que nous avons présentée précédemment pour un algorithme primal dual de points intérieurs de suivi de trajectoire 4.3.2 avec les modifications suivantes :

1. les directions de recherche sont des directions de Gauss-Newton obtenues à partir des conditions d'optimalité bilinéaires (4.49) ;
2. la linéarisation (4.51) de (4.49) est résolue au sens des moindres carrés par une méthode de gradients conjugués pré-conditionnés ;

3. une étape de "crossover" est introduite à la fin de l'algorithme une fois que l'on est arrivé dans un voisinage de l'optimum. Cela permet de récupérer de la convergence q-quadratique asymptotique.

L'algorithme de points intérieurs-extérieurs tel que présenté ci-dessus est adapté aux problèmes pour lesquels on peut calculer facilement l'opérateur linéaire $\mathcal{A}$ définissant les contraintes affines, son adjoint, son pseudo-inverse, ainsi que l'opérateur $\mathcal{N}$ définissant le noyau de l'opérateur $\mathcal{A}$. La démarche que nous venons de proposer a jusqu'à présent été appliquée à la résolution de problèmes SDP qui sont des relaxations SDP de problèmes d'optimisation combinatoire : [114] par exemple. Nous en proposons une application au problème d'approximation par matrices de corrélation au prochain chapitre.

Chapitre 5

Approximation par matrices de corrélation

Nous abordons dans ce chapitre notre second problème d'approximation matricielle : l'approximation par matrices de corrélation. Ce problème provient d'applications en Statistiques et en Finances. Nous avons mis en œuvre pour ce problème un algorithme de type points intérieurs avec directions de recherche de Gauss-Newton suivant le modèle que nous avons décrit en fin de chapitre précédent. **Ce travail a été fait en collaboration avec M.F. ANJOS, N.J. HIGHAM et H. WOLKOWICZ [9]**. Nous comparons cette approche avec celles que nous avons décrites précédemment qui ont été mises en œuvre par J. MALICK [88] en ce qui concerne l'algorithme conique dual, par N.J. HIGHAM [75] et nous-même parallèlement.

5.1 Approximation par matrices de corrélation

Nous sommes toujours placé dans l'espace de Hilbert $\mathcal{S}_n$ des matrices carrées symétriques, muni du produit scalaire associé à la norme de Fröbenius. Nous rappelons aussi qu'une matrice symétrique est dite semi-définie positive lorsque toutes ses valeurs propres sont positives.

5.1.1 Notions de matrice de corrélation

Définition 5.1.1 *On appelle* **matrice de corrélation** *toute matrice carrée symétrique semi-définie positive, dont tous les termes diagonaux sont égaux à* 1.

Proposition 5.1.1 *Les matrices de corrélation forment un ensemble convexe compact dans l'espace de Hilbert* $\mathcal{S}_n$.

Introduisons l'opérateur $\text{diag} : \mathcal{M}_n(\mathbb{R}) \to \mathbb{R}^n$ qui à une matrice carrée M associe le vecteur de $\mathbb{R}^n$ formé des termes diagonaux de M. En utilisant cet opérateur, on peut voir que les matrices de corrélation vérifient $X \succeq 0$ et $v(M) := diag(M) - e = 0$. La fonction v étant affine, il est facile de voir que l'ensemble des matrices de corrélation est convexe et fermé. De plus, cet ensemble est borné puisque ses valeurs propres le sont : elles sont positives et de somme égale à la trace de M qui vaut n puisque tous les termes diagonaux valent 1.

Définition 5.1.2 *L'ensemble des matrices de corrélation que nous notons $\mathcal{E}$ est appelé elliptope.*

Les matrices de corrélations apparaissent naturellement dans différents domaines :

- en théorie des graphes : certains problèmes de complétion matricielle sont modélisés en utilisant des graphes. Dans cette modélisation, les matrices de corrélation jouent souvent un rôle important. On pourra se référer à [1], [2], [85].
- en Statistiques et Finances : ce sont des matrices qui collectent les différents coefficients de corrélation qui existent pour un nombre fini de variables aléatoires. Dans le cas de la Finance, ces variables aléatoires sont par exemple les cours de différentes actions cotées en Bourse.

On retrouve également les matrices de corrélation en contrôle optimal, lorsque l'on applique une méthode de "décomposition orthogonale propre" où elle collecte les différents produits scalaires deux à deux d'une base orthonormée, appelée base POD, obtenue, à partir de la base classique donnée par une décomposition en éléments finis : elle y porte le nom de matrice de masse.

5.1.2 Motivations

Nous nous intéressons au problème d'approximation matricielle suivant : étant donnée une matrice symétrique A, résoudre

$$\mu^* = \min \frac{1}{2}\|A - X\|_F^2 \quad \text{tel que} \quad \text{diag}\, X = e,\, X \in \mathcal{S}^n,\, X \succeq 0. \tag{5.1}$$

Nous rappellons que $\|A\|_F = \text{ trace } (A^T A)^{1/2}$ désigne la norme de Fröbenius précédemment définie.

Ce problème provient d'applications en Statistiques, où une matrice de corrélation obtenue par calculs peut s'avérer ne plus l'être. Ceci peut être dû à des erreurs de mesure, des erreurs d'arrondis, des données manquantes. On pourra consulter à ce propos le site internet :

"http ://www.ssicentral.com/lisrel/posdef.htm".

En particulier, ce problème se pose en Finance, lorsque l'on fait de l'analyse de risques financiers. En Bourse, on appelle portefeuille un ensemble de n actions cotées. Du point de vue des Statistiques, ces actions sont des variables aléatoires, dont l'univers est par exemple les différentes cotations de ces actions. Suivant le modèle de Markovitz [49], le risque financier que l'on prend en investissant dans un portefeuille de n actions dépend de la matrice de corrélation associée aux différentes actions de ce portefeuille. Toutefois, il arrive très souvent que les données concernant une action ne soient pas accessibles ou pas totalement accessibles sur une période donnée. En conséquence, la matrice effectivement obtenue n'est pas une matrice de corrélation, parce qu'elle possède en général des valeurs propres négatives. Cela implique des erreurs dans le modèle. Pour y remédier, on se propose de chercher la matrice de corrélation la plus proche de la matrice effectivement calculée. Pour cela, on doit résoudre le poblème (5.1).

Cette idée a été mise en œuvre ces dernières années, souvent sous le nom de processus de *calibration de matrices*. Il y a eu de nombreuses tentatives algorithmiques pour résoudre ce problème. Ces algorithmes suivent les différentes approches que nous avons présentées au début de cette thèse. Nous avions commencé la mise en œuvre de l'approche par projections alternées de Boyle-Dykstra, lorsque nous avons été informé de l'existence d'un travail en parallèle effectué par HIGHAM [75] qui donnait des résultats probants. Nous sommes donc passés à l'approche via l'optimisation SDP, en collaboration avec ANJOS, HIGHAM et WOLKOWICZ. Ceci a donné lieu à des travaux [9] qui consistent en l'essentiel de ce chapitre. Parallèlement, l'approche conique *duale* a été mise en œuvre par MALICK [88].

5.1.3 Existence et unicité de solutions

Nous commençons notre étude du problème d'approximation par matrices de corrélation par l'aspect existence et unicité de solution. Cette question, comme c'était le cas pour les matrices bistochastiques, peut être tranchée grâce aux Théorème de projection 2.1.1. Puisque l'elliptope est un ensemble convexe compact, ce théorème s'applique. Il assure l'existence et l'unicité d'une solution optimale au problème (5.1), et fournit une caractérisation de la solution optimale.

Toutefois, nous ne nous sommes pas intéressé plus avant à cette caractérisation de la solution optimale. Du fait de l'expérience acquise avec les matrices bistochastiques, nous ne pensions pas que cette caractérisation fut exploitable. Nous nous sommes donc toute de suite tourné vers les différentes possibilités algorithmiques de calculer cette solution optimale. Néanmoins un tel travail a été effectué dans [75] où le fait qu'il n'est pas possible d'espérer une solution explicite à partir des caractérisations fournies par le Théorème de projection est justifié.

5.2 Approches de types projections

Dans un précédent travail (au chapitre 3), nous avons mis en lumière trois approches de résolution des problèmes d'approximation matricielle linéaires coniques utilisant elles les projections sur des convexes simples : celle par projections alternées, celle par points fixes que nous n'évoquons plus, et celle par algorithme conique dual. La dernière a été mise en œuvre, comme nous l'avons déjà dit, par MALICK [88].

On peut remarquer que l'elliptope $\mathcal{E}$ peut s'écrire comme l'intersection de deux convexes :

- le cône convexe fermé des matrices carrées symétriques semi-définies positives $\mathcal{S}_n^+$,
- le sous-espace affine $\mathcal{U}$ des matrices carrées dont tous les termes diagonaux sont égaux à 1.

On peut donc appliquer l'algorithme par projections alternées de Boyle-Dykstra que nous avons décrit eu deuxième chapitre. Pour ce faire, nous devons calculer explicitement les projections sur $\mathcal{S}_n^+$ et $\mathcal{U}$.

5.2.1 Projection sur $\mathcal{S}_n^+$

La projection d'une matrice carrée symétrique quelconque X sur le cône convexe fermé des matrices semi-définies positives est donnée par la proposition 2.1.5.

Proposition 5.2.1

$$\mathcal{P}_{\mathcal{S}_n}(X) = U^T \begin{pmatrix} \max(\lambda_1, 0) & 0 & 0 & 0 \\ 0 & \max(\lambda_2, 0) & 0 & 0 \\ \vdots & & \ddots & 0 \\ 0 & 0 & \dots & \max(\lambda_n, 0) \end{pmatrix} U, \tag{5.2}$$

où $X = U^T D U$, avec $U^T U = I_n$ et D diagonale, est une diagonalisation de X. □

On pourra se referer à [74]par exemple pour une preuve de ce résultat.

5.2.2 Projection sur $\mathcal{U}$

Pour obtenir le projeté d'une matrice symétrique quelconque X sur le sous-espace $\mathcal{U}$, nous allons procéder de la même manière qu'au chapitre 3 (voir section 3.3.2). Notons $\overline{X}$ le projeté de X sur $\mathcal{U}$. Nous avons la caractérisation suivante :

$$\begin{cases} \overline{X} \in \mathcal{U}, \\ X - \overline{X} \in \mathcal{U}^\perp. \end{cases} \tag{5.3}$$

Notons d'abord que nous avons

$$\mathcal{U} = \{X \in \mathcal{S}_n \mid \text{ diag}\,(X) = e\}, \tag{5.4}$$

alors, on a :

$$\mathcal{U}^\perp = (\text{Ker}(\text{ diag}))^\perp = \text{Im}\,[(\text{ diag})^*]\,. \tag{5.5}$$

Proposition 5.2.2

$$\mathcal{U}^\perp = \mathcal{D}_n \text{ : sous-espace des matrices carrées diagonales.} \tag{5.6}$$

En effet, si nous introduisons l'opérateur linéaire suivant $\text{Diag} : \mathbb{R}^n \to \mathcal{S}_n$ tel que

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \mapsto \begin{pmatrix} x_1 & 0 & 0 & 0 \\ 0 & x_2 & 0 & 0 \\ \vdots & & \ddots & 0 \\ 0 & 0 & \dots & x_n \end{pmatrix},$$

il vient bien que

$$\text{diag}^* = \text{ Diag}\,.$$

Par suite,

$$\text{Im}\,[(\text{ diag})^*] = \text{Im}\,[\text{ Diag}\,] = \mathcal{D}_n.$$

On déduit alors de (5.3) que l'on a la caractérisation équivalente suivante de $\overline{X}$:

$$\begin{cases} \text{diag}\,\overline{X} = e, \\ X - \overline{X} \text{ diagonale.} \end{cases} \tag{5.7}$$

Introduisons cette fois-ci l'opérateur linéaire offDiag défini par

$$\text{offDiag}\,(S) = S - \text{Diag}\,(\,\text{diag}\,(S)), \quad \forall S \in \mathcal{S}_n.$$

L'opérateur offDiag (S) est juste la matrice de diagonale nulle des termes non diagonaux de S, et on peut remarquer que si D est une matrice diagonale, offDiag $(D) = 0_{\mathcal{S}_n}$. Il vient alors immédiatement de (5.7) que :

$$\text{offDiag}\,(X) = \text{offDiag}\,(\overline{X})$$

Et, puisque Diag $(e) = I_n$, la propostion suivante est immédiate.

Proposition 5.2.3

$$\forall X \in \mathcal{S}_n, \qquad \overline{X} := \mathcal{P}_{\mathcal{U}}(X) = \textit{offDiag}\,(X) + I_n. \tag{5.8}$$

5.2.3 Algorithme de projections alternées

Nous pouvons donc, dans les mêmes conditions qu'au chapitre 3, proposer l'algorithme suivant pour la résolution par projections alternées du problème (5.1).

Algorithme 5.2.1

Initialisation $B^0 = A$
$Q^0 = 0$
Précision ε

Itération
$A^{k+1} = \textit{offDiag}\,(B^k) + I_n \quad [= \mathcal{P}_{\mathcal{U}}(B^k)]$
$B^{k+1} = \mathcal{P}_{\mathcal{S}_n^+}(A^{k+1})$
$Q^{k+1} = (A^{k+1} + Q^k) - B^{k+1}$

Test d'arrêt *si* $\|A^{k+1} - B^{k+1}\|_F < \varepsilon$ *Stop,*
sinon retour à Itération,

où A est la matrice que l'on cherche à approcher par une matrice de corrélation.

On peut faire une première remarque sur cet algorithme. La difficulté éventuelle dans sa mise en œuvre pratique proviendra selon toute vraisemblance de la projection sur $\mathcal{S}_n^+$. En effet, celle sur $\mathcal{U}$ ne nécessite pour son calcul qu'une extraction de termes hors diagonaux d'une matrice et une somme de matrices. Effectuer ces opérations ne posent aucun problème sous Matlab, quelle que soit la taille des matrices. Par contre, la projection sur $\mathcal{S}_n^+$ nécessite une décomposition en valeurs propres, un tri des valeurs propres et un changement de base de celle des valeurs propres vers la canonique. Toutes ces opérations sont coûteuses avec Matlab, et d'autant plus que la taille de la matrice augmente. De plus, lorsqu'on a des matrices de grande taille, du fait des erreurs d'arrondis, le tri parmi les valeurs propres peut

s'avérer hasardeux, or l'exactitude de ce tri est primordiale pour le calcul exact du projeté sur $\mathcal{S}_n^+$, et donc la convergence de l'algorithme.

Nous avons eu connaissance à ce moment-là de l'existence d'un travail analogue effectué par HIGHAM. En effet, dans [75], il résout, par projections alternées, un problème d'approximation par matrices de corrélation, pour lequel les normes considérées sont des pondérations de la norme de Fröbenius. Notre problème apparaît comme un cas particulier. Il a fait les mêmes remarques que celles que nous avons faites au sujet de la projection sur $\mathcal{S}_n^+$. Pour contourner ces difficultés, il exploite d'abord le fait qu'en pratique les matrices A que l'on cherche à approcher sont telles que $A \in U$ et toutes ses composantes sont plus petites en valeurs absolues que 1. Grâce à cela, on obtient une estimation (des bornes supérieures et inférieures) sur la valeur optimale du problème (5.1), et surtout, on montre qu'il y a *au moins autant de valeurs propres de la solution optimale nulles que de valeurs propres négatives de* A. D'autre part, lorsque la matrice est de trop grande taille, il se ramène à utiliser, via un interface MEX, des routines de noyau LAPACK de Matlab plus spécialisés, et plus efficaces, car écrit en *fortran* ou C/C^{++}, que la routine de diagonalisation par défaut de Matlab. C'est ainsi que, HIGHAM a pu résoudre des problèmes avec des matrices de taille allant jusqu'à 1399.

5.3 Approche de résolution par minimisation autoduale

5.3.1 Un problème équivalent : Passage à l'épigraphe

Rappelons que pour une fonction convexe $f : \mathbb{H} \to \mathbb{R} \cup \{+\infty\}$, on appelle **épigraphe** de f, et on note epi(f) l'ensemble convexe suivant :

$$\text{epi}(f) = \{(x, \alpha) \in \mathbb{H} \times \mathbb{R} \mid f(x) \leq \alpha\}.$$

Une des propriétés de l'épigraphe est que lorsque l'on veut minimiser la fonction f sur $\mathbb{H}$, on peut se ramener à minimiser le réel α sous la condition que (x, α) soit dans l'épigraphe de f. Cela permet de se ramener à un problème dont la fonction-objectif est linéaire et de faire passer la fonction-objectif originale en contraintes. Cette idée est utilisée en général lorsque la fonction-objectif est la source de complication du problème d'optimisation. On peut considérer que c'est le cas pour le problème (5.1), puisque, si la fonction-objectif était linéaire, on aurait un problème classique d'optimisation SDP. De plus, on sait que les contraintes de type quadratiques peuvent se réexprimer sous la forme de contraintes SDP.

On peut donc réécrire le problème (5.1) sous la forme suivante :

$$\begin{array}{rlll} \sqrt{2\mu^*} = & \min & \alpha & \\ & \text{tq} & \operatorname{diag} X = e, & \\ & & Y + X = A,\ \|Y\|_F \leq \alpha, & \\ & & X, Y \in \mathcal{S}^n\,, X \succeq 0. & \end{array} \tag{5.9}$$

Notre problème apparaît alors comme un problème d'optimisation sur l'intersection d'un cône du second ordre et du cône $\mathcal{S}_n^+$. On peut alors le résoudre directement, puisqu'il existe de nombreux codes du domaine public qui peuvent permettre

de résoudre (5.9). Un certain nombre de ces codes sont accessibles via le serveur NEOS [59] à l'adresse

`http ://www-neos.mcs.anl.gov/`.

On peut aussi consulter la page web de C. HELMBERG à l'adresse :

`http ://www.zib.de/helmberg/semidef.html`.

5.3.2 Tests numériques avec SeDuMi

Nous avons choisi (parmi les codes du domaine public accessibles par NEOS) de résoudre le problème en utilisant le code SeDuMi dû à J. STURM [72],[105]. Ce code utilise les techniques de plongement auto-dual (*self-dual embedding*, en anglais) pour l'optimisation sur les cônes homogènes autoduaux. Ces techniques permettent de résoudre des problèmes d'optimisation en donnant comme résultat soit une solution optimale, soit une preuve de non-réalisabilité du problème, en utilisant notamment un lemme de Farkas. On pourra se référer à [48]. L'algorithme implémenté en pratique est un algorithme de type points intérieurs avec directions de recherche de Newton, dont on peut montrer qu'il converge en $O(\sqrt{n}\ln(\varepsilon))$ itérations dans le pire des cas. C'est un algorithme qui tente d'exploiter les systèmes linéaires creux, comme par exemple lorsqu'on a un grand nombre de variables matricielles de petites dimensions. Par contre, lorsque ceux-ci sont de grande taille (et ne sont pas diagonaux par blocs), l'algorithme est lent, et très coûteux en mémoire.

Pour le problème (5.9), à chaque itération, le travail principal consiste à former et résoudre un système linéaire (souvent dense) de type complément de Schur dont la solution donne la direction de recherche de Newton. Ce système, dont la taille est déterminée par les $n + \left(\begin{array}{c} n+1 \\ 2 \end{array}\right)$ contraintes d'égalité, est de taille de l'ordre de n^2. De plus, on retrouve ici les inconvénients des directions de Newton que nous avons évoqués au chapitre précédent, tels que des systèmes mal conditionnés quand on approche de l'optimum.

Les premiers résultats sont résumés dans le tableau 5.1 ci-après.

On peut remarquer que l'on est très vite limité par la taille des matrices et le temps CPU nécessaire à la résolution du problème. Toutefois, comme cela est observés avec les méthodes de points intérieurs, le nombre d'itérations est pratiquement constant. C'est le temps de calcul nécessaire qui est influencé par la taille de la matrice, sans pour autant l'être par sa singularité, et sa progression semble exponentielle comme le montre la figure 5.1.

Rappelons que les problèmes pratiques que nous espérons résoudre sont de tailles de l'ordre de 1000. Il est clair que nous n'avons aucun espoir de les résoudre par SeDuMi.

5.4 Approche de résolution par points intérieurs

Compte tenu des limites du logiciel SeDuMi, nous nous proposons d'écrire un algorithme de points intérieurs adapté à notre problème qui nous permette de ré-

Taille de A	Rang de A	Temps CPU (en secondes)	Nombre d'itérations	Temps CPU moyen par itération
50	5	151	16	9.44
50	10	149	16	9.31
50	20	171	16	10.7
60	6	594	15	39.6
60	20	672	17	39.5
60	50	711	18	39.5
70	7	2193	15	146.2
70	15	1781	16	111.3
70	50	1894	17	111.4
80	8	5471	16	341.9
80	20	4790	16	299.4
80	50	4350	16	271.9
90	20	10904	15	726.9

TAB. 5.1 – Résultats pour l'approche par SeDuMi pour des matrices A générées aléatoirement

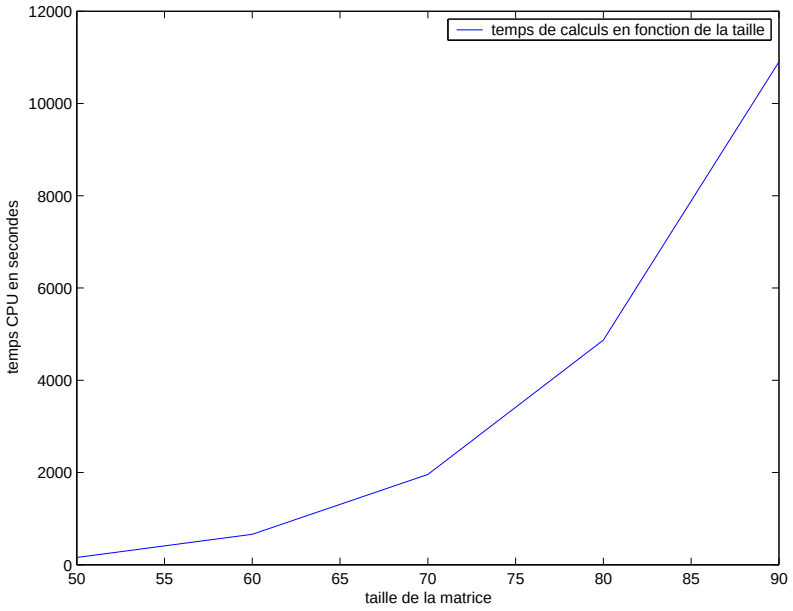


FIG. 5.1 –

soudre des problèmes de plus grande taille. Cet algorithme suivra la démarche que nous avons proposée en fin du chapitre précédent (section 4.4). Nous utiliserons une condition d'optimalité bilinéaire, dont la linéarisation conduit à des systèmes linéaires qui ont le même ordre de taille qu'avec SeDuMi mais qui sont creux, n'ont pas à être construit explicitement et sont de rang maximal à l'optimum. Ces systèmes seront résolus par gradients conjugués préconditionnés. Enfin, une étape de "crossover" sera introduite en fin d'algorithme afin de récupérer une convergence asymptotique q-quadratique.

Nous avions vu que l'algorithme de points intérieurs que nous nous proposons d'écrire serait particulièrement performant si l'on pouvait écrire les contraintes affines sous la forme d'opérateurs, dont on peut facilement calculer les adjoints, et pseudo-inverses. Nous introduisons, dans cet ordre d'idées, quelques opérateurs linéaires sur les matrices qui vont nous être utiles.

5.4.1 Quelques opérateurs

Pour une matrice $M = [m_1\, m_2\, \ldots\, m_n] \in \mathbb{R}^{m\times n}$, ($m_i \in \mathbb{R}^m, i = 1, 2, \ldots, n$),

$$v = \text{ vec}\,(M) := \begin{pmatrix} m_1 \\ m_2 \\ \vdots \\ m_n \end{pmatrix} \in \mathbb{R}^{mn}$$

est le vecteur formé en mettant les colonnes de M bout à bout. On définit ainsi l'opérateur vec dont l'inverse et l'adjoint sont donnés par

$$\text{Mat } = \text{ vec}^{-1} = \text{ vec}^*,$$

en utilisant la définition de l'adjoint d'un opérateur : $\langle \text{ vec}\,(M), u\rangle = \langle M,\ \text{vec}^*(u)\rangle$. Mat construit une matrice $m\times n$, colonne par colonne, à partir d'un vecteur de taille mn. Les opérateurs Mat et vec sont des isométries.

Pour $X \in \mathcal{S}_n$, soit $x = \text{ us2vec}\, X \in \mathbb{R}^{\binom{n}{2}}$ qui est construit en multipliant par $\sqrt{2}$, le vecteur obtenu en mettant bout à bout les termes situés strictement au dessus de la diagonale de X et considérés colonne par colonne :

$$\text{us2vec} : X = \begin{pmatrix} X_{11} & X_{12} & \ldots & X_{1n} \\ & X_{22} & \ldots & X_{2n} \\ & & \ddots & \vdots \\ & & & X_{nn} \end{pmatrix} \mapsto x = \sqrt{2}\begin{pmatrix} X_{12} \\ X_{13} \\ \vdots \\ X_{(n-1),n} \end{pmatrix}. \quad (5.10)$$

Le coefficient $\sqrt{2}$ assure que l'on a une isométrie.

Soit us2Mat := us2vec^{-1} l'opérateur inverse de us2vec, défini sur $\mathbb{R}^{\binom{n}{2}}$ à valeurs dans le sous-espace $\mathcal{S}_n^0$ des matrices de $\mathcal{S}_n$ dont tous les termes diagonaux sont nuls. On a :

$$\text{us2Mat}^* = \text{ us2vec}, \quad (5.11)$$

puisque

$$\begin{aligned}\langle \text{ us2Mat}\,(v), S\rangle &= \text{ trace us2Mat}\,(v)S\\ &= \text{ trace us2Mat}\,(v)\ \text{offDiag}\,(S)\\ &= v^T\ \text{us2vec}\,(S) = \langle\ \text{us2vec}\,(S), v\rangle\,.\end{aligned}$$

Ainsi,

$$\text{us2Mat us2Mat}^*(S) = \text{ offDiag}\,(S) = \text{ offDiag}^*(S)$$

est la projection orthogonale sur le sous-espace $\mathcal{S}_n^0$. Ceci confirme la proposition 5.2.3 puisqu'on a :

$$\mathcal{U} = I_n + \mathcal{S}_n^0.$$

Notre algorithme utilisera les opérateurs définis comme suit. Soit

$$X := A + \text{ us2Mat}\,(s) + I, \quad S^y := \text{ us2Mat}\,(s) + \text{ Diag}\,(y),$$

pour des vecteurs s and y judicieusement choisis. On définit les opérateurs linéaires suivants :

$$\mathcal{X}_u(\cdot) := X\ \text{us2Mat}\,(\cdot);\ \mathcal{X}_d(\cdot) := X\ \text{Diag}\,(\cdot);\ \mathcal{S}(\cdot) := \text{ us2Mat}\,(\cdot)S^y. \tag{5.12}$$

Ainsi,

$$\mathcal{X}_u : \mathbb{R}^{\binom{n}{2}} \to \mathcal{M}^n; \quad \mathcal{X}_d : \mathbb{R}^n \to \mathcal{M}^n; \quad \mathcal{S} : \mathbb{R}^{\binom{n}{2}} \to \mathcal{M}^n\,.$$

Nous aurons besoin des adjoints de ces opérateurs. Soit $v = \text{ us2vec}\,(V),\ V \in \mathcal{S}^n\,, z \in \mathbb{R}^n$, et $W \in \mathcal{M}^n$.

$$\begin{aligned}\langle W, \mathcal{X}_u(v)\rangle &= \text{tr}\mathbf{W}^T X\ \text{us2Mat}\,(v),\\ &= \text{tr us2Mat}\,(v)XW,\\ &= \left\langle \text{ us2Mat}\,(v), \frac{1}{2}(XW + W^TX)\right\rangle,\\ &= \left\langle v, \frac{1}{2}\ \text{us2vec}\,(XW + W^TX)\right\rangle,\\ &= \langle v, \mathcal{X}_u{}^*(W)\rangle\,.\end{aligned}$$

Par suite,

$$\mathcal{X}_u{}^*(W) = \frac{1}{2}\ \text{us2vec}\,(XW + W^TX).$$

$$\begin{aligned}\langle W, \mathcal{X}_d(z)\rangle\,, &= \text{tr}\mathbf{W}^T X\ \text{Diag}\,(z),\\ &= \text{tr Diag}\,(z)W^TX,\\ &= z^T\ \text{diag}\,(W^TX),\\ &= \langle \mathcal{X}_d{}^*(W), z\rangle\,,\end{aligned}$$

d'où,

$$\mathcal{X}_d{}^*(W) = \text{ diag}\,(W^T X),$$

ou bien

$$\mathcal{X}_d{}^*(W) = (W \circ X)^T e.$$

$$\begin{aligned}
\langle W, \mathcal{S}(v)\rangle &= \text{tr}\mathbf{W}^T \text{ us2Mat}\,(v) S^y, \\
&= \left\langle \text{ us2Mat}\,(v), \frac{1}{2}(WS^y + S^y W^T) \right\rangle, \\
&= \left\langle v, \frac{1}{2} \text{ us2vec}\,(WS^y + S^y W^T) \right\rangle, \\
&= \langle v, \mathcal{S}^*(W)\rangle\,.
\end{aligned}$$

On a donc

$$\mathcal{S}^*(W) = \frac{1}{2} \text{ us2vec}\,(WS^y + S^y W^T).$$

Nous aurons aussi besoin de différentes compositions d'opérateurs :

$$\begin{aligned}
\mathcal{X}_d{}^*\mathcal{X}_d(z) &= \mathcal{X}_d{}^*(X \text{ Diag}\,(z)) \\
&= \text{ diag}\,(\text{ Diag}\,(z) X^2);
\end{aligned}$$

$$\begin{aligned}
\mathcal{X}_d{}^*\mathcal{X}_u(v) &= \mathcal{X}_d{}^*\left(X \text{ us2Mat}\,(v)\right), \\
&= \text{ diag}\,\left(\text{ us2Mat}\,(v) X^2\right);
\end{aligned}$$

$$\begin{aligned}
\mathcal{X}_d{}^*\mathcal{S}(v) &= \mathcal{X}_d{}^*(\text{ us2Mat}\,(v) S^y) \\
&= \text{ diag}\,\left(S^y \text{ us2Mat}\,(v) X\right);
\end{aligned}$$

$$\begin{aligned}
\mathcal{X}_u{}^*\mathcal{X}_d(z) &= \mathcal{X}_u{}^*\left(X \text{ Diag}\,(z)\right), \\
&= \frac{1}{2} \text{ us2vec}\,\left(X^2 \text{ Diag}\,(z) + \text{ Diag}\,(z) X^2\right);
\end{aligned}$$

$$\begin{aligned}
\mathcal{X}_u{}^*\mathcal{X}_u(v) &= \mathcal{X}_u{}^*\left(X \text{ us2Mat}\,(v)\right), \\
&= \frac{1}{2} \text{ us2vec}\,\left(X^2 \text{ us2Mat}\,(v) + \text{ us2Mat}\,(v) X^2\right);
\end{aligned}$$

$$\begin{aligned}
\mathcal{X}_u{}^*\mathcal{S}(v) &= \mathcal{X}_u{}^*(\text{ us2Mat}\,(v) S^y), \\
&= \frac{1}{2} \text{ us2vec}\,\left(X \text{ us2Mat}\,(v) S^y + S^y \text{ us2Mat}\,(v) X\right), \\
&= \mathcal{S}^* \mathcal{X}_u(v); \\
\mathcal{S}^*\mathcal{X}_u(v) &= \mathcal{S}^*(X \text{ us2Mat}\,(v)), \\
&= \frac{1}{2} \text{ us2vec}\,\left(X \text{ us2Mat}\,(v) S^y + S^y \text{ us2Mat}\,(v) X\right);
\end{aligned}$$

$$\begin{aligned}\mathcal{S}^*\mathcal{X}_d(z) &= \mathcal{S}^*(X\ \mathrm{Diag}\,(z)),\\ &= \frac{1}{2}\,\mathrm{us2vec}\,(X\ \mathrm{Diag}\,(z)S^y + S^y\ \mathrm{Diag}\,(z)X)\,;\end{aligned}$$

$$\begin{aligned}\mathcal{S}^*\mathcal{S}(v) &= \mathcal{S}^*(\ \mathrm{vec}\,(\ \mathrm{us2Mat}\,(v)S^y)),\\ &= \frac{1}{2}\,\mathrm{us2vec}\,\left(\ \mathrm{us2Mat}\,(v)(S^y)^2 + (S^y)^2\ \mathrm{us2Mat}\,(v)\right).\end{aligned}$$

$$\begin{aligned}(\mathcal{X}_u{}^* + \mathcal{S}^*)\,(\mathcal{X}_u + \mathcal{S})\,(v) &= \frac{1}{2}\,\mathrm{us2vec}\,\left[\ \mathrm{us2Mat}\,(v)\left(X^2 + (S^y)^2\right) + \left(X^2 + (S^y)^2\right)\ \mathrm{us2Mat}\,(v)\right],\\ &+ \ \mathrm{us2vec}\,[X\ \mathrm{us2Mat}\,(v)S^y + S^y\ \mathrm{us2Mat}\,(v)X]\,.\end{aligned}$$

Proposition 5.4.1 *Nous obtenons le formulaire suivant pour les opérateurs définis en* (5.12) *:*

$$\begin{aligned}\mathcal{X}_d(\cdot) &= X\ \mathit{Diag}\,(\cdot)\\ \mathcal{X}_u(\cdot) &= X\ \mathit{us2Mat}\,(\cdot)\\ \mathcal{S}(\cdot) &= \ \mathit{us2Mat}\,(\cdot)S^y\\ \mathcal{X}_d{}^*(W) &= (W\circ X)^T e = \ \mathit{diag}\,(W^TX)\\ \mathcal{X}_u{}^*(W) &= \tfrac{1}{2}\ \mathit{us2vec}\,(XW + W^TX)\\ \mathcal{S}^*(W) &= \tfrac{1}{2}\ \mathit{us2vec}\,(WS^y + S^yW^T)\\ \mathcal{X}_d{}^*\mathcal{X}_d(z) &= \ \mathit{diag}\,(\ \mathit{Diag}\,(z)X^2)\\ \mathcal{X}_d{}^*\mathcal{X}_u(v) &= \ \mathit{diag}\ (\ \mathit{us2Mat}\,(v)X^2)\\ \mathcal{X}_d{}^*\mathcal{S}(v) &= \ \mathit{diag}\ (S^y\ \mathit{us2Mat}\,(v)X)\\ \mathcal{X}_u{}^*\mathcal{X}_d(z) &= \tfrac{1}{2}\ \mathit{us2vec}\ (X^2\ \mathit{Diag}\,(z) + \ \mathit{Diag}\,(z)X^2)\\ \mathcal{X}_u{}^*\mathcal{X}_u(v) &= \tfrac{1}{2}\ \mathit{us2vec}\ (X^2\ \mathit{us2Mat}\,(v) + \ \mathit{us2Mat}\,(v)X^2)\\ \mathcal{X}_u{}^*\mathcal{S}(v) &= \tfrac{1}{2}\ \mathit{us2vec}\ (X\ \mathit{us2Mat}\,(v)S^y + S^y\ \mathit{us2Mat}\,(v)X)\\ \mathcal{S}^*\mathcal{X}_u(v) &= \tfrac{1}{2}\ \mathit{us2vec}\ (X\ \mathit{us2Mat}\,(v)S^y + S^y\ \mathit{us2Mat}\,(v)X)\\ \mathcal{S}^*\mathcal{X}_d(z) &= \tfrac{1}{2}\ \mathit{us2vec}\ (X\ \mathit{Diag}\,(z)S^y + S^y\ \mathit{Diag}\,(z)X)\\ \mathcal{S}^*\mathcal{S}(v) &= \tfrac{1}{2}\ \mathit{us2vec}\ (\ \mathit{us2Mat}\,(v)(S^y)^2 + (S^y)^2\ \mathit{us2Mat}\,(v))\end{aligned}$$

5.4.2 Deuxième formulation équivalente

Introduisons les notations suivantes :

$$a = \ \mathrm{us2vec}\,A,\ s = \ \mathrm{us2vec}\,S$$

analogues à $x = \ \mathrm{us2vec}\,X$ que nous avions introduit précédemment.

De plus, puisque les termes diagonaux de X sont constants de même que ceux de A, leurs contributions à la norme $\|A - X\|$ reste constante. Sans perte de généralité, nous pouvons supposer désormais :

$$\mathrm{diag}\,(A) = 0.$$

Notons que ceci implique

$$a = \ \mathrm{us2vec}\,A \Leftrightarrow A = \ \mathrm{us2Mat}\,a,$$

ce qui n'est pas le cas en général, et aussi

$$\|X - A\|_F^2 = \|x - a\|_2^2 + n.$$

Afin de résoudre le problème (5.1), nous pouvons le reformuler sous la forme suivante :

$$\mu^* := \min \frac{1}{2}\|x - a\|_2^2 \quad \text{tel que} \quad \text{us2Mat}\,(x) + I \succeq 0,\, x \in \mathbb{R}^{\binom{n}{2}}, \qquad (5.13)$$

en écrivant $X = \text{ us2Mat}\,(x) + I$ dans (5.1). Cette forme est plus adaptée que la précédente à notre démarche algorithmique.

5.4.3 Conditions d'optimalité et Directions de recherche

Pour obtenir les conditions d'optimalité pour (5.13), nous en explicitons d'abord le problème dual. Notons que les contraintes de (5.13) sont qualifiées au sens de Slater (voir 1.4.2), ce qui implique qu'il y aura dualité forte pour notre dual lagrangien. :

$$\mu^* = \nu^* := \max_{S \succeq 0} \min_x \frac{1}{2}||x - a||^2 - \text{ trace}\, S(\text{ us2Mat}\,(x) + I).$$

En procédant de manière classique, on associe à la contrainte $\text{ us2Mat}\,(x) + I \succeq 0$ un multiplicateur de Lagrange $S \in \mathcal{S}_n^+$, puisque $\mathcal{S}_n^+$ est auto-dual. On construit alors le lagrangien :

$$L(x, S) = \min_{x \in \mathbb{R}^{\binom{n}{2}}} f(x) = \frac{1}{2}||x - a||^2 - \text{ trace}\, S(\text{ us2Mat}\,(x) + I) \qquad (5.14)$$

Ce problème est finalement un problème sans contraintes. Sa fonction-objectif s'écrit :

$$\begin{aligned} f(x) &= \frac{1}{2}||x - a||^2 - \text{ trace}\, S(\text{ us2Mat}\,(x) + I), \\ &= \frac{1}{2}||x - a||^2 - \text{ trace}\,[S(\text{ us2Mat}\,(x))] - \text{ trace}\,(S), \\ &= \frac{1}{2}||x - a||^2 - \langle S, \text{ us2Mat}\,(x)\rangle - \text{ trace}\,(S), \\ &= \frac{1}{2}||x - a||^2 - \langle \text{ us2Mat}^*(S), x\rangle - \text{ trace}\,(S). \end{aligned}$$

Elle est différentiable de manière évidente. Les solutions optimales de (5.14) sont donc caractérisées par :

$$\begin{aligned} 0 &= \nabla f(x), \\ &= (x - a) - \text{ us2Mat}^*(S), \\ &= (x - a) - \text{ us2vec}\,(S). \end{aligned}$$

Nous obtenons le problème dual suivant :

$$\begin{array}{rll} \mu^* = & \max & \frac{1}{2}||x-a||^2 - \text{ trace } S(\text{ us2Mat }(x)+I) \\ & \text{tel que} & x - \text{ us2vec }(S) = a, \\ & & S \succeq 0. \end{array} \tag{5.15}$$

En écrivant S sous la forme

$$S = \text{ us2Mat }(s) + \text{ Diag }(y), \quad s \in \mathbb{R}^{\binom{n}{2}}, y \in \mathbb{R}^n,$$

et en remarquant que

$$x = \text{ us2vec }(S) + a = s + a,$$

la fonction-objectif de (5.15) s'écrit :

$$\begin{aligned} f(x) &= \frac{1}{2}\|s\|^2 - \text{ trace }(S \text{ us2Mat }(x)) - \text{ trace }(S), \\ &= \frac{1}{2}\|s\|^2 - \langle S, \text{ us2Mat }(x)\rangle - \text{ trace }(\text{ us2Mat }(s) + \text{ Diag }(y)), \\ &= \frac{1}{2}\|s\|^2 - \langle \text{ us2vec }(S), x\rangle - y^T e, \\ &= \frac{1}{2}\|s\|^2 - \langle s, s+a\rangle - y^T e, \\ &= \frac{1}{2}\|a\|^2 - \frac{1}{2}\|s-a\|^2 - y^T e. \end{aligned}$$

On peut écrire le problème dual (5.15) sous la forme équivalente :

$$\begin{array}{rll} \mu^* = \frac{1}{2}\|a\|^2 + & \max & -(\frac{1}{2}\|s-a\|^2 + y^T e) \\ & \text{t.q. } S^y := \text{ us2Mat }(s) + \text{ Diag }(y) \succeq 0. & \end{array} \tag{5.16}$$

Puisque les conditions de qualification de contraintes de Slater sont vérifiées pour le problème dual aussi, nous obtenons les conditions d'optimalité primales-duales suivantes :

Théorème 5.4.2 *Les valeurs optimales primales et duales sont égales, $\mu^* = \nu^*$, et les paires primales duales $(x,(y,s))$ sont optimales pour* (5.13) *si et seulement si :*

$$\begin{array}{rcll} X & := & \textit{us2Mat}\,(x) + I \succeq 0 & \textit{(réalisabilité primale)} \\ x & = & a+s, \;\; S^y := \textit{us2Mat}\,(s) + \textit{Diag}\,(y) \succeq 0 & \textit{(réalisabilité duale)} \\ XS^y & = & 0 & \textit{(écarts complémentaires).} \end{array}$$

■

Pour la mise en œuvre de notre algorithme primal-dual de points "intérieurs-extérieurs", nous utilisons la perturbation classique de l'équation des écarts complémentaires suivante :

$$XS^y = \mu I. \tag{5.17}$$

Comme nous l'avons décrit au précédent chapitre 4.4, nous substituons ensuite les équations de réalisabilité primale et duale dans l'équation perturbée ci-dessus (5.17) et nous obtenons une **unique équation bilinéaire** en s et y qui caractérise l'optimalité pour le problème barrière logarithmique que l'on déduit de (5.13).

$$F_\mu(s,y) : \mathbb{R}^{\binom{n+1}{2}} \to \mathcal{M}^n .$$

$$F_\mu(s,y) : \quad = \quad [A + \text{ us2Mat}\,(s) + I]\,[\,\text{us2Mat}\,(s) + \text{ Diag}\,(y)] - \mu I = 0, \tag{5.18}$$

On pourra remarquer que le problème d'approximation par matrices de corrélation original a $\binom{n+1}{2}$ variables, n contraintes d'égalité (sur la diagonale de X) et la contrainte de semi-définie positivité de X. Par suite, le problème dual a $n + \binom{n+1}{2}$ variables. Ainsi, si l'on considérait des algorithmes qui résolvent uniquement le problème dual, on n'aurait pas une diminution de la taille du problème. De plus, avec les algorithmes primaux-duaux standard, on aurait $n + 2\binom{n+1}{2}$ variables, au contraire des $\binom{n+1}{2}$ variables (s et y) que nous avons ici en considérant l'équation bilinéaire (5.18).

Etant donné que cette équation (5.18) est *surdéterminée* (F_μ ne met pas en relation les mêmes ensembles à un isomorphisme près) et non linéaire, nous la résolvons en utilisant une méthode de Gauss-Newton inexacte. Par linéarisation de (5.18), nous obtenons le système linéaire donc la résolution nous donne la direction de recherche $\Delta v = \begin{pmatrix} \Delta s \\ \Delta y \end{pmatrix}$ où nous avons posé $v = \begin{pmatrix} s \\ y \end{pmatrix}$:

$$\begin{aligned}
-F_\mu(s,y) &= F'_\mu(s,y)\Delta v, && (5.19)\\
&= [A + \text{ us2Mat}\,(s) + I]\,(\,\text{us2Mat}\,(\Delta s) + \text{ Diag}\,(\Delta y)) && (5.20)\\
&\qquad + \text{us2Mat}\,(\Delta s) S^y, && (5.21)\\
&= (\mathcal{X}_u + \mathcal{S})\,(\Delta s) + \mathcal{X}_d(\Delta y). && (5.22)
\end{aligned}$$

On retrouve les opérateurs $\mathcal{X}_u$, $\mathcal{S}$ et $\mathcal{X}_d$ que nous avions introduits au paragraphe précédent, et on comprend pourquoi.

Ce système linéaire surdéterminé est de rang maximal. Nous utiliserons sa solution au sens des moindres carrés comme direction de recherche (de Gauss-Newton) dans notre algorithme. Cette solution sera calculée en utilisant une méthode de gradients conjugués, préconditionnée.

Notons que $\Delta s \in \mathbb{R}^{\binom{n}{2}}$, mais, le coût du calcul de $(\mathcal{X}_u + \mathcal{S})\,(\Delta s)$, en ne considérant pas un éventuel caractère creux, est celui de la multiplication de deux matrices symétriques. Le calcul de $\mathcal{X}_d(\Delta y)$ correspond quant à lui à un produit de Hadamard (composantes par composantes) de deux vecteurs de taille n. Ces calculs qui représentent l'essentiel d'une itération de gradients conjugués sont donc pratiquement gratuits.

5.4.4 Algorithme

Nous utilisons l'équation (5.18) pour développer un algorithme primal-dual de points intérieurs-extérieurs réalisable (c'est à dire que l'on part de points strictement réalisables pour le primal et le dual) tel que nous l'avons décrit en section 4.4 du chapitre précédent. Nous utilisons donc l'approche par Gauss-Newton de [84]. Nous introduisons un paramètre de recentrage σ_k au lieu d'une approche prédictrice-correctrice classique. Nous imposons la semi-définie positivité au cours du déroulement de l'algorithme plutôt que la définie positivité. Enfin, dès que nous sommes **suffisamment proches** de l'optimum, nous faisons du "crossover" en posant $\sigma_k = 0$ et $\alpha_k = 1$, et en n'imposant plus la semi-définie positivité des matrices. Ceci conduit à une rapide convergence quadratique asymptotiquement.

Critère de "Crossover"

Il nous faut à présent préciser les modalités pratiques suivant lesquelles l'étape de "crossover" est appliquée. Rappelons qu'il s'agit de ne plus forcer l'algorithme à demeurer réalisable une fois que l'on se trouve dans la région de convergence quadratique de l'optimum. Il nous faut donc un moyen d'estimer rapidement la région de convergence quadratique. Ceci peut être fait en utilisant le Théorème 4.4.4 que nous avons énoncé au chapitre précédent. Toutefois, les estimations du rayon de convergence quadratique fournies par le théorème dépendent de l'optimum du problème qui est inconnu. Il faut donc trouver à partir de ces estimations des heuristiques qui permettent de s'assurer que l'on est dans la région de convergence quadratique. Une heuristique possible est de considérer que le pas courant Δv par exemple, est une bonne approximation de la distance du point courant à l'optimum.

De telles heuristiques ont été étudiées dans [114] pour la résolution de la relaxation SDP d'un problème de *max-cut*. De plus, on peut remarquer que la fonction F bilinéaire d'optimalité obtenue ici est très similaire à celle qui a été obtenue dans [114]. Nous avons donc choisi d'effectuer l'étape de "crossover" dans notre cas, en utilisant le même type d'heuristique. L'étape de "crossover" sera donc déterminée par le critère sur le saut de dualité suivant :

$$\frac{tr(ZX)}{0.5\|A - X\|_F^2 + 1}. \tag{5.23}$$

Notons $\mathcal{F}^0$ l'ensemble des points primaux-duaux strictement réalisables et F' la jacobienne de la fonction F définissant les conditions d'optimalité.

Algorithme 5.4.1 (Points intérieurs-extérieurs par Gauss-Newton (G-N) et "crossover")
• **Initialisation :**

•• **Donnée :** *une matrice carrée symétrique d'ordre* n*,* A*, (fixer* $diag\,(A) = 0$*).*

•• **Tolérances :** ε_1 *(arrêt),* ε_2 *(précision pour G-N),* ε_3 *("crossover").*

•• **Trouver les points initiaux strictement réalisables**
S^0 *et* $X^0 :=$ ($offDiag\,(S^0 + A) + I) \succ 0$ *;* μ *petit*

•• **Fixer les paramètres initiaux :**

$gap := trace\, S^0X^0;\ \ \mu_0 = gap/n;\quad objval := 0.5||X^0 - A||_F^2;\ \ k = 0.$

• **Tant que** $\min\{\frac{gap}{objval+1},\ objval\} > \varepsilon_1$

•• **résoudre au sens des moindres carrés pour obtenir la direction de recherche** Δv^k *(précision* $\varepsilon_2 \min\{\mu_k, 1\}$*)*

$$F'_{\sigma_k\mu_k}(v^k)\Delta v^k = -F_{\sigma_k\mu_k}(v^k),$$

où σ_k *est le paramètre de recentrage,* $\mu_k = \frac{1}{n}\, trace\, S^k($ $offDiag\,(S^k + A) + I)$.

•• **recherche linéaire :**

$$S^{k+1} = S^k + \alpha_k \Delta S^k,\ avec\ \alpha_k > 0,$$

tel que S^{k+1} *et* $offDiag\,(S^{k+1}+A)+I \succeq 0$*, (*$\alpha_k := 1$ *après "crossover".)*

•• **Mise à jour** $k \leftarrow k + 1$

$objval := 0.5||X^{k+1} - A||_F^2, gap := trace\, S^{k+1}X^{k+1}, \mu_{k+1} = gap/n,$

$\sigma_k \quad \left(fixer\ \sigma_k = 0\ si\ \min\{\frac{gap}{objval\ +1},\ objval\} < \varepsilon_3\ (crossover)\right).$

• **fin (tant que)**.
• **Résultat :** $X \approx us2Mat\,(s) + A + I$.

La mise à jour de σ_k ci-dessus est faite de manière adaptative : elle est dépendante des valeurs courantes de X et S. Elle est faite de manière à se recentrer du mieux possible sur le chemin central, tout en évitant de trop se rapprocher du bord.

5.4.5 Préconditionnement

Comme nous l'avons vu au chapitre précédent, le préconditionnement est essentiel pour une résolution efficace du système linéaire (5.22) au sens des moindres carrés. En ce qui nous concerne, effectuer un préconditionnement consiste à trouver deux opérateurs (en pratique des matrices) P_s et P_y et à chercher la solution au sens des moindres carrés de

$$(\mathcal{X}_u + \mathcal{S})\, P_s^{-1}(\widehat{\Delta s}) + \mathcal{X}_d P_y^{-1}(\widehat{\Delta y}) = -F_\mu(s, y), \tag{5.24}$$

où

$$\widehat{\Delta s} = P_s(\Delta s), \quad \widehat{\Delta y} = P_y(\Delta y).$$

Les inverses ci-dessus ne sont pas formées explicitement. De plus, les deux opérateurs P_s et P_y ont des structures assez simples de manière à ce que les systèmes linéaires correspondants soit résolus efficacement.

Pré-conditionnement diagonal

Le pré-conditionnement diagonal a été étudié dans différents ouvrages [51], [101], [66, Sect. 10.5], et [50, Prop. 2.1(v)]. Les résultats diffèrent selon la définition du conditionnement d'une matrice, qui décrit la répartition des valeurs propres de cette matrice. Par exemple, dans [50, Prop. 2.1(v)], on prend la définition suivante du conditionnement d'une matrice $\times n\ K$:

$$\omega(K) := n^{-1} \text{ trace } (K)/\det(K)^{1/n}.$$

On y montre alors que pour une matrice $m \times n\ A$ de plein rang avec $m \geq n$, le pré-conditionneur diagonal optimal, solution du problème d'optimisation

$$\min \omega((AD)^T(AD)) \quad \text{tel que } D \text{ matrice diagonale positive,} \tag{5.25}$$

est donnée par

$$d_{ii} = 1/\|A_{:i}\|_2, i = 1, \ldots, n.$$

Par suite, pour faire un pré-conditionnement diagonal de (5.22), on peut choisir des opérateurs P_y et P_s qui sont diagonaux. Ils sont évalués en utilisant les colonnes de l'opérateur $F'_\mu(s, y)$. Ces colonnes sont de deux types : celles correspondant à s, et celles correspondant à y. Compte tenu de la forme découplée de l'équation (5.22), le calcul de P_y et P_s peut se faire de manière indépendante.

Commençons par le calcul le plus simple, celui de P_y. Nous rappelons que pour évaluer les colonnes d'un opérateur linéaire, il suffit de calculer les images des éléments de la base (canonique) de son espace de départ. Rappelons que l'on a :

$$X = A + \text{ us2Mat } (s) + I \text{ et } S = \text{ us2Mat } (s) + \text{ Diag } (y).$$

Pour toute matrice X, $X_{l:}$ désigne sa $l$ème ligne et $X_{:l}$ désigne sa l ème colonne.

△ **Pré-conditionnement de $\mathcal{X}_d$.** L'opérateur $\mathcal{X}_d$ étant défini sur $\mathbb{R}^n$, il nous suffit de calculer les images des vecteurs $e_i,\ i = 1, \ldots, n$, de la base canonique de $\mathbb{R}^n$. On a :

$$\mathcal{X}_d(e_i) = X \text{ Diag } (e_i).$$

Par suite,

$$\|\mathcal{X}_d(e_i)\|_F^2 = ||X_{i,:}||^2. \tag{5.26}$$

△ **Pré-conditionnement de $\mathcal{X}_u + \mathcal{S}$.** Les deux opérateurs $\mathcal{X}_u$ et $\mathcal{S}$ sont définis sur $\mathbb{R}^{\binom{n}{2}}$. Nous allons évaluer les images des vecteurs $e_k, k = 1, \ldots, \binom{n}{2}$ de la base canonique. A chaque $k = 1, \ldots, n$, on peut associer un unique couple $(i, j),\ i = 1, \ldots, n;\ j = 1, \ldots, n;\ i < j$ tel que lors de l'opération $x = \text{ us2vec } (X)$, l'élément x_k de x est identique à l'élément X_{ij} de X. Dans la suite, e_i et e_j représenteront respectivement le $i$ème et $j$ème vecteur de la base canonique de $\mathbb{R}^n$, tandis que e_k représente

le $k$ème un vecteur de base de $\mathbb{R}^{\left(\begin{array}{c} n \\ 2 \end{array}\right)}$. On a :

$$\begin{aligned} \mathcal{X}_u(e_k) &= X \text{ us2Mat}\,(e_k) \\ &= \tfrac{1}{\sqrt{2}} X \left(e_i e_j^T + e_j e_i^T\right) \\ &= \tfrac{1}{\sqrt{2}} \left(X_{:i} e_j^T + X_{:j} e_i^T\right). \end{aligned}$$

D'autre part,

$$\begin{aligned} \mathcal{S}(e_k) &= \text{us2Mat}\,(e_k)(S + \text{ Diag}\,(y)) \\ &= \tfrac{1}{\sqrt{2}} \left(e_i e_j^T + e_j e_i^T\right)(S + \text{ Diag}\,(y)) \\ &= \tfrac{1}{\sqrt{2}} \left\{(e_i (S + \text{ Diag}\,(y))_{j:} + e_j (S + \text{ Diag}\,(y))_{i:})\right\}. \end{aligned}$$

Par suite,

$$\begin{aligned} \|(\mathcal{X}_u + \mathcal{S})(e_k)\|_F^2 &= \tfrac{1}{2} \{\|(S + \text{ Diag}\,(y))_{:i}\|^2 + \|(S + \text{ Diag}\,(y))_{:j}\|^2 + \\ &\quad \|X_{:i}\|^2 + \|X_{:j}\|^2 + 2(S + \text{ Diag}\,(y))_{jj} X_{ii} \\ &\quad + 4(S + \text{ Diag}\,(y))_{ji} X_{ij} + 2(S + \text{ Diag}\,(y))_{ii} X_{jj}\}. \end{aligned} \tag{5.27}$$

Pour ce calcul, nous avons besoin de trois produits de Hadamard, $X \circ X, (S + \text{ Diag}\,(y)) \circ (S + \text{ Diag}\,(y)), (S + \text{ Diag}\,(y)) \circ X$, et du produit de Kronecker (vectoriel) $\text{Diag}\,((S + \text{ Diag}\,(y))) \otimes \text{ Diag}\,(X)$.

Comme on peut le voir, les pré-conditionneurs diagonaux sont très faciles à calculer en général. Mais, en général, ils sont rarement efficaces, voir par exemple [66].

Pré-conditionneur diagonal par blocs par Cholesky incomplet

En lieu et place du pré-conditionneur diagonal, pour lequel nous n'avons pas beaucoup d'espoirs, nous avons construit un pré-conditioneur diagonal par blocs. Cet choix coule de source en réalité. En effet, l'équation résolue pour obtenir la direction de recherche a naturellement une structure par blocs :

$$[(\mathcal{X}_u + \mathcal{S}) \mid \mathcal{X}_d] \left(\begin{array}{c} \Delta s \\ \Delta y \end{array}\right) = -F_\mu.$$

Puisque la résolution est faite au sens des moindres carrés, on résout effectivement les équations normales :

$$\left[\begin{array}{c|c} (\mathcal{X}_u{}^* + \mathcal{S}^*)(\mathcal{X}_u + \mathcal{S}) & (\mathcal{X}_u{}^* + \mathcal{S}^*)\mathcal{X}_d \\ \hline \mathcal{X}_d{}^*(\mathcal{X}_u + \mathcal{S}) & \mathcal{X}_d{}^* \mathcal{X}_d \end{array}\right] \left(\begin{array}{c} \Delta s \\ \Delta y \end{array}\right) = -\left(\begin{array}{c} \mathcal{X}_u{}^* + \mathcal{S}^* \\ \mathcal{X}_d{}^* \end{array}\right) F_\mu. \tag{5.28}$$

Etant donnée cette structure par blocs, il est naturel de considérer un pré-conditionnement diagonal par blocs. Suivant [66] et [10, Section 9.2] , nous avons proposé d'utiliser un pré-conditionneur basé sur les factorisations incomplètes de Cholesky des blocs diagonaux de l'opérateur défini positif

$$\widetilde{P}^* \widetilde{P} = \left[\begin{array}{c|c} (\mathcal{X}_u{}^* + \mathcal{S}^*)(\mathcal{X}_u + \mathcal{S}) & 0 \\ \hline 0 & \mathcal{X}_d{}^* \mathcal{X}_d \end{array}\right],$$

où

$$\begin{aligned}(\mathcal{X}_u{}^* + \mathcal{S}^*)\,(\mathcal{X}_u + \mathcal{S})\,(v) = \\ \tfrac{1}{2}\,\text{us2vec}\,[(X^2 + (S^y)^2)\,\text{us2Mat}\,(v) + \,\text{us2Mat}\,(v)(X^2 + (S^y)^2)] \\ + \,\text{us2vec}\,[X\,\text{us2Mat}\,(v)S^y + S^y\,\text{us2Mat}\,(v)X]\,.\end{aligned} \tag{5.29}$$

Compte tenu de la condition de complémentarité perturbée, XS^y tend vers 0 quand μ vers 0. Par suite,

$$\begin{aligned}\|X\,\text{us2Mat}\,(v)S^y + S^y\,\text{us2Mat}\,(v)X\|^2 = \,\text{trace}\,S^y\,\text{us2Mat}\,(v)XX\,\text{us2Mat}\,(v)S^y + \\ \text{trace}\,X\,\text{us2Mat}\,(v)S^yS^y\,\text{us2Mat}\,(v)X + \\ 2\,\text{trace}\,S^y\,\text{us2Mat}\,(v)XS^y\,\text{us2Mat}\,(v)X\end{aligned}$$

tend vers *zéro* quand μ tend vers zéro.

Nous pouvons alors utiliser l'approximation

$$\begin{aligned}(\mathcal{X}_u{}^* + \mathcal{S}^*)\,(\mathcal{X}_u + \mathcal{S})\,(v) \cong \\ \tfrac{1}{2}\,\text{us2vec}\,[(X^2 + (S^y)^2)\,\text{us2Mat}\,(v) + \,\text{us2Mat}\,(v)(X^2 + (S^y)^2)]\,.\end{aligned} \tag{5.30}$$

Dans la section précédente (Section 5.4.5), nous avons montré que le bloc diagonal inférieur est lui-même diagonal, donc la factorisation exacte de Cholesky pour ce bloc peut être calculée de manière peu coûteuse. De plus, même si les termes hors-diagonaux ne convergent pas vers zéro, on peut raisonnablement espérer qu'une factorisation incomplète de Cholesky pour le bloc diagonal supérieur et une factorisation exacte pour le bloc inférieur nous donnent un bon préconditionneur pour notre problème. Ceci se vérifie empiriquement, comme nous le verrons avec les résultats numériques présentés en Section 5.5.

Nous utilisons la transformation entre les indices c et (k, l) :

$$c \leftrightarrow (k, l), \quad c = \frac{(l-1)(l-2)}{2} + k, \quad k \le c; 1 \le k < l \le n.$$

Les colonnes du bloc supérieur sont les suivantes (toutes les lignes et colonnes qui ne sont pas précisées ci-dessous sont nulles) :

$$\begin{aligned}(\mathcal{X}_u{}^* + \mathcal{S}^*)\,(\mathcal{X}_u + \mathcal{S})\,(e_c) &= \tfrac{1}{2\sqrt{2}}\,\text{us2vec}\left\{Z^2\left(e_ke_l^T + e_le_k^T\right) + \left(e_ke_l^T + e_le_k^T\right)Z^2\right\} \\ &= \tfrac{1}{2\sqrt{2}}\,\text{us2vec}\begin{pmatrix} \text{en ligne } k \quad (Z^2)_{l:} \\ \text{en ligne } l \quad (Z^2)_{k:} \\ \begin{pmatrix}\text{en col } k \\ (Z^2)_{:l}\end{pmatrix} \begin{pmatrix}\text{en col } l \\ (Z^2)_{:k}\end{pmatrix}\end{pmatrix}\end{aligned} \tag{5.31}$$

où $Z = X + S^y$.

Pour $i \neq j$, nous notons $E_{ij} := \frac{1}{\sqrt{2}}\left(e_ie_j^T + e_je_i^T\right)$ l'élément (i, j) de la base orthonormale pour l'espace des matrices symétriques (quand $i = j$, on a $E_{ii} = e_ie_i^T$). Le symbole δ_{ij} représente le *produit de Kronecker*. Par suite, l'élément situé

en ligne $r \leftrightarrow (i,j)$ et colonne $c \leftrightarrow (k,l)$ est

$$\begin{aligned}
&\langle \text{ us2vec}\,(E_{ij}), (\mathcal{X}_u{}^* + \mathcal{S}^*)\,(\mathcal{X}_u + \mathcal{S})\,(\text{ us2vec}\,(E_{kl}))\rangle = \\
&\quad = \tfrac{1}{2}\,\text{us2vec}\,(E_{ij})^T\,\text{us2vec}\,\{(X^2 + (S^y)^2)E_{kl} + E_{kl}(X^2 + (S^y)^2)\} \\
&\qquad + \text{us2vec}\,(E_{ij})^T\,\text{us2vec}\,\{X E_{kl} S^y + S^y E_{kl} X\} \\
&\quad = \tfrac{1}{2}\,\text{trace}\,(E_{ij})\,\{Z^2 E_{kl} + E_{kl} Z^2\} \\
&\quad = \text{ trace}\, E_{ij} E_{kl} Z^2 \\
&\quad = \tfrac{1}{2}\,\text{trace}\,(e_i e_j^T + e_j e_i^T)(e_k e_l^T + e_l e_k^T) Z^2 \\
&\quad = \tfrac{1}{2}\,\text{trace}\,\left(e_i e_j^T e_k e_l^T + e_i e_j^T e_l e_k^T + e_j e_i^T e_k e_l^T + e_j e_i^T e_l e_k^T\right) Z^2 \\
&\quad = \tfrac{1}{2}\,\{\delta_{jk}(Z^2)_{li} + \delta_{jl}(Z^2)_{ki} + \delta_{ik}(Z^2)_{lj} + \delta_{il}(Z^2)_{kj}\}\,.
\end{aligned} \tag{5.32}$$

En pratique, l'approximation (5.30) correspond tout simplement à la suivante

$$(\mathcal{X}_u{}^* + \mathcal{S}^*)\,(\mathcal{X}_u + \mathcal{S}) \approx \mathcal{X}_u{}^* \mathcal{X}_u + \mathcal{S}^* \mathcal{S}.$$

La représentation matricielle en est obtenue à partir de celles de $\mathcal{X}_u$ et $\mathcal{S}$. Pour évaluer la matrice de $\mathcal{X}_u$, il suffit de remarquer que la colonne $c \cong (i,j)$, $(c = 1, \ldots, \left(\begin{smallmatrix} n \\ 2 \end{smallmatrix} \right))$, est obtenue à partir de la vectorisation de la matrice image de e_c, laquelle matrice a toutes ses composantes nulles, sauf les $i$ème et $j$ème colonnes qui sont respectivement les $j$ème et $i$ème de X (noter la permutation !). Cette matrice est donc naturellement creuse puisque chacune de ses colonnes, de taille n^2, a au maximum $2n$ composantes non nulles. De plus, sa construction est simple : elle consiste en fait à faire des permutations judicieuses des colonnes de X. En pratique, pour n fixé, on peut totalement déterminer les positions de ses composantes non nulles ainsi que leurs valeurs (extraites en des positions précises de X).

Pour obtenir la matrice de $\mathcal{S}$, on pourrait procéder comme ci-dessus, en raisonnant cette fois-ci sur les lignes de S^y. Toutefois, on peut aussi récupérer cette matrice directement à partir de celle de $\mathcal{X}_u$ en remarquant que, puisque
$X := A + \text{ us2Mat}\,(s) + I, \quad S^y := \text{ us2Mat}\,(s) + \text{ Diag}\,(y)$, on a

$$\mathcal{S}(\cdot) = (\mathcal{X}_u(\cdot))^T - ((A + I)\,\text{us2Mat}\,(\cdot))^T + \text{ us2Mat}\,(\cdot)\,\text{Diag}\,(y).$$

La matrice premier terme $(\mathcal{X}_u(\cdot))^T$ peut être obtenue de manière très simple à partir de celle de $\mathcal{X}_u$, en utilisant l'opérateur de transposition des matrices. Le second terme a une représentation matricielle qui s'obtient exactement comme celle de $\mathcal{X}_u$ en faisant jouer le rôle de X à $A + I$. De plus, ceci est fait une et une seule fois puisque ce terme est constant. La représentation matricielle du dernier terme est aussi facile à obtenir, puisqu'elle met en jeu des produits de matrices très creuses (deux composantes non nulles) avec une matrice diagonale. De même, pour n fixé, on peut totalement déterminer les positions de ses composantes non nulles ainsi que leurs valeurs (extraites en des positions précises de y).

5.5 Tests numériques

Dans cette section, nous présentons les différents résultats que nous avons obtenus à la suite des tests que nous avons menés avec les algorithmes que nous

avons présentés depuis le début de ce chapitre. Notons, d'une part, que dans toute la suite, nous ne considérons que des matrices A dont toutes les composantes sont inférieures à 1 en valeurs absolues. D'autre part, nous parlerons aussi de densité de matrice : il s'agit de la proportion de composantes non nulles d'une matrice (rapport entre le nombre de composantes non nulle et le nombre total de composantes).

Sauf indication contraire, nous avons fixé la précision pour tous les tests ci-après à $\varepsilon_1 = 10^{-10}$.

5.5.1 Problèmes de petite taille

Nous commençons par une présentation des résultats obtenus en appliquant la formulation mixte d'optimisation sur les cônes du second ordre et SDP (5.9) et notre algorithme de points intérieurs spécialisé à la résolution de problèmes de petites tailles ayant des propriétés particulières (problèmes provenant de la pratique). Ces tests ont été effectués en utilisant le code d'optimisation conique de J. STURM [105]. Ils ont été programmés en utilisant *MATLAB 6.5* sur un PC Pentium IV ayant 255 MO de mémoire vive.

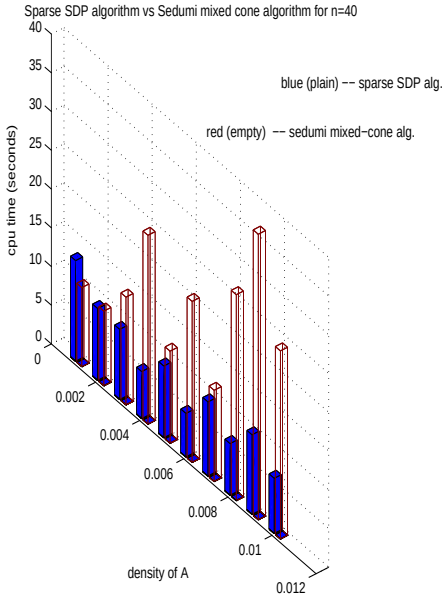
Premièrement, nous avons appliqué ces algorithmes à des problèmes denses et difficiles, de petite taille n allant de 20 à 60. La construction de ces problèmes est décrite dans [75] : il s'agit de problèmes pour lesquels la matrice A à approcher est une matrice de corrélation (obtenue à partir de la librairie disponible sous Matlab et écrite par HIGHAM) qui est perturbée par ajout de bruits (représentés par des matrices engendrées aléatoirement). Les résultats sont présentés dans le Tableau 5.2. Signalons que ces problèmes sont très dégénérés : très souvent, il n'y a pas *complémentarité stricte*, ce qui rend les algorithmes de points intérieurs inefficaces.

Taille de A n	Temps CPU pour notre algorithme avec $\varepsilon_1 = 10^{-8}$	Temps CPU pour notre algorithme avec $\varepsilon_1 = 10^{-12}$	SeDuMi
20	31.4	46.3	7.7
30	182.4	260.9	48.1
40	758.6	1041.4	269.0
50	2220.5	3197.6	1042.9
60	5139.7	7279.6	3205.9

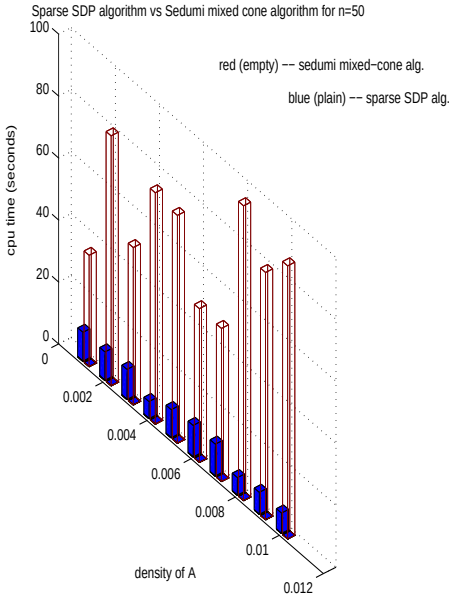
TAB. 5.2 – Résultats numériques pour A difficile et de grande taille

Il ressort de ce tableau que notre algorithme est moins efficace que SeDuMi lorsque le problème n'est pas creux. Nous attirons cependant l'attention sur le fait que notre algorithme permet tout de même d'atteindre un très grande précision dans les résultats sans aucun problème numérique, ce qui contraste avec les algorithmes de points intérieurs classiques pour lesquels l'absence de complémentarité stricte est souvent un inconvénient majeur.

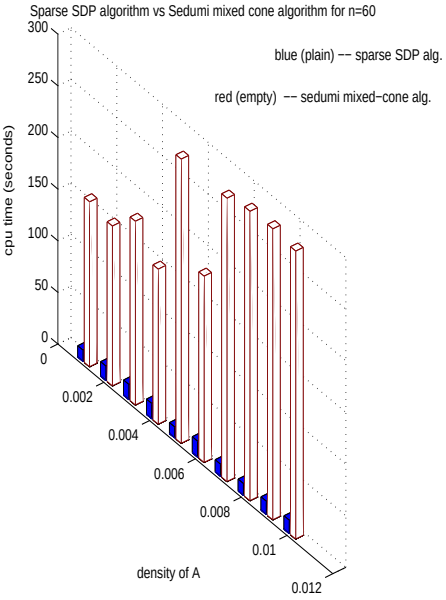
Nous avons comparé les algorithmes sur des matrices creuses engendrées aléatoirement (matrices A de dimension allant jusqu'à $n = 70$). La précision que nous avons requise pour ces tests est de 10^{-8} pour les deux algorithmes. Les résultats sont illustrés par les Figures 5.2 et 5.3.



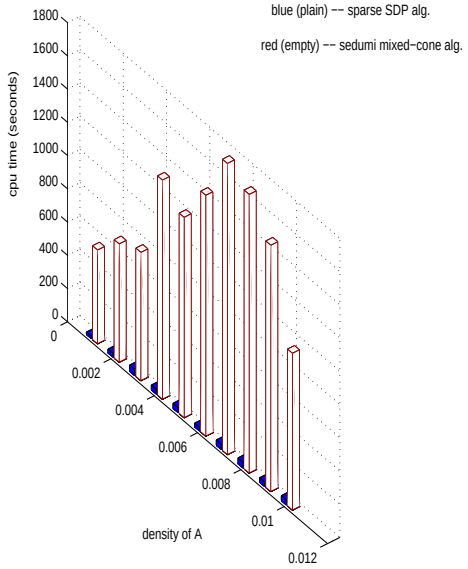
$n = 40$



$n = 50$



$n = 60$



$n = 70$

FIG. 5.2 – Comparaison SeDuMI avec nos points intérieurs

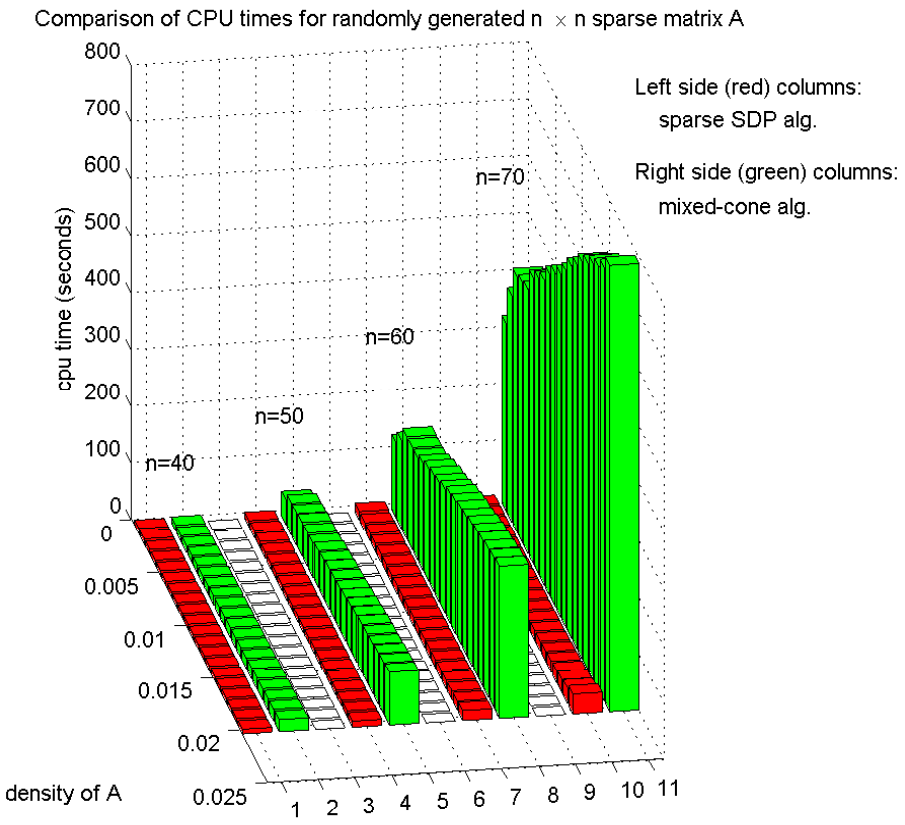


FIG. 5.3 – Temps CPU Comparaison SeDuMI avec nos points intérieurs (temps moyen après 10 tests pour chaque densité)

Comme c'est le cas pour des méthodes de points intérieurs, le nombre d'itération nécessaires à la convergence pour SeDuMi reste essentiellement constant (entre 12 et 15 itérations) indépendamment de la dimension du problème. Le temps de calcul par itération et l'espace mémoire nécessaire deviennent cependant rapidement prohibitivement élevés pour SeDuMi, alors que notre algorithme est capable d'exploiter la caractère creux et le coût par itération en est plus petit. En conclusion, notre approche permet de résoudre des problèmes plus grand en des temps de calcul beaucoup plus courts.

5.5.2 Problèmes creux de grande taille

Tout d'abord, nous illustrons notre algorithme de points intérieurs-extérieurs au travers des différents résultats obtenus au cours des itérations. Ils sont résumés dans le tableau 5.3. Ils correspondent à l'approximation d'une matrice creuse A de taille $n = 300$ et de densité 0.0005.

On peut observer sur le tableau les différentes propriétés de notre algorithme de points intérieurs-extérieurs. En particulier, puisque les systèmes linéaires résolus

Numéro d'itération	Saut de dualité en -$\log_{10}$	Valeur de l'objectif $\times 10^2$	Pas α	Paramètre σ	Itérations de gradients conjugués	Temps de calcul s
1	0.466	1.5442	0.7695	1	21	3.0840
2	0.735	1.5119	0.95	0.76915	16	3.2040
3	1.35	1.5029	0.95	0.715	18	2.6640
	crossover					
4	3.17	1.5006	1	0	31	4.2060
5	3.99	1.5001	1	0	48	8.6030
6	4.65	1.5000	1	0	51	9.0330
7	5.30	1.5000	1	0	55	10.395
8	5.96	1.5000	1	0	41	7.3900
9	6.67	1.5000	1	0	50	9.3340
10	7.32	1.5000	1	0	51	8.9330
11	7.92	1.5000	1	0	50	9.2230
12	8.52	1.5000	1	0	51	9.3030
13	9.13	1.5000	1	0	30	5.5680
14	9.73	1.5000	1	0	52	9.2430
15	10.3	1.5000	1	0	54	9.5440

TAB. 5.3 – Illustration de notre approche SDP pour une matrice de taille $n = 300$ et de densité 0.0005.

sont de taille $\begin{pmatrix} n+1 \\ 2 \end{pmatrix} = 45150$, le nombre d'itérations de gradients conjugués est au maximum de l'ordre de 45150. Ce nombre d'itérations ici reste inférieur à 55, ce qui montre l'efficacité et la robustesse de notre pré-conditionnement. De plus, on peut remarquer que nous atteignons la valeur optimale très rapidement en 6 itérations, soit environ en 30 secondes. De plus, à cette étape, nous possédons la solution optimale avec une précision de 10^{-4}. Cette solution peut être obtenue avec une plus grande précision (10^{-10}) sans aucun problème numérique et sans que le temps de calcul par itération n'explose, ce qui corrobore les propriétés de convergence quadratique asymptotique de notre algorithme.

Nous avons résolu trois ensembles de 26 à 30 problèmes avec comme dimensions $n = 200, 300, 350$, et des densités de la matrice A allant de $.0005$ à $.003$, par pas de $.001$. Ces matrices sont engendrées aléatoirement sous Matlab en utilisant la fonction *sprandsym*. Dans tous les cas, nous avons trouvé l'optimum avec une grande précision (à $\varepsilon = 10^{-10}$ près). Les résultats sont présentés sur les figures 5.4 et 5.5. Nous pouvons voir qu'il y apparaît une corrélation entre le temps de calcul et le nombre de composantes non nulles de l'optimum X.

5.5.3 Robustesse

Nous avons remarqué précédemment que notre algorithme était particulièrement efficace lorsque l'on résolvait des problèmes creux, ce qui correspond à avoir la matrice A creuse. Mais, lorsque A est dense, les opérateurs $\mathcal{X}_u, \mathcal{S}, \mathcal{X}_d$ ne sont pas creux. La résolution devient alors plus difficile, ne serait-ce que parce que l'on se trouve face à des problèmes d'espace mémoire.

Nous avons dans un premier temps étudié la robustesse de notre algorithme.

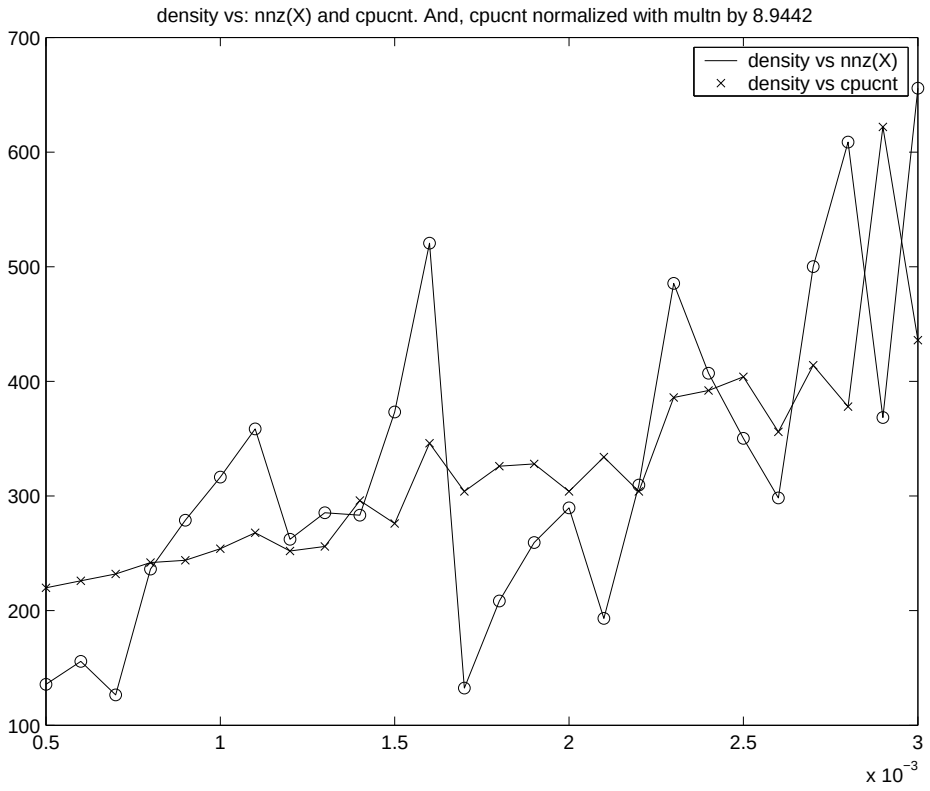


FIG. 5.4 – 30 problèmes ; dimension $n = 200$

Ceci a été fait empiriquement de la manière suivante : nous faisons tourner l'algorithme pour une certaine matrice A, engendrée aléatoirement. Puis, au cours des itérations, nous introduisons des perturbations aléatoires dans la matrice A. Ce qui, bien sûr, perturbe tout le problème. Nous avons pu remarquer, sur tous les exemples que nous avons testés, que l'algorithme restait relativement insensible à ces perturbations, notamment en termes de vitesse convergence. Il s'avère donc que l'algorithme est **robuste**.

Nous avons exploité cette robustesse de manière à résoudre des problèmes de grande taille pour lesquels la matrice A n'est pas forcément creuse, de manière à éviter les problèmes d'espace mémoire. La démarche est la suivante : on initialise à zéro toutes les composantes de A qui sont de valeur absolue inférieure à une certaine tolérance, par exemple, toutes les composantes telles que $\text{abs}\,(A_{ij}) < tol_0$, avec $tol_0 = 0.9$ initialement. Le problème est résolu avec cette tolérance jusqu'à ce que nous obtenions un saut de dualité inférieur à 10^{-3}. Nous faisons alors décroître la tolérance tol_0 (par paliers de 0.1) à chaque nouvelle itération jusqu'à obtenir $tol_0 = 0$. A partir de là, les itérations suivantes, jusqu'à la convergence, sont faites avec toutes les composantes de A.

Nous présentons dans le tableau 5.4 et dans la figure 5.7 une illustration de la manière dont nous utilisons la robustesse de notre algorithme de points intérieurs-extérieurs. Ils représentent l'évolution au cours des itérations du nombre d'éléments non nuls, du saut de dualité représenté par μ, de la valeur courante de la fonction

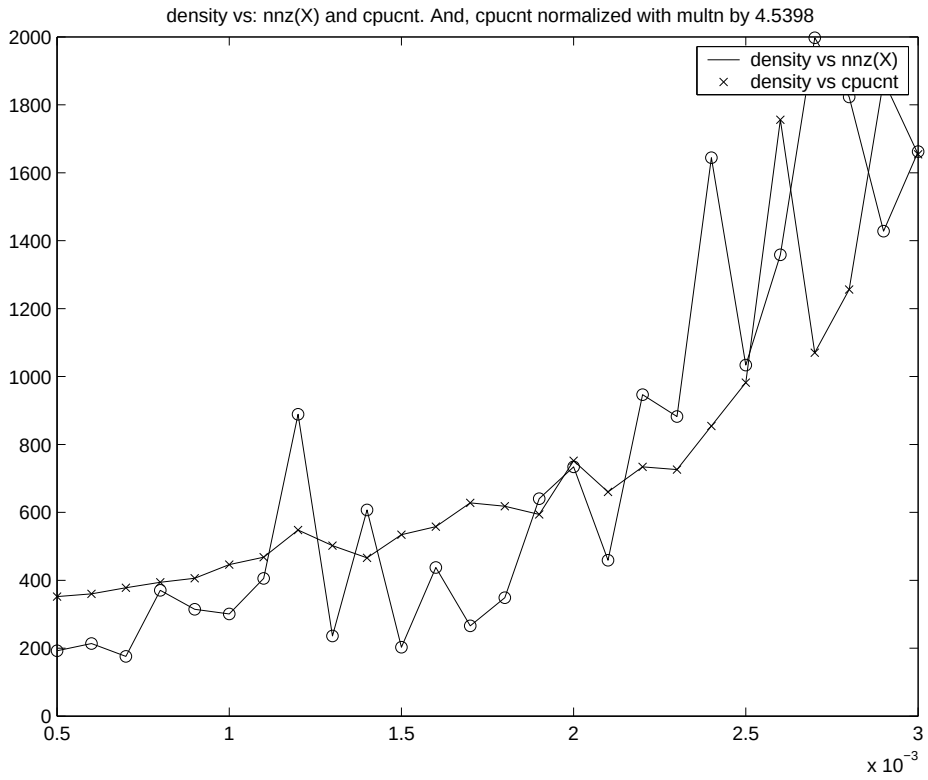


FIG. 5.5 – 30 problèmes ; dimension $n = 300$

objectif et du temps de calcul nécessaire à chaque itération pour un test effectué avec une matrice A de taille $n = 100$ et de densité 0.01. Nous faisons remarquer que, cette fois aussi, dans la quatrième colonne les résultats que nous donnons correspondent en réalité à l'opposé du logarithme décimal du saut de dualité.

Comme nous l'avons annoncé, on peut observer que pendant les trois premières itérations, on n'utilise que les 4 composantes de A qui sont plus grandes que le seuil $tol_0 = 0.9$, ce qui fait de A une matrice très creuse. Puis, puisqu'à l'itération 3, le saut de dualité courant est d'approximativement 10^3 ($10^{2.91}$, en fait). A partir de l'itération 4, on abaisse le seuil tol_0 de 0.1 à chaque itération jusqu'à ce que ce seuil soit égal à zéro. Cela permet de récupérer exactement la matrice A de départ à partir de l'itération 12. On observera aussi qu'à partir de l'itération 11, on observe une convergence quadratique car l'opposé du logarithme décimal du saut de dualité double à chaque itération.

Nous avons observé que l'algorithme est extrêmement robuste et ces perturbations ne ralentissent pas de manière appréciable la convergence. Cela montre aussi qu'avec cette approche, il est possible d'effectuer des démarrages à chaud sans détériorer les bonnes propriétés de convergence avec cette approche.

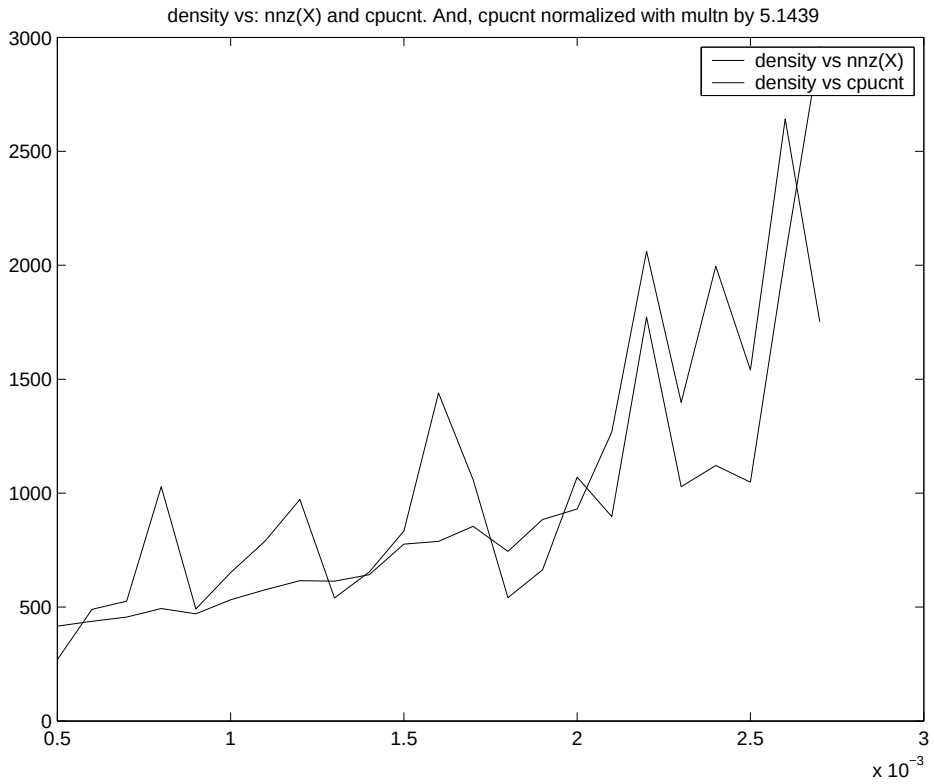


FIG. 5.6 – 28 problèmes ; dimension $n = 350$

5.6 Projections vs Points intérieurs : premières comparaisons

Pour terminer ce travail, nous avons comparé notre algorithme de points "intérieurs-extérieurs" avec l'algorithme par projections alternées de HIGHAM [75].

Du point de vue du travail de programmation à effectuer, l'algorithme de projections alternées s'avère d'une utilisation plus simple, surtout pour un novice en termes de programmation et d'Analyse numérique. Il ne requiert que le calcul préalable de projections sur des convexes simples qui peuvent s'obtenir, ainsi que nous l'avons vu, explicitement par calculs. D'un autre côté, l'algorithme de points intérieurs requiert une certaine connaissance de l'Analyse numérique, combinée avec une utilisation judicieuse de résultats d'Algèbre linéaire numérique.

Du point de vue performance par contre, l'algorithme de points intérieurs présente des qualités de robustesse, qui sont très intéressantes. Ceci s'ajoute à des qualités de convergence rapide (quadratique) et de grande précision dans les résultats. Au contraire, l'algorithme de projections alternées a une convergence sous-linéaire, puisqu'on n'effectue pas uniquement des projections sur des sous-espaces. De ce fait, une grande précision des résultats est difficile à obtenir.

En théorie, la comparaison effective entre ces deux approches est donc difficile. Seule l'utilisation future que l'on veut faire des résultats numériques donnés par les algorithmes peut permettre de se prononcer raisonnablement en faveur de l'une ou l'autre approche. De plus, en pratique, se pose aussi la question du langage

tol_0	Numéro d'itération	Nombre éléments non nuls de A	Saut de dualité en -$\log_{10}$	Valeur de l'objectif $\times 10$	Temps de calcul $\times 10^{-1} s$
0.9	1	4	1.58	5.0862	1.7
	2	4	2.23	5.0147	1.2
	3	4	2.91	5.0041	1.1
0.8	4	4	3.53	5.0009	1.3
0.7	5	10	4.13	5.0002	1.2
0.6	6	10	4.76	5.0000	1.1
0.5	7	20	5.41	5.0000	1.5
0.4	8	32	6.07	5.0000	4.2
0.3	9	42	4.41	5.0000	5.2
0.2	10	58	4.59	5.0000	3.4
0.1	11	78	7.35	5.0000	6.5
0	12	100	8.66	5.0000	16.5
	13	100	15.5	5.0000	38

TAB. 5.4 – Utilisation de la robustesse

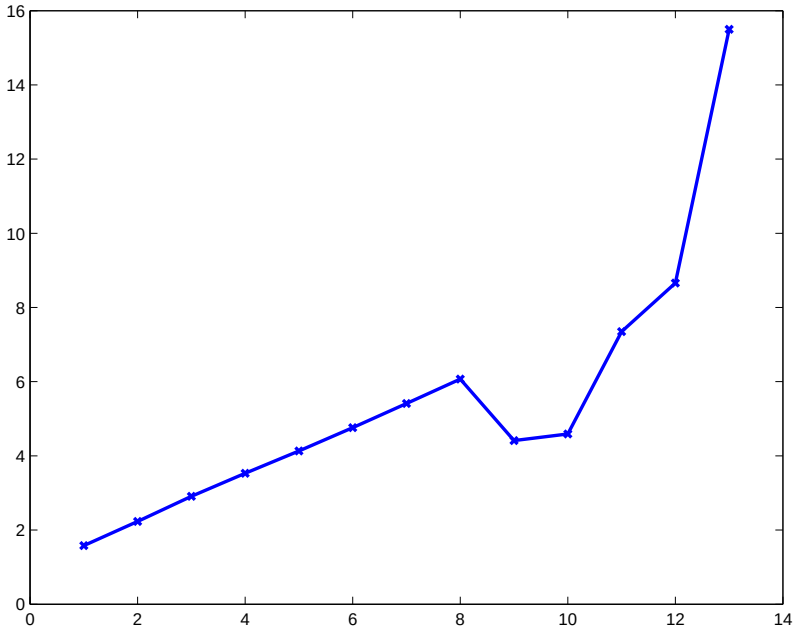


FIG. 5.7 – Utilisation de la robustesse : courbe de convergence

de programmation que l'on utilise.

Nous avons fait la comparaison entre ces deux approches en résolvant des problèmes d'approximation par matrices de corrélation, pour lesquels nous faisons varier la taille de la matrice A (entre 60 et 110) et sa densité (entre 0.001 et 0.012). Pour chaque couple (taille, densité), un ensemble de 10 problèmes est résolu et nous avons gardé les temps de calculs moyens. Ces résultats sont présentés dans les figures ci-après (Figure 5.8). Les barres peines (noires) représentent les résultats pour notre algorithme de points intérieurs, les vides (blanches) ceux de l'algorithme de projections alternées.

On peut observer deux tendances dans les résultats que nous avons obtenus : pour les matrices de taille allant jusqu'à 80, l'approche SDP est meilleure que l'approche par projections. C'est ce à quoi on s'attend naturellement, compte tenu de la différence de convergence asymptotique. Pour les tailles supérieures, l'algorithme par projections alternées prend le dessus. Ceci s'explique par la différence de langage de programmation que nous avons évoquée. En effet, l'algorithme de points intérieurs que nous avons écrit l'est entièrement en langage Matlab. Par contre, l'approche par projections alternées utilise des routines du noyau LAPACK de Matlab, écrit en C/C++ ou fortran, qui sont plus spécialisées, notamment pour le calcul des valeurs propres. En effet, dans une itération de projections alternées, le travail principal consiste en une décomposition en valeurs propres qui est effectuée au travers de la fonction *eig* de Matlab, qui est en fait une routine LAPACK, donc très rapide et robuste. Tandis que, dans l'algorithme de points intérieurs, le travail principal est une résolution d'un système linéaire au sens des moindres carrés, grâce à une fonction *lsqr* écrite totalement en langage Matlab. La comparaison entre ces deux fonctions *eig* et *lsqr* est nettement en faveur de la première. Le phénomène que l'on observe à partir de la taille 90 vient du fait qu'à partir de ce moment, la différence de vitesse de convergence entre les deux algorithmes est complètement outrepassée par la différence de temps de calculs entre *eig* et *lsqr*, rendant l'approche par projections alternées plus rapide.

Toutefois, on peut remarquer que lorsque la matrice A est très creuse (densité petite, voir les débuts de chaque figure), d'une manière générale l'algorithme par points intérieurs est meilleur. Ceci s'explique par le fait que cet algorithme, notamment en termes de pré-conditionnement des systèmes linéaires pour *lsqr*, utilise de manière quasi-optimale, le caractère creux du problème (donc de A).

A priori, on se serait attendu, du fait de la différence de convergence (quadratique contre sous-linéaire) à ce que l'approche par points intérieurs-extérieurs soit plus rapide que l'approche par projections alternées. Les tests que nous avons faits ne nous permettent cependant pas de conclure de manière définitive. Toutefois, il existe des explications, de nature essentiellement informatique, aux résultats décevants que nous venons de présenter. En conséquence, en ce qui concerne cette dernière partie de la thèse (Section 5.6), nous ne pouvons qu'ouvrir la voie vers des travaux numériques supplémentaires qui sont requis afin de trancher la question.

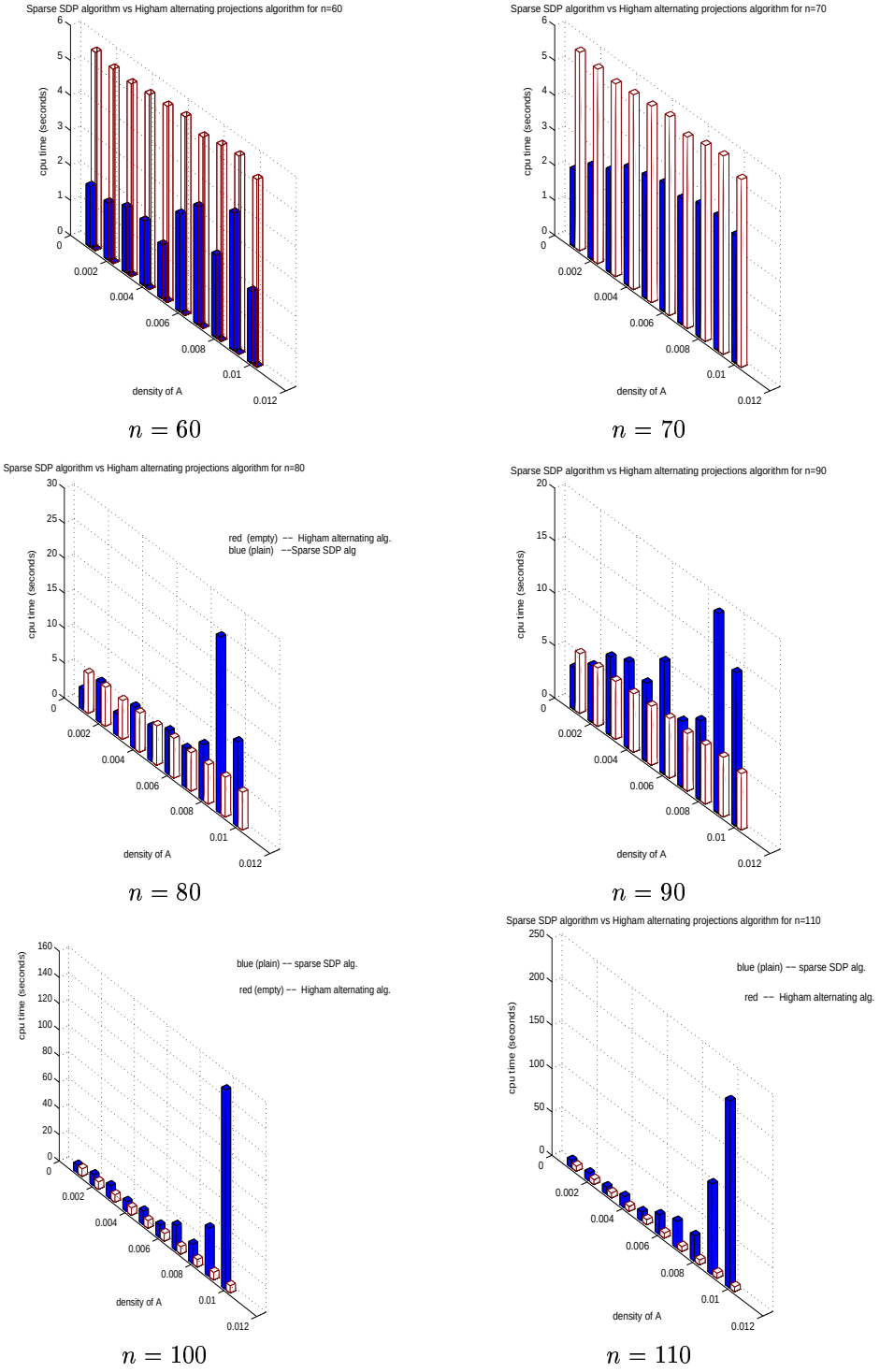


FIG. 5.8 – Comparaison de projections alternées avec points intérieurs

Conclusion

Nous nous sommes intéressé dans cette thèse à la résolution effective de problèmes d'approximation linéaires coniques. Notre objectif était de proposer, pour le résoudre effectivement, des solutions algorithmiques qui soient assez rapides pour fournir une solution à ces problèmes dans des délais raisonnables (parfois quelques secondes) et qui soient suffisamment robustes pour permettre des appels répétés à ces algorithmes.

Nous avons pour ce faire étudié différentes approches de résolutions. Nous avons retenu deux approches de natures différentes que nous avons testées sur deux problèmes d'approximation matricielle : l'approximation par matrices bistochastiques et par matrices de corrélations. Nous avons comparé ces approches essentiellement sur le dernier problème. La première approche est une approche de type primale. Elle a consisté à l'utilisation de l'algorithme modifiée de projections alternées proposé par BOYLE et DYKSTRA au cours des années quatre vingt. La seconde, primale-duale, s'appuie sur une combinaison judicieuse des très récents outils d'optimisation que sont l'optimisation sous contraintes de semidéfinie positivité et les méthodes de points intérieurs avec des techniques de pointe d'algèbre linéaire numérique. Nous en avons déduit un algorithme qui exploite au maximum la structure propre du problème, notamment sa structure creuse. Il ressort de nos tests que chacune des approches peut servir valablement à la résolution des problèmes d'approximations évoqués en des temps raisonnables. Toutefois, ces algorithmes sont de natures différentes : le premier est très simple à mettre en œuvre, au contraire du second qui requiert des connaissances plus poussées en Analyse numérique. Ils ont des propriétés différentes : le second permet d'obtenir des résultats très précis et converge quadratiquement tandis que le premier a une convergence sous-linéaire, et ne peut donner des résultats d'une grande précision. De fait, le choix entre ces deux approches apparaît comme dépendant du cadre dans lequel on cherche à résoudre le problème d'approximation.

De nombreuses perspectives s'ouvrent à la suite de ce travail concernant les différents algorithmes ci-dessus évoqués. L'algorithme par projections alternées que nous avons utilisé n'est qu'un choix parmi la large palette d'algorithmes de type projection que l'on peut appliquer à la résolution de problèmes d'approximation matriciels. Ils peuvent d'ailleurs s'appliquer à des problèmes plus généraux que ceux, linéaires coniques, considérés dans cette thèse. Il devrait être très intéressant d'orienter nos recherches dans cette voie. En ce qui concerne l'algorithme de points intérieurs, il a besoin d'être amélioré, par programmation dans un autre langage et/ou parallélisation, pour remédier aux inconvénients qui ont été décelés pour les

problèmes de grande taille et lors de la comparaison avec les projections alternées. De plus, la démarche que nous avons suivie, par Gauss-Newton et "crossover" n’en est qu’à ses débuts. Des recherches supplémentaires devraient être conduites dans cette direction.

Bibliographie

[1] A. Alfakih, A. Khandani, and H. Wolkowicz, *Solving Euclidean distance matrix completion problems via semidefinite programming*, Computational Optimization and Applications **12** (1999), no. 1-3, 13–30.

[2] A. Alfakih and H. Wolkowicz, *Matrix completion problems*, Handbook Of Semidefinite Programming : Theory, Algorithms, and Applications (R. Saigal, L. Vandenberghe, and H. Wolkowicz, eds.), Kluwer Academic Publishers, Boston, MA, 2000, pp. 533–545.

[3] A. Alfakih and H. Wolkowicz, *A new semidefinite programming model for large Sparse Euclidean distance Matrix completion problems*, Tech. report, University of Waterloo, Department of Combinatorics and Optimization, 2001, Research Report CORR # 2000-37.

[4] ———, *Two theorems on Euclidean distance matrices and Gale transform*, Linear Algebra and its Applications **340** (2002), 149–154.

[5] F. Alizadeh, *Interior point methods in semidefinite programming with applications to combinatorial optimization*, SIAM Journal on Optimization **5** (1995), no. 1, 13–51.

[6] F. Alizadeh, J-P. Haeberly, and M.L. Overton, *Primal-dual interior-point methods for semidefinite programming : convergence rates, stability and numerical results*, SIAM Journal on Optimization **8** (1998), no. 3, 746–768 (electronic).

[7] I. Amemiya and T. Ando, *Convergence of random products of contractions in Hilbert space*, Acta Universitatis Szegediensis. Acta Scianitarum Mathematicarum (Szeged) **26** (1965), 239–244.

[8] M.F. Anjos, *New convex relaxations for the maximum cut and vlsi layout problems*, Ph.D. thesis, University of Waterloo, Canada, May 2001.

[9] M.F. Anjos, N.J. Higham, P.L. Takouda, and H. Wolkowicz, *A semidefinite programming approach for the nearest correlation matrix problem*, Tech. report, Dept. of Combinatorics & Optimization, University of Waterloo, Canada, 2003, In progress.

[10] O. Axelsson, *Iterative solution methods*, Cambridge University Press, Cambridge, 1994.

[11] J.B. Baillon and R.E. Bruck, *On the random product of orthogonal projections in Hilbert Space*, Nonlinear analysis and convex analysis, World Sciences Publishing, River Edge, NJ, 1999, pp. 2126–133.

[12] M. Baïou, M. Balinski, and R. Laraki, *Dossier spécial Elections*, Pour la Science **294** (2002).

[13] C.R. Barrett, P.K. Pattanaik, and M. Salles, *Rationality and aggregation of preferences in an ordinally fuzzy framework*, Fuzzy Sets and Systems. International Journal of Soft Computing and Intelligence **49** (1992), no. 1, 9–13.

[14] H.H. Bauschke, *The approximation of fixed points of composition of nonexpansive mapping in Hilbert spaces*, Journal of Mathematical Analysis and Applications **202** (1996), no. 1, 150–159.

[15] _____, *Projections Algorithms and Monotone Operators*, Ph.D. thesis, Simon Fraser University, August 1996.

[16] _____, *Projections algorithms : results and open problems*, Inherently Parallel Algorithms in Feasibility and Optimization and their Applications (Haifa 2000) (D. Butnariu, Y. Censor, and S. Reich, eds.), Stud. Comput. Math., vol. 8, Elsevier science, 2001, pp. 409–422.

[17] H.H. Bauschke and J.M. Borwein, *On the convergence of von Neumann's alternating projection algorithm for two sets*, Set-Valued Analysis **1** (1993), no. 2, 185–212.

[18] _____, *Dykstra's alternating projection algorithm for two sets*, Journal of Approximation Theory **79** (1994), no. 3, 418–443.

[19] _____, *On projection algorithms for solving convex feasibility problems*, SIAM Review **38** (1996), no. 3, 367–426.

[20] _____, *Legendre functions and the method of random Bregmann projections*, Journal of Convex Analyis **4** (1997), no. 1, 27–67.

[21] H.H. Bauschke, J.M. Borwein, and A.S. Lewis, *The method of cyclic projections for closed convex sets in Hilbert space*, Recent developments in Optimization and nonlinear analysis (Y. Censor and S. editors Reich, eds.), Contemporary Mathematics, vol. 204, Amer. Math. Soc., Providence, RI, 1997, Proceedings on the special session on Optimization and Nonlinear Analysis, Jerusalem, May 1995., pp. 1–38.

[22] H.H. Bauschke, S.G. Kruk, and H. Wolkowicz, *Evaluating performance of algorithms for conically and linearly best approximation problems.*, Work in progress. Private communication of H.H. Bauschke at the University of Guelph, Canada., October 2002.

[23] H.H. Bauschke and A.S. Lewis, *Dykstra's algorithm with Bregman projections : a convergence proof*, Optimization **48** (2000), no. 4, 409–427.

[24] J-M. Blin, *A linear assignment formulation of the multiattribute decision problem*, RAIRO Recherche opérationnelle/Operations Research, Série Verte **10** (1976), no. 2, 21–32.

[25] A. Borobia, Z. Nutov, and M. Penn, *Doubly stochastic matrices and dicycle covers and packings in Eulerian digraphs*, Linear Algebra and its Applications **246** (1996), 361–371.

[26] J.P. Boyle and R.L. Dykstra, *A method for finding projections onto the intersection of convex sets in Hilbert spaces*, Advances in Order Restricted Statistical Inference (R. L. Dykstra, T Robertson, and F. T. Wright, eds.), Lecture Notes in Statistics, vol. 37, Springer-Verlag, 1985, pp. 28–47.

[27] L.M. Bregman, *The method of successsive projection for finding a common point of convex sets*, Soviet Mathematics Doklady **6** (1965), 605–611.

[28] L.M. Bregman, Y. Censor, S. Reich, and Y. Zepkowitz-Malachi, *Finding the projection of a point onto the intersection of convex sets via projections onto halfspaces*, Tech. report, University of Haifa, 2003, Accepté pour publication dans le *Journal of Approximation Theory*.

[29] H. Brezis, *Analyse fonctionnelle. Théories et Applications*, Masson, 1983.

[30] R.A. Brualdi, *Notes on the Birkhoff algorithm for doubly stochastic matrices*, Canad. Math. Bull. **25** (1982), no. 2, 191–199.

[31] _____, *Some applications of doubly stochastic Matrices*, Linear algebra and its applications **107** (1988), 77–100.

[32] R.A. Brualdi and P.M. Gibson, *Convex polyhedra of doubly stochastic Matrices. I : Applications of the permanent function*, Journal of combinatorial theory **22** (1977), 194–230.

[33] R.A. Brualdi and B. Liu, *The polytope of even doubly stochastic Matrices*, Journal of combinatorial theory (1991), 243–253.

[34] W. S.. Burdic, *Underwater acoustic system analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1991, 2nd edition.

[35] J. P. Burg, D. G. Luenberger, and D. L. Wenger, *Estimation of structured covariance matrices*, Proceedings of the IEEE, vol. 70, 1982, pp. 963–974.

[36] J. A. Cadzow, *Signal enhancement - a composite property mapping algorithms,*, IEEE Transactions on Acoustics, Speech, and Signal Processing **36** (1988), 49–62.

[37] I. Charon and O. Hudry, *Lamarckian genetic algorithms applied to the aggregation of preferences*, Annals of Operations Research **80** (1998), 281–297.

[38] V. Chvàtal, *Linear programming*, W.H. Freeman and Company, 1983.

[39] P.L. Combettes, *The foundations of set theoretic estimation*, Proceedings of the IEEE, vol. 81, 1993, pp. 182–208.

[40] _____, *Signal recovery by best feasible approximation*, IEEE Transactions on Image Processing **2** (1993), no. 2, 269–271.

[41] _____, *Inconsistent Signal Feasibility Problems : Least-Squares Solutions in a Product Space*, IEEE Transactions on Signal Processing **42** (1994), no. 11, 2955–2966.

[42] _____, *Convex set theoretic image recovery by extrapolated iterations of parallel subgradient projections*, IEEE Transactions on Image Processing **6** (1997), no. 4, 493–506.

[43] _____, *Hilbertian convex feasibility problem : Convergence of projection methods*, Applied Mathematics and Optimization **35** (1997), 311–330.

[44] _____, *Strong convergence of block-iterative outer approximation methods for convex optimization*, SIAM Journal on Control and Optimization **38** (2000), no. 2, 538–565.

[45] _____, *Quasi-Fejérian analysis of some optimization algorithms*, Inherently Parallel Algorithms in Feasibility and Optimization and their Applications (Haifa 2000) (D. Butnariu, Censor Y., and S. Reich, eds.), Studies in Computational Mathematics, vol. 8, Elsevier science, 2001, pp. 115–152.

[46] P.L. Combettes and P. Bondon, *Hard-constrained Inconsistent Signal Feasibility Problems*, IEEE Transactions on Signal Processing **45** (1999), no. 9, 2460–2468.

[47] E. De Klerk, J.E. Hoogenboom, T Illes, A.J. Quist, C. Roos, T. Terlaky, and R. Van Geemert, *Optimization of a nuclear reactor core reload pattern using nonlinear optimization and search heuristics*, Delft University of Technology, Departement of Operations research, draft paper, September 1997.

[48] E. De Klerk, K. Roos, and T. Terlaky, *Self-dual embeddings*, Handbook of semidefinite programming, Internat. Ser. Oper. Res. Management Sci., vol. 27, Kluwer, Boston, MA, 2000, pp. 111–138.

[49] G. Demange and J-C. Rochet, *Méthodes mathématiques de la finance*, Frontières de la Théorie économique, Economica, Paris, 1997.

[50] J. E. Dennis, Jr. and H. Wolkowicz, *Sizing and least-change secant methods*, SIAM Journal on Numerical Analysis **30** (1993), no. 5, 1291–1314.

[51] J.E. Dennis and R.B. Schnabel, *Numerical methods for unconstrained optimization and nonlinear equations*, second ed., CLASSICS in Applied Mathematics, SIAM, 1996.

[52] R.L. Dykstra, *An algorithm for Restricted Least Squares Regression*, Journal of the American Statistical Association **78** (1983), no. 384, 837–842.

[53] G. P. Egorychev, *The solution of van der Waerden's problem for permanents*, Advances in Mathematics **42** (1981), no. 3, 299–305.

[54] R. Escalante, *Dykstra's algorithm for a constrained least-squares matrix problem*, Numerical Linear Algebra with Applications **3** (1996), no. 6, 459–471.

[55] D. I. Falikman, *Proof of the van der Waerden conjecture on the permanent of a doubly stochastic matrix*, Akademiya Nauk Soyuza SSR. Matematicheskie Zametki **29** (1981), no. 6, 931–938, 957.

[56] B. Fares, *Théorie de la commande robuste et techniques d'optimisation avancées*, Ph.D. thesis, Université Paul Sabatier, Toulouse, France, July 2001.

[57] B. Fares, P. Apkarian, and D. Noll, *An augmented Lagrangian method for a class of LMI-constrained problems in robust control theory*, International Journal of Control **74** (2001), no. 4, 348–360.

[58] B. Fares, D. Noll, and P. Apkarian, *Robust control via sequential semidefinite programming*, SIAM Journal on Control and Optimization **40** (2002), no. 6, 1791–1820 (electronic).

[59] M.C. Ferris, M.P. Mesnier, and J.J. Moré, *NEOS and Condor : Solving optimization problems over the Internet*, ACM Transactions on Mathematical Software **26** (2000), no. 1, 1–18.

[60] P. Forster, *Generalized rectification of cross spectral matrices for arrays of arbitrary geometry*, IEEE Transactions on Signal Processing **49** (2001), 972–978.

[61] C. Fortin and H. Wolkowicz, *A survey of the trust region subproblem within a semidefinite programming framework*, Tech. report, University of Waterloo, Department of Combinatorics and Optimization, 2000, Research Report CORR # 2002-22.

[62] A. E. Frazho, K. M. Grigoriadis, and R. E. Skelton, *Applications of alternating convex projections methods for computation of positive toeplitz matrices*, IEEE Transactions on Signal Processing **42** (1994), 1873–1875.

[63] N. Gaffke and R. Mathar, *A cyclic projection algorithm via duality*, Metrika **36** (1989), 29–54.

[64] W. Glunt, T.L. Hayden, S. Hong, and J. Wells, *An alternating projection algorithm for computing the nearest Euclidian distance matrix*, SIAM Journal on Matrix Analysis and Applications **11** (1990), no. 4, 589–600.

[65] W. Glunt, T.L. Hayden, and R. Reams, *The nearest 'doubly stochastic' matrix to a real matrix with the same first moment*, Numerical Linear Algebra with Applications **5** (1998), 475–482.

[66] A. Greenbaum, *Iterative methods for solving linear systems*, Frontiers in Applied Mathematics, vol. 17, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.

[67] B. Gyires, *Elementary proof for a van der Waerden's conjecture and related theorems*, Computers & Mathematics with Applications. An International Journal **31** (1996), no. 10, 7–21.

[68] _____, *Contribution to van der Waerden's conjecture*, Computers & Mathematics with Applications. An International Journal **42** (2001), no. 10-11, 1431–1437.

[69] M. Halicka, E. De Klerk, and C. Roos, *Limiting behavior of the central path in semidefinite optimization*, Tech. report, Optimization Online, 2002.

[70] S.P. Han, *A successive projection method*, Mathematical Programming **40** (1988), 1–14.

[71] C. Helmberg, F. Rendl, R.J. Vanderbei, and H. Wolkowicz, *An interior-point method for semidefinite programming*, SIAM Journal on Optimization **6** (1996), no. 2, 342–361.

[72] D. Henrion, Y. Labit, and D. Peaucelle, *SeDuMi interface 1.02 : A Tool for Solving LMI Problems with SeDuMi*, Proceedings of the CACSD Conference, September 2002.

[73] N.J. Higham, *Computing a nearest symmetric positive semidefinite matrix*, Linear Algebra and its Applications **103** (1988), 103–118.

[74] _____, *Matrix nearness problems and applications*, Applications of Matrix Theory (M. J. C. Gover and S. Barnett, eds.), Oxford University Press, 1989, pp. 1–27.

[75] _____, *Computing the nearest correlation matrix—a problem from finance*, IMA Journal of Numerical Analysis **22** (2002), no. 3, 329–343.

[76] J-B. Hiriart-Urruty, *Optimisation et analyse convexe*, Presses Universitaires de France, 1998.

[77] J-B. Hiriart-Urruty and C. Lemaréchal, *Convex analysis and minimization algorithms*, Grundlehren der mathematischen Wissenchaften 305 & 306. Springer-Verlag Berlin Heidelberg, 1993, New printing in 1996.

[78] R.B. Horn and C.R. Johnson, *Matrix Analysis*, Cambridge University Press, 1985, (reprinted in 1991, 1992).

[79] N. Karmarkar, *A new polynomial-time algorithm for linear programming*, Combinatorica **4** (1984), no. 4, 373–395.

[80] R.N. Khoury, *Closest matrices in the space of generalized doubly stochastic matrices*, Journal of Mathematical Analysis and Applications **222** (1998), 562–568.

[81] K.C. Kiwiel, *The efficiency of subgradient projection methods for convex optimization, part I : general level methods*, SIAM Journal on Control and Optimization **34** (1996), no. 2, 660–676.

[82] K.C. Kiwiel and B. Lopuch, *Surrogate projection methods for finding fixed points or firmly nonexpansive mappings*, SIAM Journal on Optimization **7** (1997), no. 4, 1084–1102.

[83] M. Kojima, S. Shindoh, and S. Hara, *Interior-point methods for the monotone semidefinite linear complementarity problem in symmetric matrices*, SIAM Journal on Optimization **7** (1997), no. 1, 86–125.

[84] S. Kruk, M. Muramatsu, F. Rendl, R.J. Vanderbei, and H. Wolkowicz, *The Gauss-Newton direction in semidefinite programming*, Optimization Methods and Software **15** (2001), no. 1, 1–28.

[85] M. Laurent, *A tour d'horizon on positive semidefinite and Euclidean distance matrix completion problems*, Topics in semidefinite and interior-point methods (Toronto, ON, 1996), Fields Inst. Commun., vol. 18, Amer. Math. Soc., Providence, RI, 1998, pp. 51–76.

[86] J.-P. Lecadre and P. Lopez, *Estimation d'une matrice interspectrale de structure imposée*, Traitement du Signal **1** (1984), 4–17.

[87] J.D. Louck, *Doubly stochastic matrices in quantum mechanics*, Foundations of Physics **27** (1997), no. 8, 1085–1104.

[88] J. Malick, *An efficient dual algorithm to solve conic least-square problems*, Tech. report, Institut National de recherche en Informatique et Automatique

(INRIA), 2001, To appear in Siam Journal on Matrix Analysis and Application under title : A dual approach for conic least-squares problems.

[89] M. Marcus and R. Ree, *Diagonals of doubly stochastic matrices*, The Quarterly Journalof Mathematics. Second Series. **10** (1959), 296–302.

[90] A.W. Marshall and I. Olkin, *Inequalities : Theory of Majorization and Its Applications*, Academic press, 1979, Mathematics in Sciences and Engineering, Volume 143.

[91] B. Monjardet, *Sur diverses formes de la "règle de Condorcet" d'agrégation des préférences*, Mathématiques Informatique et Sciences Humaines **111** (1990), 61–71.

[92] R. Monteiro and M. Todd, *Path-following methods*, Handbook of semidefinite programming, Internat. Ser. Oper. Res. Management Sci., vol. 27, Kluwer Acad. Publ., Boston, MA, 2000, pp. 267–306.

[93] R.D.C. Monteiro, *Primal-dual path-following algorithms for semidefinite programming*, SIAM Journal on Optimization **7** (1997), no. 3, 663–678.

[94] Y. Nesterov and A. Nemirovskii, *Interior-point polynomial algorithms in convex programming*, SIAM Studies in Applied Mathematics, vol. 13, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1994.

[95] Y.E. Nesterov and M.J. Todd, *Primal-dual interior-point methods for self-scaled cones*, SIAM Journal on Optimization **8** (1998), no. 2, 324–364 (electronic).

[96] J. Nocedal and S.J. Wright, *Numerical optimization*, Springer Series in Operations Research, Springer-Verlag, New York, 1999.

[97] C. Papadimitriou and K. Steiglitz, *Combinatorial optimization. Algorithms and complexity*, Prentice-Hall, 1982.

[98] G. Pierra, *Decomposition trough Formalization in a product space*, Mathematical Programming **28** (1984), 96–115.

[99] B.T. Polyak, *Random algorithms for solving convex inequalities*, Inherently parallel algorithms in feasibility and optimization and their applications (Haifa 2000) (D. Butnariu, Censor Y., and S. Reich, eds.), Studies in Computational Mathematics, vol. 8, Elsevier science, 2001, pp. 409–422.

[100] R.T. Rockafeller and R. J-B. Wets, *Variational Analysis*, Grundlehren der mathematischen Wissenchaften 317. Springer-Verlag Berlin Heidelberg, 1998.

[101] Y. Saad, *Iterative methods for sparse linear systems*, SIAM Studies in Applied Mathematics, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000, to appear. Got from the web at the url : http ://www-users.cs.umn.edu/ saad/books.html.

[102] A. Shapiro and K. Scheinberg, *Duality and optimality conditions*, Handbook of semidefinite programming, Internat. Ser. Oper. Res. Management Sci., vol. 27, Kluwer Acad. Publ., Boston, MA, 2000, pp. 67–110.

[103] C. Skiadas, *Conditioning and aggregation of preferences*, Econometrica. Journal of the Econometric Society **65** (1997), no. 2, 347–367.

[104] J. H. Smith, *Aggregation of preferences with variable electorate*, Econometrica **41** (1973), no. 6, 1027–1041.

[105] J.F. Sturm, *Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones*, Optimization Methods and Software **11/12** (1999), no. 1-4, 625–653, Interior point methods.

[106] P.L. Takouda, *Décomposition lagrangienne pour les problèmes d'optimisation avec variables entières*, Master's thesis, Université Paul Sabatier, Toulouse III, 1999, Mémoire de DEA Mathématiques Appliquées.

[107] _____, *Un problème d'approximation matricielle : quelle est la matrice bistochastique la plus proche d'une matrice donnée ?*, Tech. report, Laboratoire MIP, Université Paul Sabatier, Toulouse 3, 2002, Research Report MIP 02-21. Accessible sur le web à l'adresse :*http ://mip.ups-tlse.fr/publi/2002.html*. Soumis.

[108] _____, *Résolution d'un problème d'agrégation de préférence en approximant par des matrices bistochastiques.*, Mathématiques et Sciences Humaines, "Recherche opérationnelle et aide à la décision", 41e année **161** (2003), 77 – 97.

[109] M. J. Todd, *A study of search directions in primal-dual interior-point methods for semidefinite programming*, Optimization Methods and Software **11/12** (1999), no. 1-4, 1–46, Interior point methods.

[110] L. Vandenberghe and S. Boyd, *Semidefinite programming*, SIAM Review **138** (1996), no. 1, 49–95.

[111] D. Vanderpooten, *Aide multicritère à la décision ; quelques concepts et perspectives*, Exposé de synthèse aux Quatrièmes journées nationales de la ROADEF, Paris, février 2002, 2002.

[112] P. Vincke, *L'aide multicritère à la décision.*, Ellipses, Paris, 1989.

[113] J. Von Neumann, *Functionnal Operators, volume II. The geometry of Orthogonal spaces*, Annals of mathematical studies, vol. 22, Princeton university Press, 1950, Reprints of mimeographed lectures notes first distributed in 1933.

[114] H. Wolkowicz, *Solving semidefinite programs using preconditioned conjugate gradients*, Tech. report, Dept. of Combinatorics & Optimization, University of Waterloo, Canada, 2001, Research Report CORR 01-49, April 2001. Accessible on the web at the url :*http ://orion.math.uwaterloo.ca/ hwolkowi*. Submitted.

[115] H. Wolkowicz, R. Saigal, and L. Vandenberghe (eds.), *Handbook of semidefinite programming*, International Series in Operations Research & Management Science, 27, Kluwer Academic Publishers, Boston, MA, 2000, Theory, algorithms, and applications.

[116] S.J. Wright, *Primal-dual interior-point methods*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.

[117] H. P. Young, *Social choice scoring functions*, SIAM Journal on Applied Mathematics **28** (1975), no. 4, 824–838.

[118] E.H. Zarantonello, *Projections on convex sets in Hilbert spaces and spectral theory*, Contributions to Nonlinear Functionnal Analysis (E.H. Zarantonello, ed.), University of Wisconsin. Mathematics Research Center Publications, no. 27, Academic Press, New york, 1971, pp. 1–38.