Rational models optimized exactly for solving signal processing problems

Arthur Marmin

université
PARIS-SACLAY

Rational models optimized exactly for solving signal processing problems

Thèse de doctorat de l'Université Paris-Saclay

École doctorale n° 580, Sciences et Technologies de l'Information et de la Communication (STIC)
Spécialité de doctorat: Traitement du signal et des images
Unité de recherche: Université Paris-Saclay, CentraleSupélec, Centre de Vision Numérique, 91190, Gif-sur-Yvette, France
Référent: CentraleSupélec

Thèse présentée et soutenue en visioconférence totale, le 8 décembre 2020, par

Arthur MARMIN

Composition du jury:

Pascal Bondon Directeur de recherche CNRS, Université Paris-Saclay, CentraleSupélec, L2S (Gif-sur-Yvette, France)	Président du jury/Examinateur
Bodgan Dumitrescu Professor, University Politehnica of Bucharest, Department of Automatic Control and Computers (Bucarest, Roumanie)	Rapporteur/Examinateur
Didier Henrion Directeur de recherche CNRS, LAAS (Toulouse, France)	Rapporteur/Examinateur
Caroline Chaux Chargée de recherche CNRS, Université d'Aix-Marseille, I2M (CNRS UMR 7373) (Marseille, France)	Examinatrice
Laurent Albera Maître de conference, Université de Rennes 1, LTSI, INSERM U642 (Rennes, France)	Examinateur
Jean-Christophe Pesquet Professeur, Université Paris-Saclay, CentraleSupélec, Inria, CVN (Gif-sur-Yvette, France)	Directeur de thèse
Marc Castella Maître de conférence, Télécom SudParis, Institut Polytechnique de Paris, SAMOVAR (CNRS UMR 5157) (France)	Codirecteur de thèse
Laurent Duval Chef de projet, IFP Energies nouvelles (Rueil-Malmaison)	Co-encadrant

Thèse de doctorat

NNT: 2020UPASG017

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre: Modèles rationnels optimisés de manière exacte pour la résolution de problèmes de traitement du signal

Mots clés: Optimisation rationnelle, Problème des moments, Programmation semi-definie positive, Tenseur et décomposition CP, Traitement du signal

Résumé: Une vaste classe de problèmes d'optimisation non convexes est celle de l'optimisation rationnelle. Cette dernière apparaît naturellement dans de nombreux domaines tels que le traitement du signal ou le génie des procédés. Toutefois, trouver les optima globaux pour ces problèmes est difficile. Une approche récente, appelée la hiérarchie de Lasserre, fournit néanmoins une suite de problèmes convexes assurée de converger vers le minimum global. Cependant, cette approche représente un défi calculatoire du fait de la très grande dimension de ses relaxations. Dans cette thèse, nous abordons ce défi pour divers problèmes de traitement du signal.

Dans un premier temps, nous formulons la reconstruction de signaux parcimonieux en un problème d'optimisation rationnelle. Nous montrons alors que ce dernier possède une structure que nous exploitons afin de réduire la complexité des relaxations associées. Nous pouvons ainsi résoudre plusieurs problèmes pratiques comme la restoration de signaux de chromatographie. Nous étendons également notre méthode à la restoration de signaux dans différents contextes en proposant plusieurs modèles de bruit et de signal.

Dans une deuxième partie, nous étudions les relaxations convexes générées par nos problèmes et qui se présentent sous la forme de problèmes d'optimisation semi-définie positive de très grandes dimensions. Nous considérons plusieurs algorithmes basés sur les opérateurs proximaux pour les résoudre efficacement.

La dernière partie de cette thèse est consacrée au lien entre les problèmes d'optimisation polynomiaux et la décomposition de tenseurs symétriques. En effet, ces derniers peuvent être tous deux vus comme une instance du problème des moments. Nous proposons ainsi une méthode de détection de rang et de décomposition pour les tenseurs symétriques basée sur les outils connus en optimisation polynomiale. Parallèlement, nous proposons une technique d'extraction robuste des solutions d'un problème d'optimisation poylnomiale basée sur les algorithmes de décomposition de tenseurs. Ces méthodes sont illustrées sur des problèmes de traitement du signal.

Title: Rational models optimized exactly for solving signal processing problems

Keywords: Rational optimization, Moment problem, Semi-definite programming, Tensor and CP decomposition, Signal processing

Abstract: A wide class of nonconvex optimization problem is represented by rational optimization problems. The latter appear naturally in many areas such as signal processing or chemical engineering. However, finding the global optima of such problems is intricate. A recent approach called Lasserre's hierarchy provides a sequence of convex problems that has the theoretical guarantee to converge to the global optima. Nevertheless, this approach is computationally challenging due to the high dimensions of the convex relaxations. In this thesis, we tackle this challenge for various signal processing problems.

First, we formulate the reconstruction of sparse signals as a rational optimization problem. We show that the latter has a structure that we wan exploit in order to reduce the complexity of the associated relaxations. We thus solve several practical problems such as the reconstruction of chromatography signals. We also extend our method to the reconstruction of various types of signal corrupted by different noise models.

In a second part, we study the convex relaxations generated by our problems which take the form of high-dimensional semi-definite programming problems. We consider several algorithms mainly based on proximal operators to solve those high-dimensional problems efficiently.

The last part of this thesis is dedicated to the link between polynomial optimization and symmetric tensor decomposition. Indeed, they both can be seen as an instance of the moment problem. We thereby propose a detection method as well as a decomposition algorithm for symmetric tensors based on the tools used in polynomial optimization. In parallel, we suggest a robust extraction method for polynomial optimization based on tensor decomposition algorithms. Those methods are illustrated on signal processing problems.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France

Remerciements

Je souhaite commencer par remercier mes encadrants de thèse qui ont joué des rôles parfaitement complémentaires lors de ses trois dernieres années. Mes discussions avec Laurent Duval ont toujours été un plaisir et j'ai apprécié ses conseils de lecture scientifique et de méthodologie. Jean-Christophe Pesquet a toujours été de bons conseils et a su me donner un cap tout au long de la thèse. Enfin Marc Castella s'est toujours montré présent pour discuster de mes avancées, m'aider dans mes réflexions et répondre à mes questions. Cette thèse est ainsi le résultat d'un travail collaboratif et ne serait pas ce qu'elle est aujourd'hui sans l'investissement complet de chaucn de mes encadrants.

Je souhaite ensuite remercier tous les membres de mon jury qui ont accepté de consacrer de leur temps à la lecture de cette thèse. En particulier, les professeurs Bogdan Dumitrescu et Pascal Bondon m'ont accompagné depuis mon évaluation de mi-parcours de thèse et ont su m'encourager depuis ce moment là. Je remercie les professeurs Bogdan Dumitrescu et Didier Henrion qui ont accepté de jouer le rôle de rapporteurs et ont contribué à la qualité du présent manuscrit par leur remarques et questions pertinentes. Enfin, je remercie les professeurs Caroline Chaux et Laurent Albera d'avoir accepté de participer à mon jury et d'avoir soulevé des pistes intéressantes.

Pendant ma thèse, j'ai eu la chance de pouvoir travailler avec Luis Briceño Arias durant ses séjours dans notre laboratoire. J'ai également été chaleureusement accueilli à l'université technologique Gdansk par le professeur Ania Jezierska. J'ai eu grand plaisir à collaborer avec eux et je leur suis reconnaissant pour leur aide et leur soutien.

Je remercie également Brice Hannebicque pour m'avoir aidé dans mes enseignements tout au long de ces trois années. J'ai beaucoup appris de son expérience et je lui suis redevable d'avoir pu progresser dans ma vision de l'enseignement.

Toute ma gratitude va enfin aux membres du CVN qui m'ont accueilli durant ces trois années. Grâce à eux, j'ai bénéficié de l'environement de travail idéal m'ayant permis de mener à bien cette thèse. Un grand merci à Jana pour son efficacité ainsi que son soutien logistique précieux, et à Anthony pour sa gestion des ressources informatiques. Merci également aux professeurs du laboratoire, notamment Emilie et Fragkiskos, avec qui j'ai pu échanger et qui m'ont guidé dans ma vie professionnelle de jeune chercheur. Mes derniers mots seront pour tous mes camarades, l'ancienne génération — Hari, Alp, Stefan, Marie-Caroline —, mes compères de thèse — Kadir, Kavya, Maissa, Maria P., Sagar, Yun Shi, Ying Ping — et enfin la nouvelle génération — Ana, Jean-Baptiste, Marion, Mario, Ségolène, Tasnim. Une pensée particulière pour Mihir, Maria V. et Matthieu qui ont été de précieux amis durant ces années franciliennes.

Résumé

Cette thèse de doctorat porte sur l'application des méthodes d'optimisation polynomiale et rationnelle à des problèmes du traitement du signal. Récemment, plusieurs méthodes d'optimisation globales ont été développées pour des problèmes polynomiaux. Ces dernières sont basées sur des certificats de positivité pour les polynômes et garantissent sous certaines conditions, de converger vers l'optimum global et de retrouver tous les minima globaux. En contrepartie, ces méthodes sont souvent très exigeantes en coût de calcul et leurs applications sont encore limitées. Dans cette thèse, nous nous concentrons sur la méthode connue sous le nom de la hiérarchie de Lasserre et nous montrons que cette dernière peut être utilisée pour résoudre avec succès plusieurs problèmes difficiles et de taille modérée en traitement du signal. Nous illustrons notamment comment la structure de ces problèmes peut être utilisée pour diminuer la complexité calculatoire de l'algorithme.

La hiérarchie de Lasserre consiste à reformuler les problèmes d'optimisation polynomiale ou rationelle dans l'espace des mesures positives afin de les linéariser. Le problème de mesure est ensuite transformé en un problème de moments, avant que la séquence de ces moments ne soient tronqués jusqu'à un certain ordre pour donner une suite de relaxations convexes sous forme de problèmes d'optimisation semi-définie positives (SDP). La complexité calculatoire de la hiérarchie est alors due à la grande dimension de ces relaxations SDP. Nous rappelons les principes de la hiérarchie de Lasserre dans le Chapitre 3 ainsi que sa généralisation à la minimisation d'une somme de fonctions rationnelles. Les problèmes auxquels est ramené le problème de départ et qui sont résolus numériquement sont ici des problèmes SDP. Nous étudions donc la complexité de ces derniers en fonction des données de départ du problème rationnel.

La première motivation de notre travail a été la résolution de problèmes inverses pour la reconstruction de signaux de chromatographie. Deux difficultés se présentent: les signaux reconstruits doivent posséder une structure parcimonieuse et la reconstruction doit prendre en compte des effets non-linéaires, tels que la saturation. Ces deux obstacles aboutissent à la résolution de problèmes d'optimisation non-convexes. Nous montrons dans le Chapitre 4 que ces problèmes peuvent être traduits sous la forme d'un problème d'optimisation rationnelle. En effet, de nombreuses approximations continues et exactes de la fonction ℓ_0 sont polynomiales par morceaux. Nous appliquons alors la hiérarchie de Lasserre en tirant parti de la structure du problème pour reconstruire le signal parcimonieux recherché. Notre étude de la complexité des relaxations SDP montre que le défi calculatoire de la méthode de Lasserre peut être relevé pour ces problèmes de reconstruction de signaux parcimonieux dans un contexte où des non-linéarités apparaissent dans le modèle. De plus, nous traitons dans notre modèle la contrainte expérimentale où seul un signal sous-échantillonné est observé. Cette limitation sur le signal observé émerge naturellement lors de l'acquisition de nombreux échantillons en un temps limité.

De nombreuses fonctions d'intérêt dans les applications de traitement du signal sont rationnelles par morceaux ou peuvent être approchées précisément par une telle fonction.

Dans le Chapitre 5, nous traitons notamment de la reconstruction de signaux corrompus par un bruit de Poisson-Gauss. Nous montrons notamment que le calcul de l'estimateur de maximum de vraisemblance peut être approché par un problème d'optimisation rationelle que nous résolvons en utilisant le cadre développé aux chapitres précédents. Nous étendons également notre méthode à des signaux parcimonieux dans un domaine transformé comme par exemple des signaux dont le gradient discret est parcimonieux. Nous illustrons nos résultats sur la reconstruction de signaux parcimonieux et de signaux de communication en lumière visible (VLC). Une autre classe intéressante de fonctions rationnelles par morceaux est la classe des fonctions objectifs robustes au bruit des données aberrantes. En effet, les résultats expérimentaux contiennent souvent quelques valeurs aberrantes pour lesquelles nous ne souhaitons pas grandement pénaliser la vraisemblance pour ne pas fausser notre estimateur. Ainsi la fonction de Huber ou les fonction ℓ_1 et ℓ_2 tronquées sont des exemples de fonctions robustes rationnelles par morceaux. Nous montrons que notre méthode s'adapte parfaitement à ces fonctions robustes et donne de meilleures reconstructions. Nous traitons également de contraintes exprimées sous la forme d'union de sous-ensembles. Bien que difficiles à faire respecter par de nombreuses méthodes d'optimisation convexe, ces contraintes peuvent s'exprimer sous la forme de contraintes polynomiales et donc rentrer dans notre cadre.

Afin d'étendre l'application de notre méthode des chapitres précédents à des problèmes de plus grandes dimensions, nous développons dans le Chapitre 6, des algorithmes de résolution pour des problèmes SDP de grande tailles. En effet, les limites calculatoires se situent dans la résolution de ces derniers par les méthodes de points intérieurs qui sont très vite limitées lorsque la dimension et le nombre de variables augmentent. Nous explorons notamment une dizaine d'algorithmes basés sur l'opérateur proximal ainsi que quelques méthodes basées sur les points intérieurs. Bien que certains de ces algorithmes se révèlent meilleurs que les méthodes de point intérieurs pour des situations données, ils ne peuvent rivaliser avec eux pour la résolution des relaxations SDP de problèmes d'optimisation polynomiale sous contraintes.

La dernière partie de cette thèse est consacrée au lien entre la décomposition d'un tenseur symétrique en une somme de tenseurs de rang un, appelée décomposition canonique polyadique (CP), et les problèmes d'optimisation polynomiale. En effet, nous réécrivons dans le Chapitre 7, la décomposition CP d'un tenseur symétrique en un problème de moment. En utilisant les outils du problème des moments, nous dérivons alors une méthode de détection de rang ainsi qu'une méthode de décomposition CP. Cette dernière est inspirée par l'extraction des minima dans la hiérarchie de Lasserre et est basée sur un algorithme de résolution d'un système de polynômes. Dans un second temps, nous effectuons le cheminement inverse et utilisons les méthodes de décomposition CP robustes, telles que les moindres carrés non linéaires, pour effectuer une extraction des minima globaux dans la hiérachie de Lasserre à partir d'un vecteur de moments bruité. En effet, la méthode algébrique classiquement utilisée est très sensible au bruit. L'intérêt de notre proposition est donc de pouvoir retrouver les minima globaux lorsque seuls les premiers problèmes SDP de la hiérarchie peuvent être résolus numériquement et que donc seul un nombre limité de moments bruités est disponible.

Finalement, dans le Chapitre 8, nous résumons nos différentes contributions et nous suggérons plusieurs pistes afin d'étendre les résultats présentés dans cette thèse.

Symbols

$\lfloor \cdot \rfloor$	: Greatest integer lower than its argument
$\lceil \cdot \rceil$	: Smallest integer greater than its argument
$\binom{n}{k}$	: Binomial coefficient "among n choose k"
$\mathbb{S}^n$	: Set of $n \times n$ real symmetric matrices
$\mathbb{S}^n_+$	: Set of $n \times n$ real symmetric positive semi-definite matrices
$\mathbb{S}^n_{++}$	: Set of $n \times n$ real symmetric positive definite matrices
$\mathbb{S}^n_-$	: Set of $n \times n$ real symmetric negative semi-definite matrices
$\mathbb{N}^n_t$	: Subset of n-tuples of natural integers whose absolute value is less than or equal to t
$\mathbb{R}[\mathbf{x}]$	: Set of multivariate polynomials in $\mathbf{x}$ having real coefficients
Σ_n	: Cone of Sum-of-Squares polynomials on n variables
$\mathcal{M}_+(\mathcal{X})$	: Cone of positive measures supported on $\mathcal{X}$
$\mathcal{D}(\mathcal{X})$	: Cone of the infinite moment vectors of positive measure supported on $\mathcal{X}$
$^\top$	: Transpose of a matrix
$*$	: Convolution operator
$\otimes$	: Kronecker product
$\odot$	: Hadamard product
$\langle . \mid . \rangle$	: Inner product
ld	: Barrier function on $\mathbb{R}^{n \times n}$
$\mathbb{1}_{\mathcal{X}}$	: Characteristic function of $\mathcal{X}$
$\iota_{\mathcal{X}}$	: Indicator function of $\mathcal{X}$
$\Pi_{\mathcal{X}}$	: Projection on $\mathcal{X}$
$\mathcal{N}_{\mathcal{X}}(x)$	: Normal cone to $\mathcal{X}$ at x
Id_n	: Identity matrix of $\mathbb{R}^{n \times n}$
Diag	: Diagonalizing operator

Acronyms

ADAL Alternative Direction Augmented Lagrangian method

ADMM Alternative Direction Method of Multipliers

CPD Canonical Polyadic Decompostion

DSoS Diagonally-dominant-Sum-of-Squares

FB Forward-Backward algorithm

FBHF Forward-Backward Half-Forward Algorithm

FISTA Fast Iterative Shrinkage-Thresholding Algorithm

FPR False Positive Rate

GAST Generalized Anscombe Transform

KKT conditions Karush-Kuhn-Tucker conditions

LP Linear Programming

MAP Maximum A Posteriori

PG Poisson-Gaussian

PSNR Peak Signal to Noise Ratio

REP Relative Entropy Programming

RFBPD Rescaled Forward-Backward Primal-Dual algorithm

SDP Semi-Definite Programming

SDSoS Scaled-Diagonally-dominant-Sum-of-Squares

SOCP Seconde-Order Cone Programming

SONC Sum-of-Non-negative Circuit

SoS Sum-of-Squares

TPR True Positive Rate

WLS Weighted Least Squares

Contents

- Chapter 1 -

General Introduction

§ 1.1 Context

The work in this thesis was firstly motivated by the reconstruction of chromatography signals from their corrupted measurements through sensors in chemical engineering. This inverse problem can be solved by variational methods which consist in minimizing a well-chosen criterion. Due to the advances in convex optimization of the last decades, variational methods often result in the minimization of a convex criterion (often under convex constraints) that is successfully solved with one of the many efficient existing algorithms.

However, although convex optimization has become a cornerstone of engineering, many realistic physical models, such as the reconstruction of chromatography signals, are more accurately modelled by nonconvex formulations for which scarcer theoretical guarantees are available. A research direction consists of convexifying problems to benefit from previously developed theory and tools. The main difficulty resides in finding a convex equivalent problem or a faithful convex relaxation of the original problem leading hopefully to global optimal solutions. Recent theoretical results in real algebraic geometry show that one can successfully follow this path when dealing with optimization of rational functions (i.e. functions that are ratios of two polynomials) under polynomial constraints. In this context, remarkable properties indeed hold providing global optimal guarantees.

Polynomial and rational functions encompass a wide class of practically relevant models. In view of approximation theory, the latter functions provide optimal approximations, globally or piecewise, for most functions of practical interest. Hence, they are nowadays the workhorses for many numerical computations. As a consequence, rational optimization can be used to approximate most optimization problems. For instance, optimization problems mixing real-valued and binary variables can be translated into polynomial problems.

In this work, we focus on the application of the moment/SoS hierarchy, also known as Lasserre's hierarchy, to relax rational optimization problems into a hierarchy of convex problems. While being a breakthrough from a theoretical standpoint, the obtained relaxations have very high dimensions that unfortunately often prohibit any numerical resolution in a reasonable time. Our main goal here is to show the benefit of the moment/SoS hierarchy in practical signal processing problems and overcome the computational limitations. More specifically, we look at various inverse problems for signal reconstruction involving nonlinear models and nonconvexity. For instance, our framework is relevant to deal with models in gas chromatography.

Leveraging the structures of the latter problems, we are able to reduce drastically the complexity of the relaxations and therefore solve them in a fair amount of time. After studying the general complexity of the convex relaxation, we study the beneficial impact of exploiting these structures and develop algorithms to solve them efficiently.

We show in numerical simulations that our proposed framework is able to solve various challenging signal processing problems and compares favorably to existing approaches. Hence, the outcomes of our work show that global optimization methods emerging from real algebraic geometry, although known to be computationally intensive, can have a practical impact in signal processing applications.

This work has been done in the framework of a fruitful collaboration with IFP Energies nouvelles, and more specifically Dr Laurent Duval who was at the origin of the applicative motivation of this PhD work.

§ 1.2 Outline

This manuscript is organized as follows.

Chapter 2 first exposes through the light of approximation theory why polynomial and rational functions form an important class of modelling functions. Afterwards, it presents the general rational optimization problem as well as the different strategies to solve it. The connection between rational optimization and certificates of positivity for polynomials is then recalled and recent optimization frameworks based on these certificates are summarized.

In Chapter 3, we detail Lasserre's hierarchy, the current state-of-the-art framework for global optimization of rational problems. We then propose to adapt this framework for problems that have some specific structures and study the complexity of the resulting method. The latter forms the foundation of our work.

In Chapter 4, we apply the previous methodology to the reconstruction of a nonlinearly distorted and subsampled sparse signal. This problem is highly challenging due to its nonconvexity and its high computational complexity. Moreover, it has practical interest in various areas such as gas chromatography. We thereby detail how to use the structure of the problem in order to solve it in a fair amount of time. Numerical simulations illustrate the performance of our method.

Chapter 5 extends the results of Chapter 4 in several ways. First, we consider different fit functions that are successfully used to handle Poisson-Gaussian noise as well as outliers. Second, we also extend our framework to signals sparse under a linear transformation such as a discrete gradient. This adds extra complexity in our final relaxation. Finally we show how our framework can successfully handle some nonconvex constraints expressed as unions of subsets that cannot be handled by standard methods.

Chapter 6 proposes several proximal algorithms to solve Semi-Definite Programming (SDP) problems arising from Lasserre's hierarchy. Those SDP problems are high-dimensional, both in terms of variables and constraints and are challenging for current interior point methods. We explore several possibilities on the original problems and suggest some equivalent problems that are easier to solve.

Chapter 7 sets out the link between tensors decomposition and the moment problem. Using results from the literature on the truncated moment problem, we derive a rank detection and a Canonical Polyadic Decompostion (CPD) algorithm for symmetric tensors. On the other hand, we propose a robust extraction method for rational optimization problem based on tensor decomposition methods.

Finally, Chapter 8 draws conclusions and gives several perspectives.

§ 1.3 Contributions

We summarize here the contributions of this work.

In Chapter 3, we study a specialization of Lasserre's hierarchy for structured problems. More specifically, we use the recent work of Bugarin et al. [BHL15] to adapt the general framework of Lasserre to a sum of rational functions in a few variables. Our main contribution here is a study of the complexity of the obtained relaxations. We show that this approach is important to reduce the computation complexity in signal processing applications.

In Chapter 4, our contribution is twofold:

(i) We investigate a wide range of continuous approximations to the ℓ_0 penalty and we extend the framework of Lasserre's hierarchy to piecewise rational functions in order to minimize the resulting nonconvex criterion. Unlike standard approaches, we are able to find and certify the global optimum of this nonconvex optimization problem. Compared to previous works, the class of regularizers is much richer and is combined with a nonlinear observation model and a subsampling.

(ii) Through a complexity analysis and extensive simulations, we show how the structure of the problem and the subsampling allow us to alleviate the computational burden of the original framework presented in Chapter 3. Our approach can be successfully applied to signal processing and compressed sensing problems as illustrated by the provided example inspired by the acquisition of signals in gas chromatography.

In Chapter 5, we extend the previous methodology to noise distributions that differ from the standard additive white Gaussian model.

(i) We propose a new rational approximation to the Poisson-Gaussian data fidelity term and we show that rational approximations to the ℓ_0 sparsity measure introduced in Chapter 4 can be combined efficiently with a linear operator such as a discrete gradient. The resulting nonconvex Maximum A Posteriori (MAP) problem is then successfully solved with our methodology of Chapter 3 and practical examples illustrate the good performance.

(ii) We adapt our methodology to make it robust to the presence of possible outliers. Moreover we consider extra nonconvex constraints on the signals that are modelled as unions of subsets. While this type of constraints cannot be solved, we show that our methodology can handle them.

In Chapter 6, we develop many proximal algorithms to handle high-dimensional SDP problems and our contributions are as follows.

(i) We derive several proximal algorithms to solve SDP problems in their primal, dual and primal-dual forms.

(ii) We propose several equivalent reformulations of SDP problems together with algorithms to solve them. We also suggest two algorithms inspired by interior point methods.

(iii) We conduct extensive simulations to determine the most suitable algorithm depending on the dimensions of the SDP problem. Especially, we show that, although some proximal methods are more interesting to solve unconstrained polynomial problems, interior point methods are currently the most efficient solvers for SDP problems emerging from our methodology.

In Chapter 7, we explore the link between tensor decomposition and moment problem. This yields the three following main results

(i) We propose a symmetric rank detection method for symmetric tensors based on the tool of moment problem. This method is based on a necessary condition on the rank of moment matrices built with the tensor elements. We illustrate it on a blind source separation problem.

(ii) We develop an algebraic method for tensor CPD that yields higher accuracy than standard optimization-based methods in a noiseless context. The latter method is based on the extraction step in polynomial optimization problem that amounts to solving a polynomial system.

(iii) We provide a robust extraction method for polynomial optimization. In contrast to the standard extraction method, our method allows us to recover the exact global minima from a noisy moment vector. This is of practical interest to reduce the computational burden and to cope with the inaccuracy of the solver for the convex relaxations.

§ 1.4 Related publications

The work in this thesis has given rise to the followings publications:

Published journal articles

- Marc Castella, Jean-Christophe Pesquet, and Arthur Marmin. Rational optimization for nonlinear reconstruction with approximate ℓ_0 penalization. *IEEE Trans. Signal Process.*, 67(6):1407–1417, March 2019 [link]

- Arthur Marmin, Anna Jezierska, Marc Castella, and Jean-Christophe Pesquet. Global optimization for recovery of clipped signals corrupted with poisson-gaussian noise. *IEEE Signal Process. Lett.*, 27:970–974, May 2020 [link]

- Arthur Marmin, Marc Castella, Jean-Christophe Pesquet, and Laurent Duval. Sparse signal reconstruction for nonlinear models via piecewise rational optimization. *Signal Process.*, 179:107835, February 2021 [link]

Conference proceedings

- Arthur Marmin, Marc Castella, Jean-Christophe Pesquet, and Laurent Duval. Signal reconstruction from sub-sampled and nonlinearly distorted observations. In *Proc. European Signal Processing Conference*, pages 1970–1974. IEEE, September 2018 [link]

- Arthur Marmin, Marc Castella, and Jean-Christophe Pesquet. How to globally solve non-convex optimization problems involving an approximate ℓ_0 penalization. In *Proc. Int. Conf. Acoust. Speech Signal Process.*, pages 5601–5605. IEEE, May 2019 [link]

- Arthur Marmin, Marc Castella, and Jean-Christophe Pesquet. Sparse signal reconstruction with a sign oracle. In *Proc. Signal Processing with Adaptive Sparse Structured Representations (SPARS) workshop*, July 2019 [link]

- Arthur Marmin, Marc Castella, and Jean-Christophe Pesquet. Detecting the rank of a symmetric tensor. In *Proc. European Signal Processing Conference*, pages 1–5. IEEE, September 2019 [link]
- Arthur Marmin, Marc Castella, and Jean-Christophe Pesquet. A moment-based approach for guaranteed tensor decomposition. In *Proc. Int. Conf. Acoust. Speech Signal Process.*, pages 3927–3931. IEEE, May 2020 [link]
- Arthur Marmin, Marc Castella, and Jean-Christophe Pesquet. Globally optimizing owing to tensor decomposition. In *Proc. European Signal Processing Conference*. IEEE, September 2020. to appear
- Arthur Marmin, Marc Castella, and Jean-Christophe Pesquet. Robust reconstruction with nonconvex subset constraints: A polynomial optimization approach. In *IEEE Int. Workshop Mach. Learn. Signal Process.* IEEE, September 2020 [link]

§ 1.5 General notation

Throughout this thesis, we use the following notation: $\mathbb{N}$ denotes the set of nonnegative integers, $\mathbb{R}$, $\mathbb{R}_+$, and $\mathbb{R}^*$ denote respectively the sets of real, positive real, and non-zero real numbers. For any nonnegative integer n, $\mathbb{S}^n$, $\mathbb{S}^n_+$, and $\mathbb{S}^n_{++}$ are respectively the set of $n \times n$ real symmetric, symmetric positive semi-definite, and symmetric positive definite matrices. Id_n is the identity matrix of size $n \times n$. For any pair of nonnegative integers n and k, $\binom{n}{k}$ is the binomial coefficient "among n choose k" given by $\frac{n!}{k!(n-k)!}$. $\lfloor\cdot\rfloor$ (resp. $\lceil\cdot\rceil$) denotes the greatest (resp. smallest) integer lower (resp. greater) than its argument. For a multi-index $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)$ of $\mathbb{N}^n$, $|\boldsymbol{\alpha}| = \alpha_1 + \cdots + \alpha_n$ denotes its absolute value and $\mathbb{N}^n_t$ is the subset of multi-indices in $\mathbb{N}$ whose absolute value is less than or equal to t. We define the graded lexicographic ordering for two multi-indices $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ as follows: we write $\boldsymbol{\alpha} > \boldsymbol{\beta}$ if $|\boldsymbol{\alpha}| > |\boldsymbol{\beta}|$ or if $|\boldsymbol{\alpha}| = |\boldsymbol{\beta}|$ and the leftmost non-zero entry of $\boldsymbol{\alpha} - \boldsymbol{\beta}$ is positive.

Upper case calligraphic letters denote tensors ($\mathcal{T}$), bold upper case letters ($\mathbf{M}$) denote matrices, bold lower case letters ($\mathbf{v}$) denote vectors, and lower case letters (s) denote scalars. We use the same script to denote a linear application and its corresponding matrix representation.

The superscript $^\top$ indicates the transpose of a matrix, Diag is the operator that creates a diagonal matrix with its arguments on the diagonal. Scalar, Kronecker, and Hadamard products are respectively denoted with $\langle ., . \rangle$, $\otimes$, and $\odot$. For a given set $\mathcal{X}$, $\mathbb{1}_{\{\cdot \in \mathcal{X}\}}$ is the characteristic function of $\mathcal{X}$ with $\mathbb{1}_{\{x \in \mathcal{X}\}} = 1$ if x is in $\mathcal{X}$ and 0 otherwise. For a given polynomial $p(x) = \sum_{|\boldsymbol{\alpha}| \leq \text{degree}\, p} p_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}}$, we define the following operator d

$$\mathrm{d}_p = \left\lceil \frac{\text{degree}\, p}{2} \right\rceil , \tag{1.1}$$

and we denote by $\mathbf{p}$ a vector composed of the coefficients corresponding to monomials in p up to the total degree of p.

- Chapter 2 -

Background on polynomial optimization

§ 2.1 Polynomial and rational functions as modelling tools

The large flexibility of polynomial and rational functions has made them a vigorous research topic for ages. Their great modelling power makes them one of the pillars of approximation theory and they are consequently often used as tractable surrogates for more intricate functions.

A systematic use of power series to approximate transcendental functions was started in the late sixteenth century and early seventeenth century by Newton and Taylor. However, those formulae do not provide a uniform approximation and the error beyond the expansion point usually grows quickly. At the same time, Gauss developed the least squares regression method but the latter can still undergo an arbitrary high error at a given point. A specific approximation of periodic continuous functions using trigonometric series instead of power series was suggested by Fourier in the early eighteenth century. The essential step in approximation theory is the Stone-Weierstrass approximation theorem that allows uniform approximations of any continuous function defined on a closed interval by polynomials. A generalization of this result to functions defined on a compact subset of the complex plane was proved by Mergelyan in 1951. Bernstein and Chebyshev polynomials provide two different constructive examples of such approximations to any continuous function on the interval $[0, 1]$. Moreover, the Remez exchange algorithm proposed in 1934 builds, for a given degree, a polynomial approximation of a continuous function on a bounded interval that minimizes the error in the sense of the uniform norm.

In contrast, local polynomial approximations were also considered. Gluing together such local approximations under some smoothness conditions gave birth to piecewise approximations called splines. Especially, cubic splines have been widely used because of their quick computation. Refinement of splines based on Bernstein polynomials yields the Beziers curves and their generalization, the B-splines and the NURBS (Non-Uniform Rational Basis Splines) which form now the cornerstone of computer graphics and computer-aided design.

Although polynomials yield good approximations, rational approximations require less terms and also provide smoother and better approximations in the neighborhood of a singularity or a discontinuity by avoiding oscillation effects. As early as the seventh century, Bhāskarā I exposed an approximation of sinus as the ratio of two quadratic polynomials. In the late nineteenth century, Padé gave a systematic construction of a good rational approximation of an analytic function. A benefit of Padé approximants is

the high accuracy obtained only with a few terms in the involved polynomial compared to polynomial approximations. Indeed, Padé approximants can be interpreted as the non-linear acceleration of an expansion into powers of x or x^{-1} known as the generalized Shanks transformation. Acceleration of convergence aims to construct a new sequence that converges to the same limit but faster than the initial one. The Padé approximants can be viewed as truncations of continued fractions.

More details about approximation theory and its link with polynomial and rational functions can be found in the book [Pow81].

§ 2.2 Global rational optimization

In view of Section 2.1, polynomial regressions, interpolations and approximations are therefore workhorses for many numerical methods. The realm of rational functions has subsequently pervaded many scientific fields and consequently the problem of finding global extrema of multivariate rational criteria over polynomial constraints emerges naturally. As the involved criteria and constraints are not necessarily convex, finding global solutions is intricate as well-known convex optimization methods become only local. The general topic of this thesis consists in solving rational optimization problems which takes, for a given dimension T, the following general form

$$\begin{aligned} \mathcal{J}^* = \underset{\mathbf{x}\in\mathbb{R}^T}{\text{minimize}} \quad & \frac{p(\mathbf{x})}{q(\mathbf{x})} \\ \text{subject to} \quad & \mathbf{x} \in \mathcal{K}\,, \end{aligned} \tag{2.1}$$

where p and q are polynomial functions on $\mathbb{R}^T$ and, for a collection of J polynomials $S = (s_j)_{j\in[\![1,J]\!]}$, the feasible set $\mathcal{K}$ is a subset of $\mathbb{R}^T$ defined by polynomials inequalities

$$\mathcal{K} = \left\{ \mathbf{x} \in \mathbb{R}^T \mid (\forall j \in [\![1, J]\!]) \quad s_j(\mathbf{x}) \geq 0 \right\}. \tag{2.2}$$

Such a set $\mathcal{K}$ is said to be a closed basic semi-algebraic set [AB08, Las09]. When q is the constant polynomial equal to 1, Problem (2.1) includes the case of polynomial optimization. Note that if p and q have no common roots on $\mathcal{K}$, then a sufficient condition for $\mathcal{J}^*$ to not be equal to $-\infty$ is that q should have a constant sign on $\mathcal{K}$. Hence, we assume that q is strictly positive on $\mathcal{K}$.

Finding the global optimal value $\mathcal{J}^*$ as well as the feasible points where it is reached is a critical problems in many applications. Indeed, most optimization problems can be approximated by a problem in the form of (2.1). Some examples are provided in the next section.

2.2.1 Examples in signal processing

We give here a few uses of rational functions in signal processing:

- The Rational Function Model [TH01] used for the geometric processing of satellite or remote sensing images is based on rational approximation. Physical-based models were first used to link the pixel coordinates to the object coordinates in the 3D scene. However, those models are computationally expensive and, in order to perform real-time processing, rational approximations are used as a substitute and proved to be highly accurate. Ratio of first and second order terms model respectively optical distortion and correction due to Earth curvature and atmospheric refraction. Ratio of higher orders may be included to model unknown noise such as camera vibration.

- Using the Z-transform of discrete signals, the transfer function of IIR digital filters becomes a rational function. For a FIR filter, the Z-transform is a polynomial in z^{-1} which can be seen also as a rational function. Designing filters is then often equivalent to the minimization of such rational function under polynomial constraints. For instance, one can consider the design of a passband FIR filter of maximum magnitude where the constraints linking the magnitude to the stop frequencies and the attenuation coefficients are polynomial inequalities [Dum07]. In a similar spirit, the design of filter banks such as the two-channel conjugate quadrature filter banks can be interpreted as the minimization of a polynomial under polynomial constraints [YL09],

- In the control of congestion in communication networks, one goal is to maximize the utility of the network with respect to the flow constraints corresponding of the capacity of each link. Interesting utility functions are often modelled with rational function such as sigmoidal-like functions for voice applications [Chi09].

- The aim of optimal design is to plan an optimal set of experiments in order to find the coefficients of a large class of regression models, usually rational regression models. There are different classes of optimal design depending on the statistical criterion to be optimized such as A-optimal, D-optimal, and T-optimal. All those problems can be unified as a root search for a given polynomial [Pap12].

- A recent trend in machine learning has been the emergence of neural networks and their numerous applications. Building neural networks consists in approximating unknown functions of interest using a class of highly tunable functions composed of a large number of parameters. For some kinds of networks, this class shares a close link with rational functions: it does not only approximate well rational functions but is also well approximated by such functions [HW92, Tel17]. The ubiquity of these neural networks should therefore not be surprising in light of approximation theory.

In this thesis, we will mainly focus on the reconstruction of various corrupted signals.

2.2.2 Different classes of algorithms

Problem (2.1) is highly challenging in general due its nonconvexity. Indeed, many local optima may trap standard algorithms used in convex optimization. Note that in some very specific cases, even if Problem (2.1) is nonconvex, it possesses hidden convexity and can be solved globally in polynomial time with a convex optimization method [BTT96, KS15]. However, these are rare and some global optimization methods have been developed for the general case. They mainly divide into two families:

- Deterministic methods which offer theoretical guarantees on the optimality of the solutions. It includes the exact convex relaxations as well as the branch-and-bound/branch-and-reduce algorithms [RS96, BD14]. There exist also iterative numerical methods that guarantee, after a finite number of steps, to return a point at which the value of the criterion is the global optimum within a chosen tolerance. An example of such method is the successive incumbent transcending scheme [Tuy16] inspired from monotonic optimization. The main drawback of this family is its expensive computation time.

- Randomized methods that aims to solve problems more quickly than deterministic methods at the cost of losing the theoretical guarantees on global optimality. The underlying idea is to explore the feasible space in a clever fashion but with a

certain randomness in order to decrease the computations. It includes for instance the simulated annealing and the particle swarm optimization.

In the following, we focus exclusively on deterministic methods as one of the goal of this thesis to obtain theoretical guarantees on the retrieved solutions. Moreover, we do not look at general methods from global optimization that are not limited to rational objective functions. More specifically, we only deal with methods from real algebraic geometry, i.e. methods that rely on the properties of polynomials, especially their certificates of positivity. The remaining of this chapter is dedicated to provide an overview of such methods.

§ 2.3 Hilbert 17$^{\text{TH}}$ problem and Positivstellensätze

2.3.1 Rational optimization and certificates of positivity

Rational optimization is closely connected to the certificate of positivity of polynomials. Indeed, Problem (2.1) can be written as the maximization of a lower bound γ on $\mathcal{J}^*$

$$\begin{aligned} \gamma_* = \underset{\gamma\in\mathbb{R}}{\text{maximize}} \quad & \gamma \\ \text{subject to} \quad & (\forall \mathbf{x} \in \mathcal{K}) \quad p(\mathbf{x}) - \gamma q(\mathbf{x}) \geq 0\,. \end{aligned} \tag{2.3}$$

Problem (2.3) has solutions only if the polynomial $p - \gamma q$ is non-negative on $\mathcal{K}$ for some values of γ. The feasibility problem hence amounts to determine whether a polynomial is non-negative on a closed basic semi-algebraic set. This problem is known to be NP-hard [DG13] and certificates of positivity are used as substitutes.

2.3.2 Certificates of positivity on $\mathbb{R}^T$

Certificates of positivity for polynomials have a long history. It started with the decomposition into a Sum-of-Squares (SoS) of polynomials, which are clearly non-negative on the whole space $\mathbb{R}^T$. Especially, Hilbert proved that all polynomials taking non-negative values on $\mathbb{R}^n$ are exactly the polynomials that can be decomposed into SoS for the following three cases: univariate polynomials ($T = 1$), quadratic poynomials, and bivariate polynomials ($T = 2$) of degree 4. He also showed in 1888 the existence of a non-negative bivariate polynomial of degree 6 which is not a SoS. Nevertheless, his proof does not give an explicit construction and the first constructive example was given by Motzkin in 1967 with the famous bivariate polynomial given below

$$p(x, y) = x^4y^2 + x^2y^4 + 1 - 3x^2y^2\,. \tag{2.4}$$

The impossibility of the SoS decomposition follows from the arithmetic-geometric mean inequality. On the other hand, Hilbert proved that any bivariate polynomial can be written as a sum of squares of rational functions. The generalization of the latter result for any dimension T gave birth to the famous Hilbert's 17$^{\text{th}}$ problem, which was proved by Artin in 1927.

2.3.3 Certificates of positivity restricted to compact sets

The next step was to search for similar certificates of positity on closed basic semi-algebraic sets of the form of $\mathcal{K}$ instead of the whole space. In the second half of the twentieth century, a first result in this direction is Krivine-Stengle's Positivstellensatz [Ste74].

For a collection of polynomials $S = (s_j)_{j\in[\![1,J]\!]}$, we defined the associated preorder T_S as

$$T_S = \left\{ \sum_{\boldsymbol{\epsilon}\in\{0,1\}^m} s_1^{\epsilon_1} \dots s_m^{\epsilon_m} \sigma_{\boldsymbol{\epsilon}} \mid \sigma_{\boldsymbol{\epsilon}} \in \Sigma_T \right\},$$

where Σ_T is the cone of the SoS polynomials in $\mathbb{R}^T$. The polynomials in the preorder T_S are all non-negative on $\mathcal{K}$ but all non-negative polynomials on $\mathcal{K}$ are not necessarily in T_S. Krivine-Stengle's Positivstellensatz generalizes Artin's results by giving a description of all non-negative and positive polynomials on $\mathcal{K}$ as ratio of polynomials in T_S:

Theorem 1 (Krivine-Stengle's Positivstellensatz)
Let f be a polynomial on $\mathbb{R}^T$.

- *f is non-negative on $\mathcal{K}$ if and only if there exist g and h in T_S and k in $\mathbb{N}$ such that $gf = f^{2k} + h$.*
- *f is strictly positive on $\mathcal{K}$ if and only if there exist g and h in T_S such that $gf = 1 + h$.*

Note that the polynomial g plays the role of the denominator.

Furthermore, Schmüdgen observed that when $\mathcal{K}$ is compact, polynomials that are strictly positive on $\mathcal{K}$ always belong to T_S. He proved the following Positivstellensatz [Sch91] which is a denominator-free formulation of Krivine-Stengle's Positivstellensatz

Theorem 2 (Schmüdgen's Positivstellensatz)
If $\mathcal{K}$ is compact and a polynomial f is strictly positive on $\mathcal{K}$, then f belongs to T_S.

A stricter version of Schmüdgen's Positivstellensatz was given by Putinar [Put93]. Let us define the quadratic module associated to the collection S as

$$M_S = \Sigma_T + s_1\Sigma_T + \dots + s_J\Sigma_T\,.$$

Notice that M_S is a subset of the preorder T_S

$$T_S = \underbrace{\Sigma_T + s_1\Sigma_T + \dots + s_m\Sigma_T}_{= M_S} + s_1s_2\Sigma_T + \dots + s_1\dots s_J\Sigma_T\,.$$

The module M_S is said to be Archimedean if for any polynomial f in $\mathbb{R}[\mathbf{x}]$, there exists a natural integer N such that $N \pm f$ belongs to M_S, or equivalently such that $N - \sum_{t=1}^{T} x_t^2$ belongs to M_S. Putinar's Positivstellensatz can then be stated as follows

Theorem 3 (Putinar's Positivstellensatz)
If M_S is Archimedean and a polynomial f is strictly positive on $\mathcal{K}$, then f belongs to M_S.

Note that a similar Archimedean property can be defined for the preorder T_S and, in this case, this property is equivalent to the compactness of $\mathcal{K}$ as proved by Schmüdgen [Sch91]. However, although the Archimedean property of M_S implies the compactness of the set $\mathcal{K}$, the reverse does not always hold [Sch08].

The main advantage of Putinar's Positivstellensatz is that it needs only $J + 1$ SoS polynomials compared to the 2^J required by Schmüdgen's. Finally, notice that both Schmüdgen's and Putinar's Positivstellensätze certify strict positivity and they usually do not hold for non-negative polynomials.

The three Positivstellensätze presented here have therefore been used to relax the positivity constraint on $p - \gamma q$ in Problem (2.3). They form the basis for today's state-of-the-art real algebraic methods to solve rational problem. Several other certificates of positivity have been developed, for example by Reznick [Rez95] and Polyá [Pól28], and they lie at the heart of the methods solving polynomial and rational optimization problem. An overview of such methods is given in the next section.

§ 2.4 Overview on real algebraic methods

Problem (2.1) being NP-hard, different relaxations have been proposed based on Positivstellensätze. We present here some of the most prominent techniques. More details about the dependencies and connections of the four hierarchies exposed in this section can be found in the recent work of Kurpisz and De Wolff [KdW19].

2.4.1 Hierarchy of LP relaxations

Sherali and Adams proposed the relaxation and linearization technique [SA99] that aims to solve several optimization problems by using Linear Programming (LP) relaxation. This framework was one of the first method to solve polynomial optimization problems of the form (2.1) when q is the constant polynomial equal to 1. The main idea is to lift Problem (2.1) by introducing a new optimization variable for each nonlinear monomial appearing in the polynomials. The constraint linking those variables to the monomials are neglected and therefore yield an LP problem. Moreover, the original optimization variable $\mathbf{x}$ is assumed to be in a bounded box. A branch and bound algorithm is finally used, where an LP problem is solved at each iteration and the size of the bounded box is then reduced.

A more general framework to relax Problem (2.1) into an LP problem is based on the following Positivstellensatz from Krivine [Lau08]

Theorem 4
If $\mathcal{K}$ is compact, the polynomials $(s_j)_{j\in[\![j,J]\!]}$ have values between 0 and 1, and together with the constant polynomial 1, they generate the algebra $\mathbb{R}[\mathbf{x}]$, and a polynomial f is positive on $\mathcal{K}$, then f can be written as

$$f = \sum_{(\boldsymbol{\alpha},\boldsymbol{\beta})\in\mathbb{N}^J\times\mathbb{N}^J} \lambda_{\boldsymbol{\alpha}\boldsymbol{\beta}} \prod_{j=1}^{J} s_j^{\alpha_j} \prod_{j=1}^{J} (1-s_j)^{\beta_j}$$

for finitely many non-negative scalar coefficients $\lambda_{\boldsymbol{\alpha}\boldsymbol{\beta}}$.

Hence fixing a maximal degree on the multi-indices $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ to d, we can use Theorem 4 to replace the constraint $p - \gamma q$ of Problem (2.3) and we obtain a hierarchy of LP problems

$$\begin{cases} \gamma_d = \underset{\gamma,\boldsymbol{\lambda}}{\text{maximize}} & \gamma \\ \text{subject to} & p - \gamma q = \displaystyle\sum_{(\boldsymbol{\alpha},\boldsymbol{\beta})\in\mathbb{N}^J\times\mathbb{N}^J} \lambda_{\boldsymbol{\alpha}\boldsymbol{\beta}} \prod_{j=1}^{J} s_j^{\alpha_j} \prod_{j=1}^{J} (1-s_j)^{\beta_j}. \end{cases} \tag{2.5}$$

Problem (2.5) is indeed an LP problem since the number of non-zero coefficients $\lambda_{\boldsymbol{\alpha}\boldsymbol{\beta}}$ is finite and thus there is a finite number of linear constraints. It has been proved [Las09] that the sequence $(\gamma_d)_{d\in\mathbb{N}}$ is monotone, non-increasing, and that it converges to the optimal value γ_* of Problem (2.3). The relaxation and linearization technique is a special

case of the formulation (2.5). Since current LP solvers can solve numerically high-dimensional problems with millions of variables and constraints, this method scales well. However, a main drawback of LP hierarchies is their convergence which is only asymptotic and not finite as proved by Lasserre [Las09].

2.4.2 Sum-of-Squares and moment problem

Using a Positivstellensatz from Section 2.3, we can transform Problem (2.3) into a SoS problem. For instance, assuming that M_S is Archimedean, we can use Theorem 3 to replace the constraint in Problem (2.3) by the constraint [JdK05]

$$p - \gamma q \in M_S \,. \tag{2.6}$$

Furthermore, any decomposition into SoS can be characterized using a semi-definite matrix as follows [PW98].

Lemma 1 (Semi-definite characterization of SoS)
A polynomial s of degree $2d$ is a SoS if and only if there exist a semi-definite matrix $\mathbf{X}$ such that $s = [\mathbf{x}]_d^\top \mathbf{X} [\mathbf{x}]_d$, where $[\mathbf{x}]_d$ is the vector of all monomials up to degree d arranged according to the graded lexicographic order.

Remark that, for clarity, the basis of monomials is used here but for numerical computation other choices of basis may be more suitable [LP04].

Constraint (2.6) reads

$$p - \gamma q = r_0 + \sum_{j=1}^{J} s_j r_j \,, \tag{2.7}$$

where $(r_j)_{j \in [\![0,J]\!]}$ is a collection of SoS polynomials. Using Lemma 1 and fixing the maximal degree of the polynomials in the right hand side of (2.7) to $2k$, we obtain at most $\binom{T+2k}{2k}$ linear equality constraints between the coefficients together with the $J+1$ semi-definite constraint on the matrices $(V_j)_{j \in [\![0,J]\!]}$ corresponding to each s_i. The problem is now in the form of an SDP problem for each fixed degree $2k$. We remark that V_0 is a matrix of dimension $\binom{T+k}{k} \times \binom{T+k}{k}$ while the other matrices V_j has lower dimension $\binom{T+k-\mathrm{d}_{s_j}}{k-\mathrm{d}_{s_j}} \times \binom{T+k-\mathrm{d}_{s_j}}{k-\mathrm{d}_{s_j}}$ since the overall degree is $2k$. Solving the SDP problem for the different order k yields an increasing sequence of lower bounds on γ_* which converges to the latter [Par03]. This hierarchy of SDP convex relaxation is often called the SoS hierarchy.

However, this result was already proved by Lasserre in the dual perspective of the K-moment problem [Las01] where the derived SDP problems are exactly the dual of the ones in the SoS approach. For this reason, the hierarchy is also often named Lasserre's hierarchy in the literature. The moment approach lifts the original polynomial problem into the space of probability measures on $\mathcal{K}$ where we minimize over a measure. Measures are then replaced by their moments at the cost of additional semi-definite constraints that insure that the sequence of moments represents a measure of probability supported on $\mathcal{K}$. The approach of Lasserre is explained in detail in Chapter 3. The proof of the convergence of the hierarchy relies heavily on Putinar's Positivstellensatz, which emphasises the importance of those certificates of positivity.

This application of the Positivstellensätze illustrates the main advantage of Putinar's Positivstellensatz over Schmüdgen's. Indeed the number of semi-definite constraints is here decreased from 2^J to $J+1$ which is crucial to perform the computation in the SDP solvers.

One of the main benefit of SDP hierarchy over the previous LP hierarchy is its finite convergence. Indeed, the convergence of SDP hierarchy occurs at a finite order generically [Nie13]. Generic finite convergence means that for any instance of Problem (2.1) where the coefficients of the polynomials are drawn from an absolutely continuous probability distribution, there exists almost surely a finite relaxation order for which the optimal value of the SDP relaxation is equal to $\mathcal{J}^*$.

SoS hierarchy and its dual approach, Lasserre's hierarchy, are very efficient for low-scale problems. However, the dimensions of the SDP relaxations are growing quickly with the order k and the number of variables n and limit the practical application of the method. Sparse versions of the Positivstellensätze have been developed [GNS07] to transfer the sparsity of the polynomials involved in Problem (2.1) to its SDP relaxations and consequently lighten the computational burden of the latter.

2.4.3 Scaled diagonally dominant Sum-of-Squares

To tackle the computational complexity of Lasserre's hierarchy, several representations stepping aside SoS certificates have been proposed. A first scheme was proposed by Ahmadi et al. [AM19, AH19] and relies on a more limited Positivstellensatz.

Putinar's and Schmüdgen Positivstellensätze can be seen as approximations to the cone of non-negative polynomials by the cone of SoS polynomials. For a fixed degree, Blekherman gave a bound on the ratio of the volume of both cones and showed that the ratio of SoS polynomials among the non-negative ones decreases with the dimension T [Ble06]. However, they are numerous non-negative polynomials that can be expressed as SoS of higher degree polynomials. In fact, Lasserre even showed that non-negative polynomials can be approximated by SoS polynomials coefficient-wide when the degrees of the approximating polynomials goes to infinity. From this point, Ahmadi et al. proposed to approach the cone of non-negative polynomials with the cone of Diagonally-dominant-Sum-of-Squares (DSoS) polynomials which has the advantage to be characterized by Seconde-Order Cone Programming (SOCP) instead of SDP problems [AM19]. Indeed, current SOCP solvers scale better than SDP solvers and changing the approximating cone allows to solve some rational problems of higher dimensions.

Diagonally dominant matrices are defined as matrices whose diagonal terms are greater than the sum of the absolute value of all the elements on their corresponding row. Similarly, a matrix $\mathbf{A}$ is said to be scaled diagonally dominant if there exists a diagonal matrix $\mathbf{D}$ with positive coefficients such that $\mathbf{DAD}$ is diagonally dominant. In the spirit of Lemma 1, Ahmadi et al. define a polynomial p to be a DSoS (respectively Scaled-Diagonally-dominant-Sum-of-Squares (SDSoS)) polynomial if it can be expressed under the form $[\mathbf{x}]_d^\top \mathbf{X}[\mathbf{x}]_d$ where $\mathbf{X}$ is a diagonally dominant matrix (respectively scaled-diagonally matrix) [AH19]. DSoS and SDSoS polynomials are always non-negative [AM19], therefore a decomposition into such polynomials is a certificate of positivity. The corresponding Positivstellensatz called optimization-free is derived by the authors in [AH19]. Its use with DSoS or SDSoS polynomials leads respectively to LP for DSoS polynomials and SOCP problems [AH17, AM19] which have a lighter computational complexity.

The drawback of this approach is that the expressiveness of SDSoS polynomials is more limited than the one of SoS polynomials. Indeed, SDSoS polynomials can be seen as binomial squares and thus are SoS. Conversely, SoS polynomials are not necessary SDSoS.

2.4.4 Sum of non-negative circuit polynomials

The Sum-of-Non-negative Circuit (SONC) polynomials approach is similar to the SoS approach except that polynomials are now decomposed into non-negative circuit polynomials. Circuit polynomials were introduced by Iliman and De Wolff [IdW16] and are polynomials composed of $T+2$ monomials whose multi-indices form a circuit, i.e. an affinely dependent subset of $\mathbb{N}^T$ whose any proper subset is affinely independent. Most polynomials used in science and engineering are composed only of a few monomials. This property allows to reduce the complexity when the dimension of the variable or the degree of the involved polynomials are high. Inherently, circuit polynomials are composed of a few monomials and thus they can transfer the structure of Problem (2.1) into the hierarchy of relaxations. In contrast, in Lasserre's and the SoS hierarchies, we consider all the monomials up to a given degree, which is colossal for a high degree or a high number of variables.

Furthermore, the non-negativity of a circuit polynomial is easily verified by computing its circuit number and deciding whether a polynomial belongs to the SONC cone is equivalent to solving a Relative Entropy Programming (REP) problem [IdW16]. Similarly to LP, SOCP, and SDP, REP is a conic optimization problem where a linear cost function is minimized under linear constraints and conic constraint. The cone here is defined by a relative entropy and have the form $\{\ (\boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{\delta}) \in \mathbb{R}_+^T \times \mathbb{R}_+^T \times \mathbb{R}^T \mid \mu_t \log\left(\frac{\mu_t}{\nu_t}\right) \leq \delta_t\ \}$. Note that REP problems are convex and thus can be efficiently solved.

In the spirit of Schmüdgen and Putinar, a Positivstellensatz [DIdW17], has been proved for SONC polynomials.

Theorem 5 (Positivstellensatz for SONC polynomials)
If $\mathcal{K}$ is compact and a polynomial f is strictly positive on $\mathcal{K}$, then f is a sum of products of the polynomials $(s_j)_{j\in[\![1,J]\!]}$ and SONC polynomials.

Using this certificate in Problem (2.3) and fixing the degrees of the SONC polynomials yields a hierarchy of REP problems whose optimal values form a non-decreasing sequence that converges to the optimal value γ_* of the original problem [DIdW17]. Note that the SONC and SoS cones are not contained in each other. Thereby, some polynomials are SONC but not SoS and vice versa. For instance, Motzkin polynomial given in Equation (2.4) is SONC but not SoS. However, a polynomial can belong to both cones.

Although promising, the SONC approach is very recent and the theory is not yet mature enough for applications. Hence, there exists currently no numerical solver able to solve problems in the form of (2.1).

§ 2.5 Summary

Polynomial and rational functions prevail in many scientific and engineering domains due to their great modelling flexibility. As a consequence, many optimization problems involving polynomials naturally appear. Although optimizing rational functions under polynomial constraints is in general an NP-hard problem, recent techniques based on certificate of positivity for polynomials have shown some successful results. In the next chapter, we focus on Lasserre's hierarchy to solve rational problems. The main challenge is the computational workload which we tackle by handling the structure of the problem with special care.

- Chapter 3 -

Rational optimization with Lasserre's hierarchy

This section is concerned with the detailed resolution of a rational optimization problem using Lasserre's hierarchy presented in a progressive manner. In Section 3.1 we present the method for a general rational optimization problem while in Section 3.2, we explicitly show how the structure of a sum of rational functions allows us to reduce the dimensions of the final convex relaxation.

§ 3.1 Minimizing a rational function

In this section, we consider the following generic problem:

$$\begin{aligned} \mathcal{J}^* = \underset{\mathbf{x}\in\mathbb{R}^T}{\text{minimize}} \quad & \frac{p(\mathbf{x})}{q(\mathbf{x})} \\ \text{subject to} \quad & \mathbf{x} \in \mathcal{K}\,, \end{aligned} \tag{3.1}$$

where p and q are polynomial functions on $\mathbb{R}^T$ and $\mathcal{K}$ is the basic semi-algebraic set defined below:

$$\mathcal{K} = \left\{ \mathbf{x} \in \mathbb{R}^T \mid (\forall j \in [\![1, J]\!]) \quad s_j(\mathbf{x}) \geq 0 \right\}. \tag{3.2}$$

3.1.1 Condition on the feasible set

As we often work with bounded variables in practical applications, we make the mild assumption that $\mathcal{K}$ contains T polynomial constraints of the form

$$(\forall t \in [\![1, T]\!]) \quad x_t^2 \leq B^2\,, \tag{3.3}$$

where B is a positive constant. Since $\mathcal{K}$ is a closed set in a finite dimensional space, the above boundedness condition ensures that $\mathcal{K}$ is a compact set. Moreover, this condition ensures that the quadratic module associated to $\mathcal{K}$ is Archimedean as explained in Section 2.3.3. To simplify the notation, we write those constraints into a vector form as

$$(\mathbf{B} - \mathbf{x}) \odot (\mathbf{x} + \mathbf{B}) \geq \mathbf{0}\,,$$

where $\mathbf{B}$ and $\mathbf{0}$ are the vectors composed solely of B and 0, respectively and the operator $\odot$ denotes the element-wise Hadamard product.

3.1.2 Reformulation as a moment problem

As shown in [Las09, Proposition 5.20], Problem (3.1) is equivalent to solve

$$\begin{aligned} \inf_{\mu \in \mathcal{M}_+(\mathcal{K})} & \int_{\mathcal{K}} p(\mathbf{x}) \mu(\mathrm{d}\mathbf{x}) \\ \text{s.t.} & \int_{\mathcal{K}} q(\mathbf{x}) \mu(\mathrm{d}\mathbf{x}) = 1 \,, \end{aligned} \tag{3.4}$$

where $\mathcal{M}_+(\mathcal{K})$ denotes the cone of positive finite measures supported on $\mathcal{K}$. The equivalence between Problems (3.1) and (3.4) relies on the possibility to link any optimal point $\mathbf{x}_*$ of (3.1) to a Dirac measure $\delta(\mathbf{x}_*)/q(\mathbf{x}_*)$ solution to (3.4). The main idea here is to embed the original problem in a higher dimensional space in order to linearize it. At first glance, (3.4) looks more intricate than (3.1) since we need to minimize over an infinite-dimensional cone of measures supported by $\mathcal{K}$ instead of minimizing on $\mathcal{K}$ itself. However, the objective function and the constraint are now both linear in the new optimized variable μ. Furthermore, by defining $\mathbf{x}^{\boldsymbol{\alpha}} = x_1^{\alpha_1} \dots x_T^{\alpha_T}$, notice that

$$\int_{\mathcal{K}} p(\mathbf{x}) \mu(\mathrm{d}\mathbf{x}) = \int_{\mathcal{K}} \sum_{\boldsymbol{\alpha} \in \mathbb{N}^T} p_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}} \mu(\mathrm{d}\mathbf{x}) = \sum_{\boldsymbol{\alpha} \in \mathbb{N}^T} p_{\boldsymbol{\alpha}} v_{\boldsymbol{\alpha}} \,, \tag{3.5}$$

where $v_{\boldsymbol{\alpha}} = \int_{\mathcal{K}} \mathbf{x}^{\boldsymbol{\alpha}} \mu(\mathrm{d}\mathbf{x})$ denotes the moment of order $\boldsymbol{\alpha}$ of the measure μ. For convenience, we will use infinite vectors to write sums such as the rightmost member of (3.5). We define the infinite vector $\widetilde{\mathbf{p}} = (p_{\boldsymbol{\alpha}})_{\boldsymbol{\alpha} \in \mathbb{N}^T}$ and the infinite moment vector $\widetilde{\mathbf{v}} = (v_{\boldsymbol{\alpha}})_{\boldsymbol{\alpha} \in \mathbb{N}^T}$. Because $\widetilde{\mathbf{p}}$ has a finite number of nonzero elements, the sum in (3.5) is well-defined and can be written $\widetilde{\mathbf{p}}^\top \widetilde{\mathbf{v}}$.

Since $\mathcal{K}$ is a compact set, the measure μ is uniquely defined by its moments and we reformulate Problem (3.4) as

$$\begin{aligned} \inf_{\widetilde{\mathbf{v}} \in \mathbb{R}^{\mathbb{N}^T}} \quad & \widetilde{\mathbf{p}}^\top \widetilde{\mathbf{v}} \\ \text{s.t.} \quad & \widetilde{\mathbf{q}}^\top \widetilde{\mathbf{v}} = 1 \\ & \widetilde{\mathbf{v}} \in \mathcal{D}(\mathcal{K}) \end{aligned} \,, \tag{3.6}$$

where $\widetilde{\mathbf{q}}$ is defined similarly to $\widetilde{\mathbf{p}}$ as the infinite vector extensions of $\mathbf{q}$ obtained by zero padding and $\mathcal{D}(\mathcal{K})$ is the cone of moments of positive measures supported on $\mathcal{K}$. Our objective now is to replace this difficult conic constraint by simpler constraints. We introduce two tools, respectively, the moment matrix $\mathbf{M}(\widetilde{\mathbf{v}})$ associated to the moment vector $\widetilde{\mathbf{v}}$ and the localizing matrix $\mathbf{M}^s(\widetilde{\mathbf{v}})$ associated to $\widetilde{\mathbf{v}}$ with respect to a given polynomial s. Those matrices are infinite-dimensional and are both defined through their entries as follows

$$\begin{aligned} \big(\forall (\boldsymbol{\alpha}, \boldsymbol{\beta}) \in \mathbb{N}^T \times \mathbb{N}^T\big) \quad & \mathbf{M}_{\boldsymbol{\alpha},\boldsymbol{\beta}}(\widetilde{\mathbf{v}}) = \widetilde{v}_{\boldsymbol{\alpha}+\boldsymbol{\beta}} \\ \big(\forall (\boldsymbol{\alpha}, \boldsymbol{\beta}) \in \mathbb{N}^T \times \mathbb{N}^T\big) \quad & \mathbf{M}^s_{\boldsymbol{\alpha},\boldsymbol{\beta}}(\widetilde{\mathbf{v}}) = \sum_{\boldsymbol{\gamma} \in \mathbb{N}^T} s_{\boldsymbol{\gamma}} \widetilde{v}_{\boldsymbol{\alpha}+\boldsymbol{\beta}+\boldsymbol{\gamma}} \,. \end{aligned}$$

We define such infinite-dimensional matrices to be positive semi-definite if all their finite-dimensional principal submatrices are positive semi-definite. Since $\mathcal{K}$ is compact, Putinar's theorem [Hen13, Proposition 3.1] states that $\widetilde{\mathbf{v}}$ has a representing measure in $\mathcal{M}_+(\mathcal{K})$ if and only if the corresponding moment matrix $\mathbf{M}(\widetilde{\mathbf{v}})$ and localizing matrices $(\mathbf{M}^{s_j}(\widetilde{\mathbf{v}}))_{j \in [\![1,J]\!]}$ are positive semi-definite. The positive semi-definiteness of the moment and localizing matrices guarantee respectively that $\widetilde{\mathbf{v}}$ represents a positive measure and ensures that its support is $\mathcal{K}$.

3.1.3 Converging hierarchy of SDP problems

To solve numerically Problem (3.6), we replace the conic constraint with semidefinite constraints and then truncate the moment vector $\widetilde{\mathbf{v}}$, as well as its associated moment and localizing matrices, up to a degree $2k$ for a given integer k. This yields a hierarchy of convex SDP problems, known as Lasserre's hierarchy [Las01]. For a given relaxation order k, we have to find

$$\begin{aligned} \mathcal{J}_k^* = \inf_{\mathbf{v}\in\mathbb{R}^m} \quad & \sum_{\boldsymbol{\alpha}\in\mathbb{N}^T_{2k}} p_{\boldsymbol{\alpha}} v_{\boldsymbol{\alpha}} \\ \text{s.t.} \quad & \sum_{\boldsymbol{\alpha}\in\mathbb{N}^T_{2k}} q_{\boldsymbol{\alpha}} v_{\boldsymbol{\alpha}} = 1 \\ & \mathbf{M}_k(\mathbf{v}) \in \mathbb{S}^{n_0}_+ \\ & (\forall j \in [\![1, J]\!]) \quad \mathbf{M}^{s_j}_{k-\mathrm{d}_{s_j}}(\mathbf{v}) \in \mathbb{S}^{n_j}_+ \,, \end{aligned} \tag{3.7}$$

where $(\mathrm{d}_{s_j})_{j\in[\![1,J]\!]}$ are defined in (1.1). The cardinality of $\mathbb{N}^T_{2k}$ is $\binom{T+2k}{2k}$ and thus $\mathbf{v}$ is a vector containing $m = \binom{T+2k}{2k}$ moments. Furthermore, the truncated moment matrix $\mathbf{M}_k(\mathbf{v})$ is the principal submatrix of the infinite-dimensional moment matrix $\mathbf{M}(\widetilde{\mathbf{v}})$ that has dimension $n_0 \times n_0$ with $n_0 = \binom{T+k}{k}$. Thereby, $\mathbf{M}_k(\mathbf{v})$ is indexed by $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ in $\mathbb{N}^T_k \times \mathbb{N}^T_k$ and contains all the moments up to degree $2k$. Similarly, the truncated localizing matrices are the submatrix of their infinite-dimensional counterparts that have the dimensions $n_j \times n_j$ with $n_j = \binom{T+k-\mathrm{d}_{s_j}}{k-\mathrm{d}_{s_j}}$.

Problem (3.7) is an SDP problem in its dual form with linear equality constraints. Indeed, aggregating the moment and localizing matrices into a single symmetric block diagonal matrix before separating it into a sum along the elements of $\mathbf{v}$, we obtain

$$\begin{aligned} \mathcal{J}_k^* = \underset{\mathbf{v}\in\mathbb{R}^m}{\text{minimize}} \quad & \mathbf{p}^\top \mathbf{v} \\ \text{s.t.} \quad & \mathbf{C} - \sum_{i=1}^m v_i \mathbf{A}_i \in \mathbb{S}^n_+ \\ & \mathbf{a} - \mathbf{G}^\top \mathbf{v} = \mathbf{0} \,, \end{aligned} \tag{3.8}$$

where $\mathbf{C}$ and $(\mathbf{A}_i)_{i\in[\![1,m]\!]}$ are symmetric matrices, $\mathbf{a}$ is a vector of $\mathbb{R}^\ell$, and $\mathbf{G}$ is a matrix of $\mathbb{R}^{m\times\ell}$. The dimension n is thus given by $n = \sum_{j=0}^J n_j$. Notice that, in this section, (3.7) has only one linear constraint thus $\ell = 1$, $\mathbf{a} = 1$, and $\mathbf{G} = \mathbf{q}$. However, in Section 3.2, more linear constraints will be involved, so that we prefer to employ this matrix-vector notation here. Chapter 6 provides a detailed discussion on SDP problems and how to solve them efficiently. After truncation, the semi-definite positivity condition on the moment and localizing matrices is still a necessary but a not sufficient condition for $\mathbf{v}$ to have a representing measure. Hence, solving each SDP problem yields a lower bound $\mathcal{J}_k^*$ on the optimal value $\mathcal{J}^*$. Furthermore, the higher the order k, the tighter the bound $\mathcal{J}_k^*$ but the higher also the dimensions of the SDP problem. In our context where the optimized variable $\mathbf{x}$ is bounded, $(\mathcal{J}_k^*)_{k\in\mathbb{N}}$ is an increasing convergent sequence whose limit is $\mathcal{J}^*$ [Las01]. Moreover, the hierarchy has finite convergence generically, i.e. for any instance of Problem (3.1) where the coefficients of the polynomials are drawn from an absolutely continuous probability distribution, there exists almost surely a finite relaxation order k for which the optimal value $\mathcal{J}_k^*$ is equal to the optimal value $\mathcal{J}^*$ [Nie13]. Finally, the exact global solutions of (3.1) can be extracted from the solution of an SDP problem (3.7) indexed by a relaxation order at which convergence has occurred through an algebraic method [HL05]. The extraction method is detailed in Appendix D.

Note that the relaxation order k should be chosen such that

$$k \geq \max \left\{ \mathrm{d}_p, \mathrm{d}_q, \max_{j \in [\![1,J]\!]} \mathrm{d}_{s_j} \right\} .$$

This is a necessary condition which ensures that $2k$ is greater than the maximum degree of p, q and all the $(s_j)_{j \in [\![1,J]\!]}$, and prevents truncation of the latter polynomials. There is no a priori known sufficient relaxation order to ensure the convergence of the hierarchy. However, once the SDP relaxation is solved, there exists a sufficient condition that guarantees the convergence. Namely, if the moment matrices $\mathbf{M}_k$ and $\mathbf{M}_{k-1}$ have the same rank, then convergence has occurred [Las09].

We remark that the dimensions n and m of the SDP problem grows respectively as T^k and T^{2k} when T is large, hence exponentially in the degree of the involved polynomials.

§ 3.2 Problem structure emerging from a sum of rational functions

Although constituting the theoretical foundation of our work, the approach presented in Section 3.1 is computationally inefficient for many practical problems and requires further improvements that we now explain. Indeed, in many encountered rational problems, the objective function is a sum of rational functions. Reducing a sum of rational functions using a common denominator often yields a rational function with very high degree, which then requires a high relaxation order k in the hierarchy of SDP problems. As a consequence, the obtained SDP problems are too high-dimensional to be solvable in a reasonable time using state-of-the-art solvers. However, a more ingenious method is to use the structure induced by the sum to yield a block SDP problem [CPM19]. There are two types of structure to consider in our problem: first we deal with a sum of rational functions instead of a single one, and then each of those functions depend on a small subset of variables only. The latter structure is sometimes referred to as sparse problems and sparse polynomials [WKKM06, BHL15]. However, in order to prevent confusion with the sparsity of the signals considered in our applications of Chapters 4 and 5, we will not use this terminology. To illustrate our methodology, let us turn our attention on finding a global minimizer $\mathbf{x}_*$ and the optimal value $\mathcal{J}^*$ such that

$$\mathcal{J}^* = \min_{\mathbf{x} \in \mathcal{K}} \sum_{i=1}^{\tilde{I}} \frac{p_i(\mathbf{x}_{E_i})}{q_i(\mathbf{x}_{E_i})} , \tag{3.9}$$

where p_i and q_i are polynomials in T_i variables and $\mathcal{K}$ is a compact subset of $\mathbb{R}^T$ having the form of (3.2). The vector $\mathbf{x}_{E_i}$ denotes the subvector of $\mathbf{x}$ composed of the elements indexed by the set E_i, the latter being a subset of $[\![1, T]\!]$ of cardinality T_i. We further assume that the polynomials in (3.9) involve only a few variables, i.e.

$$\left(\forall i \in [\![1, \tilde{I}]\!] \right) \quad T_i \ll T . \tag{3.10}$$

3.2.1 Exploiting the sum of rational functions structure

Instead of introducing a single measure on all the variables, we now introduce a measure μ_i for each rational function p_i/q_i of the sum. However, coupling between variables from different measures appears when the sets $(E_i)_{i \in [\![1,\tilde{I}]\!]}$ intersect. We therefore need to add moment equality constraints to ensure that the overlapping moments of two measures are identical. Moreover, we need some restrictions on how variables can appear in several

measures in order to keep the problem consistent. The required condition is that the sets $(E_i)_{i\in[\![1,\tilde{I}]\!]}$ verify the so-called running intersection property [Las09, BHL15] which is stated as

$$(\forall i \in [\![2,\tilde{I}]\!], \exists \tilde{\jmath} \in [\![1, i-1]\!]) \quad E_i \bigcap \left(\bigcup_{j=1}^{i-1} E_j \right) \subseteq E_{\tilde{\jmath}}\,.$$

Together with the compactness of $\mathcal{K}$, it guarantees that Problem (3.9) is equivalent to

$$\begin{aligned} &\inf_{\boldsymbol{\mu}\in\Xi} \sum_{i=1}^{\tilde{I}} \int_{\mathcal{K}_i} p_i(\mathbf{x}_{E_i}) \mu_i(\mathrm{d}\mathbf{x}_{E_i}) \\ &\text{s.t. } (\forall i \in [\![1,\tilde{I}]\!]) \int_{\mathcal{K}_i} q_i(\mathbf{x}_{E_i}) \mu_i(\mathrm{d}\mathbf{x}_{E_i}) = 1 \\ &\quad (\forall (i,j) \in [\![1,\tilde{I}]\!] \times [\![1,\tilde{I}]\!])(\forall \boldsymbol{\gamma} \in \mathbb{N}^{\mathrm{Card}(E_{i,j})}) \\ &\quad \int_{\mathcal{K}_i} q_i(\mathbf{x}_{E_i}) \mathbf{x}_{E_{i,j}}^{\boldsymbol{\gamma}} \mu_i(\mathrm{d}\mathbf{x}_{E_i}) = \int_{\mathcal{K}_j} q_j(\mathbf{x}_{E_j}) \mathbf{x}_{E_{i,j}}^{\boldsymbol{\gamma}} \mu_j(\mathrm{d}\mathbf{x}_{E_j})\,, \end{aligned} \tag{3.11}$$

where $E_{i,j} = E_i \bigcap E_j$ and $\boldsymbol{\mu} = (\mu_i)_{i\in[\![1,\tilde{I}]\!]}$ is the new optimization variable belonging to the product $\Xi = \bigtimes_{i\in[\![1,\tilde{I}]\!]} \mathcal{M}_+(\mathcal{K}_i)$. The sets $(\mathcal{K}_i)_{i\in[\![1,\tilde{I}]\!]}$ are subsets of $\mathbb{R}^{T_i}$ defined by the subsets of polynomials in variables $\mathbf{x}_{E_i}$ defining $\mathcal{K}$. The last equality constraints in (3.11) enforce equality between the marginal distributions of $q_j\mu_j$ and $q_i\mu_i$ along $\mathbf{x}_{E_{ij}}$. In other words, those constraints ensure the equality of overlapping moments between different measures.

3.2.2 Block structure in the SDP problems of the hierarchy

As in Section 3.1, we now use Putinar's theorem to replace each measure by its moment vector at the cost of additional semi-definite constraints. We then truncate the moment vectors as well as the moment and localizing matrices, before stacking them. As a result, the moment vector $\mathbf{v} = [\mathbf{v}_1^\top, \dots, \mathbf{v}_{\tilde{I}}^\top]^\top$ is a stack of the moment vectors of each measure μ_i. Similarly, the moment matrix $\mathbf{M}_k(\mathbf{v}) = \mathrm{Diag}\left(\mathbf{M}_{1,k}(\mathbf{v}_1), \dots, \mathbf{M}_{\tilde{I},k}(\mathbf{v}_{\tilde{I}})\right)$ and the localizing matrices $\mathbf{M}^{s_j}_{k-\mathrm{d}_{s_j}}(\mathbf{v}) = \mathrm{Diag}\left(\mathbf{M}^{s_j}_{1,k-\mathrm{d}_{s_j}}(\mathbf{v}_1), \dots, \mathbf{M}^{s_j}_{\tilde{I},k-\mathrm{d}_{s_j}}(\mathbf{v}_{\tilde{I}})\right)$ have a block diagonal structure where each diagonal block corresponds respectively to the moment or localizing matrix of one of the measures μ_i. This leads to the following SDP problem:

$$\begin{aligned} \mathcal{J}_k^* = \quad \inf_{\mathbf{v}\in\mathbb{R}^m} \quad & \mathbf{p}^\top \mathbf{v} \\ \text{s.t.} \quad & (\forall i \in [\![1,\tilde{I}]\!]) \quad \mathbf{q}_i{}^\top \mathbf{v}_i = 1 \\ & \mathbf{M}_k(\mathbf{v}) \in \mathbb{S}_+^{n_0} \\ & (\forall j \in [\![1,J]\!]) \quad \mathbf{M}^{s_j}_{k-\mathrm{d}_{s_i}}(\mathbf{v}) \in \mathbb{S}_+^{n_j} \\ & \mathbf{F}\mathbf{v} = \mathbf{0}\,, \end{aligned} \tag{3.12}$$

where

$$m = \sum_{i=1}^{I} \binom{T_i + 2k}{2k}\,,\ n_0 = \sum_{i=1}^{I} \binom{T_i + k}{k}\,,\ n_j = \sum_{i=1}^{I} \binom{T_i + k - \mathrm{d}_{s_j}}{k - \mathrm{d}_{s_j}}\,,$$
$$\mathbf{p} = [\mathbf{p}_1^\top, \dots, \mathbf{p}_{\tilde{I}}^\top]^\top\,,$$

and $\mathbf{F}$ is a matrix in $\mathbb{R}^{\ell\times m}$ representing the linear constraints linking the $(\mathbf{v}_i)_{i\in[\![1,\tilde{I}]\!]}$ together and coming from the constraints between the projections in (3.11). Similarly to Section 3.1, Problem (3.12) can be finally expressed in the canonical form (3.8).

There are two main differences with the situation discussed in Section 3.1:

- Instead of having a single measure on all the variables, we obtain several measures on different smaller subsets of variables. The SDP optimization variable $\mathbf{v}$ is now a vector built by stacking the different truncated moment vectors $(\mathbf{v}_i)_{i\in[\![1,\tilde{I}]\!]}$ of each measure. As a consequence, the moment and localizing matrices have a block diagonal structure, each block corresponding to a measure, or equivalently to a term in the sum of Problem (3.9). Thanks to Assumption (3.10), the size of the blocks in the moment matrix, equal to $\binom{T_i+2k}{2k}$, is much smaller than the size $\binom{T+2k}{2k}$ obtained in Section 3.1. Especially when k increases, the difference in size becomes even more significant. The block structure can then be efficiently exploited by SDP solvers to decrease the computational time.

- Extra moment constraints due to the coupling between variables arise. Although those constraints may be numerous, they are linear equality constraints in the SDP problem; their impact on the computational time of the SDP solver is minor.

3.2.3 Complexity of the SDP relaxations

The complexity of an SDP problem under the form (3.8) is expressed as a quadruple of integers $(n, m, m_{\mathrm{s}}, \ell)$. The integer m denotes the size of the vector of optimized variables, n is the size of the semi-definite inequality constraint, ℓ is the number of linear equality constraints, and m_{s} is the number of block matrices involved in the semi-definite constraint. Note that n is related to m_{s} since it is the sum of the size of each block. The above quadruple therefore does not fully characterize the structure of an SDP problem. For example, having one huge block of size 90×90 and nine tiny ones of size 10×10 is not equivalent in terms of complexity to having ten medium blocks of size 18×18. In both cases, n is equal to 180 while the semi-definite constraint in the SDP relaxation of order 3 has size 132340 in the first case and 13300 in the second. However, knowing n and m_{s} is usually enough to get a good evaluation of the complexity of the problem.

We give here an expression for n, m_{s}, and m as they are usually the main bottleneck for SDP solvers.

Number of blocks m_{s}
In order to solve Problem (3.9), we introduce $\tilde{I}$ measures. Moment and localizing matrices of each measure yield a block in the relaxed SDP problems. The number of localizing matrices for each measure is equal to the number of polynomial constraints defining the set $\mathcal{K}_i$ denoted η_i. We have at least T_i constraints for each set $\mathcal{K}_i$ due to the bound condition (3.3) and at most J. The final number of blocks in the matrices of the SDP relaxation is thus

$$m_{\mathrm{s}} = \sum_{i=1}^{\tilde{I}} (1 + \eta_i)\,. \tag{3.13}$$

Dimension of the global moment vector m
The dimension m of the vector $\mathbf{v}$ is the sum of the dimension of the moment vectors for each measure $(\mu_i)_{i\in[\![1,\tilde{I}]\!]}$, i.e.

$$m = \sum_{i=1}^{\tilde{I}} \binom{T_i + 2k}{2k}\,. \tag{3.14}$$

Dimension of the semi-definite constraint n
The dimension n of the conic constraint is the sum of the dimensions of all the moment and localizing matrices. The moment matrices have a size of $\binom{T_i+2k}{2k}$ whereas the localising matrices have size $\binom{T_i+k-\mathrm{d}_{s_j}}{k-\mathrm{d}_{s_j}}$, which gives

$$n = \sum_{i=1}^{\tilde{I}} \left(\binom{T_i + 2k}{2k} + \sum_{j=1}^{\eta_i} \binom{T_i + k - \mathrm{d}_{s_j}}{k - \mathrm{d}_{s_j}} \right) . \tag{3.15}$$

3.2.4 Coupling and linear equality constraints

As noted, introducing several measures numerous extra linear moment constraints due to the coupling between the variables of the different measures. However, among those extra constraints, we remark that many of them are actually redundant. Let us take a simple example to illustrate this fact.

Example Assume that we want to minimize over the variable $\mathbf{x} = (x_t)_{t\in[\![1,5]\!]}$, a sum of three rational functions which has the following form

$$\frac{p_1(x_1,x_2,x_3)}{q_1(x_1,x_2,x_3)} + \frac{p_2(x_2,x_3,x_4)}{q_2(x_2,x_3,x_4)} + \frac{p_3(x_3,x_4,x_5)}{q_3(x_3,x_4,x_5)} .$$

Following the method developed in Section 3.2.1, we introduce three measures μ_1, μ_2 and μ_3, one for each term of the sum. We thus need the following equality constraints between moments, for every (α,β) in $\mathbb{N}^2$,

$$\begin{aligned}
\int q_1(x_1,x_2,x_3)x_2^\alpha x_3^\beta \mu_1(\mathrm{d}x_1,\mathrm{d}x_2,\mathrm{d}x_3) &= \int q_2(x_2,x_3,x_4)x_2^\alpha x_3^\beta \mu_2(\mathrm{d}x_2,\mathrm{d}x_3,\mathrm{d}x_4) \\
\int q_2(x_2,x_3,x_4)x_3^\alpha x_4^\beta \mu_2(\mathrm{d}x_2,\mathrm{d}x_3,\mathrm{d}x_4) &= \int q_3(x_3,x_4,x_5)x_3^\alpha x_4^\beta \mu_3(\mathrm{d}x_3,\mathrm{d}x_4,\mathrm{d}x_5) \\
\int q_1(x_1,x_2,x_3)x_3^\alpha \mu_1(\mathrm{d}x_1,\mathrm{d}x_2,\mathrm{d}x_3) &= \int q_3(x_3,x_4,x_5)x_3^\alpha \mu_3(\mathrm{d}x_3,\mathrm{d}x_4,\mathrm{d}x_5) .
\end{aligned}$$

We observe that the variable x_3 appears in each term of the sum and thus also in moments of each measure. In particular, we notice that the last constraint is redundant with the first two ones when $\beta = 0$. It is thus sufficient to consider only moment equality constraints on consecutive measures $(\mu_i)_{u\in[\![1,3]\!]}$.

3.2.5 Linear versus quadratic polynomial constraints

The methodology explained in this chapter is valid if only if the constraint set $\mathcal{K}$ is compact. We therefore assume bound conditions (3.3) in the definition of $\mathcal{K}$ Those bound constraints can be expressed in two ways:

- first, as two linear vector constraints

$$\begin{aligned}
\mathbf{B} - \mathbf{x} &\geq \mathbf{0} \\
\mathbf{x} + \mathbf{B} &\geq \mathbf{0} ,
\end{aligned}$$

- or as a single quadratic vector constraint

$$(\mathbf{B} - \mathbf{x}) \odot (\mathbf{x} + \mathbf{B}) \geq \mathbf{0} .$$

Following Section 3.2.3, using two linear inequality constraints per variable introduces $2\sum_{i=1}^{\tilde{I}} T_i$ localizing matrices and consequently as many blocks in our SDP problems while using a quadratic inequality constraint only adds $\sum_{i=1}^{\tilde{I}} T_i$ blocks. Moreover, linear and quadratic constraints yield blocks of identical size. Indeed, the size of a localizing matrix $\mathbf{M}_k^s$ corresponding to a polynomial s in ω variables is given by $\binom{\omega+k-\mathrm{d}_s}{k-\mathrm{d}_s}$ and here $\mathrm{d}_s = 1$ for both linear and quadratic constraints. Therefore, formulating the bound constraints as quadratic constraints reduces by a factor two the number of blocks associated to such bounds.

§ 3.3 Summary

This chapter forms the backbone of the methodology developed in Chapters 4 and 5. We have recalled the basic framework of Lasserre's hierarchy to solve rational optimization problems before specializing it to leverage a particular structure of the criterion. We have then studied the complexity of the obtained SDP relaxations in order to understand the impact of the different parameters. In the next chapters where we apply the framework to signal processing problems, the latter study will be essential to obtain problems that are computationally feasible.

- Chapter 4 -

Sparse signal reconstruction for nonlinear models

§ 4.1 Motivation

Sparse signals, i.e. signals composed of a few spikes, are of particular interest. They either occur naturally in many areas or emerge after sparsifying transformations such as time-frequency or wavelet decompositions [GDP09, PDCP14]. However, accurate data acquisition of sparse signals from real-world measurements remains an open challenge. The difficulty of the problem is further increased when acquiring data at a reduced rate. This is however an important practical situation, since it permits faster acquisitions for high-throughput experiments and analysis.

A common approach to recover the original signal from the observations is first to define a well-chosen criterion and then to minimize it. The criterion is often composed of two terms: a fit function depending on the investigated model as well as the observations, and a (possibly composite) regularization term that allows good estimates to be selected among those consistent with the data [CCPW07]. However, few methods today are able to deal with nonlinear models and to globally optimize sparsity promoting criteria. Indeed, integrating any of these two properties in the criterion often yields an intricate optimization problem that is difficult to solve due to nonconvexity.

Thence, to deal with nonlinear effects, linearization techniques are often used since the vast majority of available methods only apply to linear models [Tib96, BD08, SBFA15] or to models with weaker linearity assumptions [Sch10, DTR+14, DD15]. On the other hand, the standard approach to promote sparse solutions consists in adding an ℓ_0 penalization to a data-fit cost function which leads to NP hard optimization problems [Nik13, BNCM16]. Consequently, several surrogates to the ℓ_0 penalization have been suggested, the simplest one being the ℓ_1 norm. The latter has the enjoyable property of being convex, which simplifies the optimization task [CP08, CP11], but it also strongly penalizes high values of the variables and thus introduces a bias in the solutions. Albeit providing good results, the nonconvex Geman-McClure function [CP15] also tends to introduce bias. Therefore further relaxations of ℓ_0 function have been investigated [FL01]. A major drawback is that those relaxations are nonconvex and result in optimization problems which are difficult to solve globally in the sense that currently available algorithms only converge to local solutions and therefore may be highly dependent on their initialization [FL01, ODBP15, CWB08, BH11, PN15, BD08, Sel17].

In the case of a linear model, a first approach for ensuring global convergence of an exact relaxation of the ℓ_0 function has been proposed in [BNCM16] and is based on mixed-integer programming. Our work proposes a different approach grounded on the global minimization of the broad class of piecewise rational functions under polynomial

constraints. Based on it, we propose a novel recovery method for sparse signals from subsampled observations obtained through a noisy model involving nonlinear functions. More precisely, we show that the fit function and the regularization term can be modelled as piecewise rational functions. Fortunately, many well-known good approximations to the ℓ_0 penalization satisfy the latter property [Zha10b, FL01, Zha10a, AFS13, JTVW11, SBFA15]. Moreover, various nonlinear degradations, such as saturation, can be modelled with rational functions. Hence, several criteria of interest for reconstructing sparse signals which have been nonlinearly degraded can be modelled, or faithfully approximated, as piecewise rational. We then reformulate the corresponding piecewise rational optimization problem as the minimization of a sum of rational functions, for which the methodology of Chapter 3 can be applied. As SDP problems are playing an important role in our methodology, we study the overall complexity of the SDP relaxations and show how to reduce it efficiently in several ways. We especially emphasize the advantageous effect of subsampling.

The chapter is organized as follows: after introducing our model and criterion in Section 4.2, we present in Section 4.3 the class of approximations to the ℓ_0 penalty we consider and we reformulate the minimization of our criterion as a rational optimization problem. Section 4.4 details how to solve such optimization problems by leveraging its inherent structure. Section 4.5 studies the complexity of the obtained SDP problems before explaining how to decrease it efficiently. Section 4.6 presents numerical simulations in order to validate our method.

§ 4.2 Observation and signal model

4.2.1 Our observation model

We consider the reconstruction of an unknown discrete-time sparse signal $\overline{\mathbf{x}}$ of length T. The measurement process deteriorates $\overline{\mathbf{x}}$ in the following way: the peaks it contains are enlarged and the sensors introduce a saturation effect. In the literature, these degradations are commonly modelled respectively by a convolution with a finite impulse response filter and by a memoryless nonlinear function Φ. The filter coefficients are given by a vector $\mathbf{h}$ of length L. Finally, a noise is superimposed, which is modelled by an additive vector term $\mathbf{w}$ with samples drawn from an i.i.d. zero-mean Gaussian distribution. Different noise models will be discussed in Chapter 5.

An important feature of our model is its ability to deal with a subsampling of the measured signal during the acquisition, which is an interesting property as in many applications such as spectroscopy, the physical limitations may allow only subsampled data acquisition. We thereby introduce a decimation operator D. Interestingly, we will see that our approach is applicable in this context and allows one to use well-suited penalization terms to promote sparsity. Defining the observation vector $\mathbf{y}$ of size U after subsampling and the convolution operator $*$, the equation corresponding to our model finally reads

$$\mathbf{y} = D\big(\Phi(\mathbf{h} * \overline{\mathbf{x}}) + \mathbf{w}\big)\,. \tag{4.1}$$

For instance, Model (4.1) can emulate narrow-peak signals from gas chromatography experiments [VRGB+05, VRGB+07]. In this case, the filter $\mathbf{h}$ has a discretized Gaussian shape. This choice arises from traditional stochastic or plate modeling, representing a Galton-Hennequin bell distribution [Fel98, Chapter 3]. Peak saturation is also modelled, which cannot be done in standard analytical chemistry practice. According to [KKS18], filter lengths L from 3 to 9 samples may suffice for a relatively accurate estimation of the peak area, a quantity related to the concentration of a particular molecule.

We will be interested in regular decimation patterns D_δ where all the elements indexed by a multiple of an integer δ are deleted, namely

$$D_\delta\big((s_t)_{t\in[\![1,T]\!]}\big) = (s_{\Delta(u,\delta)})_{u\in[\![1,U]\!]}\,, \tag{4.2}$$

where

$$U = T - \lfloor T/\delta \rfloor \tag{4.3}$$

and Δ is defined as

$$(\forall u \in [\![1,U]\!]) \quad \Delta(u,\delta) = u + \left\lfloor \frac{u-1}{\delta-1} \right\rfloor . \tag{4.4}$$

We denote by D_∞ the identity operator that preserves the entire signal. Let us illustrate the two decimation patterns D_2 and D_4 on the example vector $\mathbf{s} = [s_1, s_2, \ldots, s_7, s_8]^\top$

$$\begin{aligned}
\mathbf{s} &\overset{D_2}{\longmapsto} [s_1, s_3, s_5, s_7]^\top = (s_{\Delta(u,2)})_{u\in[\![1,4]\!]} \\
\mathbf{s} &\overset{D_4}{\longmapsto} [s_1, s_2, s_3, s_5, s_6, s_7]^\top = (s_{\Delta(u,4)})_{u\in[\![1,6]\!]}\,.
\end{aligned}$$

The smaller parameter δ, the higher the decimation and harder the reconstruction of the signal $\overline{\mathbf{x}}$.

To estimate the original signal $\overline{\mathbf{x}}$, we minimize a penalized criterion $\mathcal{J}$ composed of two terms:

$$(\forall \mathbf{x} \in \mathbb{R}^T) \quad \mathcal{J}(\mathbf{x}) = f_{\mathbf{y}}(\mathbf{x}) + \mathcal{R}_\lambda(\mathbf{x})\,. \tag{4.5}$$

The first one $f_{\mathbf{y}}$ is a fit measure with respect to the acquired measurements $\mathbf{y}$ while the second one $\mathcal{R}_\lambda$ is a regularization term which will be discussed in more detail in Section 4.2.2.

As a fit function, we choose the standard least-squares error between $\mathbf{y}$ and the output of the noiseless model for a given estimate $\mathbf{x}$ of the original signal $\overline{\mathbf{x}}$

$$\begin{aligned}
(\forall \mathbf{x} \in \mathbb{R}^T) \quad f_{\mathbf{y}}(\mathbf{x}) &= \|\mathbf{y} - D_\delta(\Phi(\mathbf{h} * \mathbf{x}))\|_2^2 \\
&= \|\mathbf{y} - D_\delta(\Phi(\mathbf{H}\mathbf{x}))\|_2^2\,,
\end{aligned}$$

where $\mathbf{H}$ is a Toeplitz band matrix corresponding to the convolution with $\mathbf{h}$ with convolution boundaries using zero padding. Because of the transformation Φ, the fit function $f_{\mathbf{y}}$ is possibly nonconvex. This is in contrast with more classical linear models in which the fit function reduces to the quadratic function $\mathbf{x} \longmapsto \|\mathbf{y} - D_\delta(\mathbf{H}\mathbf{x})\|_2^2$. In our approach, other fit functions $f_{\mathbf{y}}$ can be chosen to model different problems as long as they are rational. Examples of different fit functions are studied in Chapter 5 to reconstruct signals deteriorated by non-Gaussian noise. In the following, the nonlinear function Φ is assumed to be rational and to act component-wise. Note that it is not a restrictive assumption as any function can be tightly approximated with a suitable rational or piecewise rational function [DB08]. Setting the components of $\mathbf{x}$ with non-positive index to be identically zero in order to unclutter notation, $f_{\mathbf{y}}$ hence reads as a sum of rational functions

$$f_{\mathbf{y}}(\mathbf{x}) = \sum_{u=1}^{U} \underbrace{\left(y_u - \Phi\Big(\sum_{l=1}^{L} h_l x_{\Delta(u,\delta)-l+1} \Big) \right)^2}_{g_u(x_{\Delta(u,\delta)-L+1}, \ldots, x_{\Delta(u,\delta)})}\,,$$

where $(g_u)_{u\in[\![1,U]\!]}$ are rational functions in L variables.

4.2.2 Sparsity and examples of ℓ_0 approximations

The unknown original signal $\overline{\mathbf{x}}$ sought by the reconstruction method is assumed to be sparse. In other words, it comprises only few peaks and many of its components are zero. Following this assumption, the second term $\mathcal{R}_\lambda$ in (4.5) is a sparsity-promoting penalization weighted by a positive parameter λ. Ideally, we would like $\mathcal{R}_\lambda$ to be the sparsity measure $\lambda\ell_0$ (where ℓ_0 counts the number of nonzero elements) but, in order to derive computationally efficient optimization techniques, a suitable separable approximation is substituted for it, which reads

$$(\forall \mathbf{x} = (x_t)_{t\in[\![1,T]\!]} \in \mathbb{R}^T) \quad \mathcal{R}_\lambda(\mathbf{x}) = \sum_{t=1}^{T} \Psi_\lambda(x_t)\,. \tag{4.6}$$

Common approaches consist in using either convex functions Ψ_λ such as the ℓ_1 norm, or nonconvex ones that still maintain the convexity of the overall criterion [Sel17]. However, a good approximation $\Psi_\lambda : \mathbb{R} \to \mathbb{R}$ to the ℓ_0 function requires the following three properties [FL01] leading to nonconvex criteria: unbiasedness for large values, sparsity to reduce the complexity of the model by setting small values to zero, and continuity to ensure the stability of the model. In previous works [CP15, CP17, CPM19], the nonconvex Geman-McClure function was used as an approximation to the ℓ_0 function but it introduced bias in the estimate. Here, in contrast, we propose a much wider class of piecewise rational function approximations that satisfy the three mentioned properties. Those approximations extend significantly our previous work to settings of more practical interest.

Several examples of functions Ψ_λ shown in the literature to yield good approximations to the ℓ_0 function are actually piecewise rational functions, for which we will show in this chapter that exact minimization is achievable. We list below examples of the most commonly used piecewise rational approximations to the ℓ_0 penalization that appear in several areas such as imaging or statistics. Figure 4.1 displays the graph of those functions on $[-3, 3]$.

- Capped ℓ_p [Zha10b, AFS13, JTVW11]:

$$\Psi_\lambda(x) = |x|^p \mathbb{1}_{\{|x|\leq\lambda\}} + \lambda^p \mathbb{1}_{\{|x|>\lambda\}}\,.$$

- Smoothly clipped absolute deviation (SCAD) [FL01]: $(\gamma \in]2, +\infty[)$

$$\Psi_\lambda(x) = \lambda\,|x|\,\mathbb{1}_{\{|x|\leq\lambda\}} + \frac{(\gamma+1)\lambda^2}{2}\mathbb{1}_{\{|x|>\gamma\lambda\}} - \frac{\lambda^2 - 2\gamma\lambda\,|x| + x^2}{2(\gamma-1)}\mathbb{1}_{\{\lambda<|x|\leq\gamma\lambda\}}\,,$$

- Minimax concave penalty (MCP) [Zha10a]: $(\gamma \in \mathbb{R}_+^*)$

$$\Psi_\lambda(x) = \left(\lambda\,|x| - \frac{x^2}{2\gamma}\right)\mathbb{1}_{\{|x|\leq\gamma\lambda\}} + \frac{\gamma\lambda^2}{2}\mathbb{1}_{\{|x|>\gamma\lambda\}}\,,$$

- Continuous exact ℓ_0 (CEL0) [SBFA15]: $(\gamma \in \mathbb{R}_+^*)$

$$\Psi_\lambda(x) = \lambda - \frac{\gamma^2}{2}\left(|x| - \frac{\sqrt{2\lambda}}{\gamma}\right)^2 \mathbb{1}_{\left\{|x|\leq\frac{\sqrt{2\lambda}}{\gamma}\right\}}\,.$$

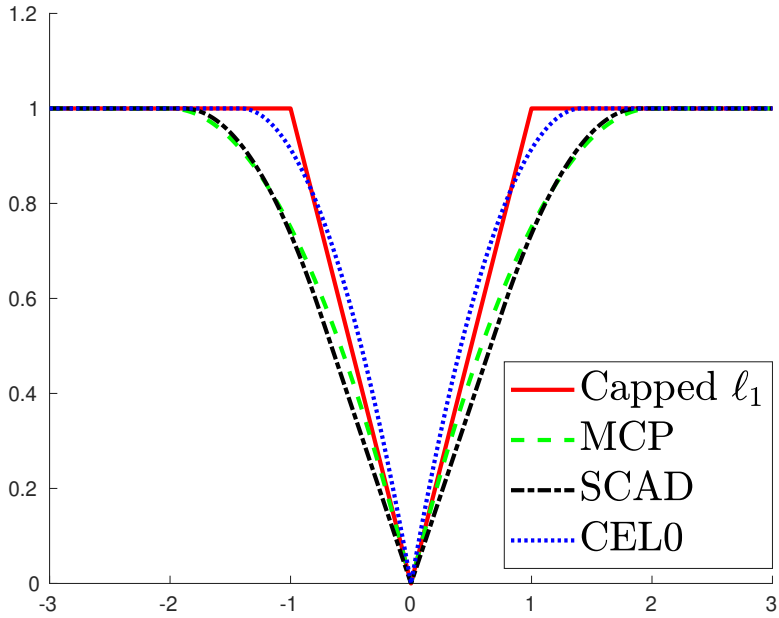


Figure 4.1: *Examples of continuous relaxation of ℓ_0 penalization ($\lambda = 1$, $\gamma_{\text{SCAD}} = 2.5$, $\gamma_{\text{MCP}} = 2$, $\gamma_{\text{CEL0}} = 1$).*

Although CEL0 and MCP share a similar expression for the function Ψ_λ, they are quite different in their overall form $\mathcal{R}_\lambda$ due to the choice of the parameter γ. In (4.6), this parameter for MCP is fixed for all the samples x_t, while for CEL0, its value is adapted to each sample. In the above penalization, the lower the parameter γ, the tighter the approximation to the ℓ_0 penalization but the stronger also the nonconvexity. An important remark concerning the above examples is that, when Φ is set to the identity, a suitable choice of the parameter γ guarantees that the global minimizers of the criterion $f_{\mathbf{y}} + \mathcal{R}_\lambda$ are exactly the global minimizers of the criterion $f_{\mathbf{y}} + \lambda\ell_0$ [SBFA17]. The choice of γ depends on the parameter λ and the norm of the columns of $D_\delta\mathbf{H}$. This behavior provides important insights and guarantees on the quality of the above functions as penalization terms to enforce sparsity of the solutions.

§ 4.3 Rational formulation of the problem

4.3.1 Exact polynomial reformulation of the ℓ_0 function

Let us remind that the signal reconstruction problem is tackled through the minimization of criterion $\mathcal{J}$ which has been defined in (4.5). We thus want to find

$$\mathcal{J}^* = \min_{\mathbf{x}\in\mathbb{R}^T} \quad \mathcal{J}(\mathbf{x})\,. \tag{4.7}$$

We emphasize that formulating our problem as a polynomial/rational one allows us to apply the framework developed in Chapter 3. First, let us show that there exists a polynomial reformulation of the ℓ_0 criterion. Indeed, choosing $\mathcal{R}_\lambda = \lambda\ell_0$ in the penalization term of criterion (4.5), the original problem (4.7) can be reformulated by using a rational function and polynomial constraints, as follows:

$$\begin{aligned} \underset{(\mathbf{x},\boldsymbol{\xi})\in\mathbb{R}^T\times\mathbb{R}^T}{\text{minimize}} \quad & \|\mathbf{y} - D_\delta(\Phi(\mathbf{h} * (\mathbf{x}\odot\boldsymbol{\xi})))\|^2 + \lambda\sum_{t=1}^{T}\xi_t \\ \text{s.t.} \quad & (\forall t \in [\![1,T]\!]) \quad \xi_t = \xi_t^2\,. \end{aligned} \tag{4.8}$$

The ξ_i's are introduced to formulate the ℓ_0 penalization in a polynomial form, while the constraints ensure that they are binary variables. Unfortunately, this formulation results in twice as many variables than in the original problem. As a consequence, we will show in Section 4.5 that the relaxation resulting from (4.8) has a high complexity. The method presented below allows us to overcome this complexity barrier by using a different formulation of (4.5).

Looking closer at the relaxations of ℓ_0 mentioned in Section 4.2.2, an original alternative approach consists in considering penalization functions that are piecewise rational and can be expressed under the general form

$$(\forall x \in \mathbb{R}) \quad \Psi_\lambda(x) = \sum_{i=1}^{I} \zeta_i(x) \mathbb{1}_{\{\sigma_{i-1} \leq x < \sigma_i\}}\,, \tag{4.9}$$

where I is a nonzero integer, $(\zeta_i)_{i\in[\![1,I]\!]}$ are rational functions, and $(\sigma_i)_{i\in[\![0,I]\!]}$ is an increasing sequence of real values. The resulting criterion is thus a sum of rational and piecewise rational functions:

$$\begin{aligned} \mathcal{J}(\mathbf{x}) = &\sum_{u=1}^{U} g_u(x_{\Delta(u,\alpha)-L+1}, \ldots, x_{\Delta(u,\alpha)}) \\ &+ \sum_{t=1}^{T} \sum_{i=1}^{I} \zeta_i(x_t) \mathbb{1}_{\{\sigma_{i-1} \leq x_t < \sigma_i\}}\,. \end{aligned} \tag{4.10}$$

4.3.2 Piecewise rational criteria

In this section, we first show how to transform the piecewise rational criterion in Problem (4.10) into the equivalent minimization of a sum of rational functions under polynomial constraints. To do so, we introduce the binary variables $\left(z^{(i)}\right)_{i\in[\![1,I]\!]}$ such that

$$(\forall i \in [\![0, I]\!]) \quad z^{(i)} = \mathbb{1}_{\{\sigma_i \leq x\}}\,.$$

We set $\sigma_0 = -\infty$, $z^{(0)} = 1$ and $\sigma_I = +\infty$, $z^{(I)} = 0$ to define Ψ_λ on the whole real line $\mathbb{R}$. From the definition of $\left(z^{(i)}\right)_{i\in[\![1,I]\!]}$, we deduce that

$$(\forall i \in [\![0, I]\!]) \quad \mathbb{1}_{\{\sigma_{i-1} \leq x < \sigma_i\}} = z^{(i-1)}(1 - z^{(i)}). \tag{4.11}$$

Finally, the constraint $z^{(i)} = \mathbb{1}_{\{\sigma_i \leq x\}}$ is equivalent to two polynomial constraints

$$z^{(i)} = \mathbb{1}_{\{\sigma_i \leq x\}} \iff \begin{cases} \left(z^{(i)}\right)^2 - z^{(i)} = 0 \\ \left(z^{(i)} - \frac{1}{2}\right)(x - \sigma_i) \geq 0\,. \end{cases} \tag{4.12}$$

Indeed, the polynomial equality constraint enforces $z^{(i)}$ to be a binary variable while the polynomial inequality constraint ensures that it takes the same values as $\mathbb{1}_{\{\sigma_i \leq x\}}$ for every x in $\mathbb{R}$. Therefore, substituting (4.12) in (4.10), Problem (4.7) reads as the minimization of a sum of rational functions depending on vectors $\mathbf{x}$ and $\mathbf{z} = \left(\mathbf{z}^{(i)}\right)_{i\in[\![0,I]\!]}$ under polynomial constraints, namely

$$\begin{aligned} \underset{(\mathbf{x},\mathbf{z})\in\mathbb{R}^T\times\mathbb{R}^{IT}}{\text{minimize}} \quad & \sum_{u=1}^{U} g_u(x_{\Delta(u,\alpha)-L+1}, \ldots, x_{\Delta(u,\alpha)}) \\ & + \sum_{t=1}^{T} \sum_{i=1}^{I} \zeta_i(x_t) z_t^{(i-1)}(1 - z_t^{(i)}) \\ \text{s.t.} \quad & (\forall (i,t) \in [\![0, I]\!] \times [\![1, T]\!]) \quad \begin{cases} \left(z_t^{(i)}\right)^2 - z_t^{(i)} = 0 \\ \left(z_t^{(i)} - \frac{1}{2}\right)(x_t - \sigma_{i+1}) \geq 0\,. \end{cases} \end{aligned} \tag{4.13}$$

More generally, this reformulation can be applied to the minimization of any piecewise rational function. For instance, a piecewise rational fit function $f_{\mathbf{y}}$ could also be chosen.

4.3.3 Symmetry of regularizers

All the piecewise rational approximations to the ℓ_0 penalty listed in Section 4.2.2 are even functions. This symmetry property is expressed here by an absolute value on the input variable x_t in the expressions of function Ψ_λ. This absolute value is handled in our framework by adding an additional variable r_t for each x_t and adding the two constraints

$$\begin{cases} r_t^2 = x_t^2 \\ r_t \geq 0\,. \end{cases} \tag{4.14}$$

This symmetry is important to decrease the number I of variables $\mathbf{z}$ involved in (4.13) and therefore to reduce the overall complexity of the final problem to be solved, as will be explained in Section 4.5. Indeed, it can divide by two the number of pieces in Ψ_λ, leading to only $I/2$ pieces instead of I. Taking the example of the MCP penalization, instead of having the four intervals, $]-\infty, -\gamma\lambda[$, $[-\gamma\lambda, 0[$, $[0, \gamma\lambda[$, and $[\gamma\lambda, +\infty[$, we have only the two intervals, $[0, \gamma\lambda[$ and $[\gamma\lambda, +\infty[$. Using symmetry results in adding one variable r_t and $I/2$ variables $\left(z_t^{(i)}\right)_{i\in[\![1,I/2]\!]}$ for each x_t as well as $2+I/2$ polynomial constraints corresponding to constraints (4.12) and (4.14). This has to be compared with the direct formulation where we introduce I variables $\left(z_t^{(i)}\right)_{i\in[\![1,I]\!]}$ with I polynomial constraints. Note that in our analysis of Section 4.5, we omit the equality constraints that force $\left(z_t^{(i)}\right)_{i\in[\![1,I/2]\!]}$ to be binary variables since substitution will be performed for those constraints in Section 4.5.2.

§ 4.4 Solving the optimization problem

This section is concerned with the resolution of Problem (4.13). We apply the method developed in Chapter 3. More specifically, we leverage the structure of our problem as detailed in Section 3.2. Indeed, we have to handle a sum of $\tilde{I} = U + T$ terms. We hence introduce a measure for each term, i.e. U measures $(\mu_u)_{u\in[\![1,U]\!]}$ for the rational functions $(g_u)_{u\in[\![1,U]\!]}$ and T measures $(\nu_t)_{t\in[\![1,T]\!]}$ for the rational functions in the reformulated penalization. The measures $(\mu_u)_{u\in[\![1,U]\!]}$ are measures on at most L variables $x_{\Delta(u,\alpha)-L+1}, \ldots, x_{\Delta(u,\alpha)}$ while the measures $(\nu_t)_{t\in[\![1,T]\!]}$ are measures on $I+1$ scalar variables, corresponding to x_t and $\mathbf{z}_t$.

4.4.1 Feasible set

As we often work with bounded signals, we assume that our signal verifies Assumption (3.3). In Problem (4.13), the sets $(\mathcal{K}_i)_{i\in[\![1,U+T]\!]}$ are thus defined by the bound constraints and the polynomial constraints arising from the reformulation of Section 4.3.2. Namely, the sets $(\mathcal{K}_i)_{i\in[\![1,U]\!]}$ are defined by

$$(\forall i \in [\![1, U]\!]) \quad (\mathbf{B} - \mathbf{x}_{E_i}) \odot (\mathbf{x}_{E_i} + \mathbf{B}) \geq \mathbf{0}\,, \tag{4.15}$$

while the sets $(\mathcal{K}_i)_{i\in[\![U+1,U+T]\!]}$ are defined by

$$(\forall i \in [\![U+1, U+T]\!]) \quad \begin{cases} (B - x_{i-U})(x_{i-U} + B) \geq 0 \\ (\forall j \in [\![1, I]\!]) \quad \left(z_{i-U}^{(j)}\right)^2 - z_{i-U}^{(j)} = 0 \\ (\forall j \in [\![1, I]\!]) \quad \left(z_{i-U}^{(j)} - \dfrac{1}{2}\right)(x_{i-U} - \sigma_{j+1}) \geq 0\,. \end{cases} \tag{4.16}$$

Note that, since we introduce a measure for each rational function in the sum, we have to cope with more than T bound constraints. Indeed several measures are defined on identical variables and we need to introduce bound constraints for each of those measures.

By definition of the convolution matrix $\mathbf{H}$ in Section 4.2, the sets $(E_i)_{i\in[\![1,\tilde{I}]\!]}$ satisfy the running intersection property. We then perform the relaxation (3.12) to generate a hierarchy of SDP problems.

4.4.2 Coupling and linear equality constraints

We observe two kinds of coupling as discussed in Section 3.2.1: one between the different measures $(\mu_u)_{u\in[\![1,U]\!]}$ and one between the measures $(\mu_u)_{u\in[\![1,U]\!]}$ and the measures $(\nu_t)_{t\in[\![1,T]\!]}$. We remark that many of moment equality constraints between moments of the measures $(\mu_u)_{u\in[\![1,U]\!]}$ are redundant. Following Section 3.2.4, it is thus sufficient to consider only constraints on consecutive measures $(\mu_u)_{u\in[\![1,U]\!]}$ in (4.13). We can thus drastically reduce the number of moment linear equality constraints.

§ 4.5 Reducing the complexity of the relaxations

Current state-of-the-art SDP solvers are interior point methods which are known to be very efficient for small and medium scale problems. Nevertheless, their running time becomes prohibitive for large scale problems. More specifically, the current bottleneck is mainly the dimension of both n and m. This is a major drawback when relaxing rational optimization problems into SDP problems. Chapter 6 is dedicated to explore alternative methods to solve such SDP problems. Nevertheless, in Sections 3.2 and 4.4, we worked on the structure of the relaxation and leveraged the latter to yield a structured and alleviated SDP problem. This section gives an asymptotic estimation for n and m depending on the parameters of our initial model (4.1) and the relaxation order k. We give a more detailed derivation for the expression of (n, m, m_s, ℓ) in Appendix A.

Our analysis reveals that signal processing problems are computationally tractable when using sparsity patterns and subsampling. We first show that subsampling and sparsity allow to overcome the latter bottleneck and make the numerical resolution of the associated SDP problems tractable. Then, we introduce tools and tricks that allow us to decrease further the dimension of the SDP problems to be solved, so reducing the computational time of our method.

4.5.1 Consequence of the subsampling on the dimensions of the SDP problem

For a given relaxation order k, when the number of samples T goes to infinity and $L \gg 1$ (i.e. we lose the band structure of $\mathbf{H}$), the size of the SDP problem asymptotically becomes of the order (see Appendix A)

$$m = \mathcal{O}(UL^{2k} + T) \quad , \quad n = \mathcal{O}(UL^k + T)\,. \tag{4.17}$$

We note that both sizes n and m grow exponentially with k and blow up quickly. In particular, m grows faster than n. However, we will see that the SDP hierarchy often converges quickly in practice, that is $(\mathcal{J}_k^*)_{k\in\mathbb{N}}$ converge to $\mathcal{J}^*$ for a relaxation order k of 2, 3, or 4. From our analysis, we observe that the main bottleneck of our method is the number of variables per measure and the order of relaxation. While the number of variables in measures $(\nu_t)_{t\in[\![1,T]\!]}$ is fixed to $I+1$, the total number of variables in measures $(\mu_u)_{u\in[\![1,U]\!]}$ is L and (4.17) shows that m and n rise quickly with L.

Although the subsampling reduces the quality of the reconstruction by eliminating some information of the signal, it also has, in our context, the beneficial side effect of allowing the size of the SDP relaxation to be reduced. As shown by (4.3), decimation decreases U, which plays a prominent role in the complexity parameters $(n, m, m_{\mathrm{s}}, \ell)$ of the SDP problem. Table 4.1 compares the size of SDP relaxations for the SCAD penalization without decimation (D_∞) and with D_2 (resp. D_4) decimation. As discussed above, the dimensions n and m increase quickly with the relaxation order k and the length of the filter L. Note that because of the approximation made in Appendix A.1, stating that measures $(\mu_u)_{u\in[\![1,U]\!]}$ are on L variables, the SDP dimension presented here are slightly overestimated.

Table 4.1: *Dimension of the relaxation of the SCAD penalization for different decimations.*

			m			n			m_{s}			ℓ		
T	L	k	D_∞	D_4	D_2	D_∞	D_4	D_2	D_∞	D_4	D_2	D_∞	D_4	D_2
50	3	3	8400	7476	6300	7000	6450	5750	600	556	500	1035	735	420
100	3	3	16800	14784	12600	14000	12800	11500	1200	1104	1000	2085	1755	845
50	4	3	14700	12390	9450	9250	8205	6875	650	595	425	2015	1355	660
100	4	3	29400	24360	18900	18500	16220	13750	1300	1180	1050	4065	3405	1335
100	5	3	54600	43512	31500	25600	21236	17050	1400	1256	1100	7530	6375	2315
50	3	4	16500	14685	12375	13500	12455	11125	600	556	500	1772	1184	568
100	3	4	33000	29040	24750	27000	24720	22250	1200	1104	1000	3572	2102	1143
100	4	4	66000	54120	41250	38500	33460	28000	1300	1180	1050	9116	7268	2172

4.5.2 Polynomial equality constraints and substitution

For a given measure, equality constraints involving monic monomials in the definition of the support set $\mathcal{K}_i$ can be substituted. The constraint is then used to reduce the number of moments in the vector of moments. We clarify this process here through the example of the SCAD penalization. Substitution is carried out automatically by some software [HLL09], but has not been clearly documented.

Let us focus our attention on the measure ν_t, depending on the three variables x_t, $z_t^{(1)}$, and $z_t^{(2)}$, as well as on the associated truncated vector $\mathbf{v}_t$ of moment up to degree 2. Using the equality constraints in (4.16), we substitute the related monomial in $\mathbf{v}_t$. The moments associated with monomials $\left(z_t^{(1)}\right)^2$ and $\left(z_t^{(2)}\right)^2$ are thus the same as the ones associated with $z_t^{(1)}$ and $z_t^{(2)}$. Therefore, the moment vector $\mathbf{v}_t$ has a dimension reduced by two. When $\mathbf{v}_t$ contains moments up to degree $2k$, substitution reduces the number of moments from $\binom{3+2k}{2k}$ to $8k$.

In the general case, for a given relaxation order k, $\mathbf{v}_t$ contains only $2k(k+1)$ moments after substitution which is much fewer than the original $\binom{1+I+2k}{2k}$ moments. Substitution significantly decreases the values of n, m and m_{s} which all have a major impact on the computational cost of SDP solvers. However, it does not impact the number of linear constraints ℓ.

4.5.3 Sign oracle

For real-valued signals $\overline{\mathbf{x}}$, convergence is observed at orders k for which building and solving the corresponding SDP problems is highly demanding in terms of computation and memory storage. Conversely, when $\overline{\mathbf{x}}$ is a positive signal, we observed [CPM19] convergence at a lower order k. This suggests a method yielding similar results for real-valued signals using an oracle. Instead of

$$(\forall \mathbf{x} \in \mathbb{R}^T) \quad \mathcal{J}(\mathbf{x}) = \frac{1}{2} \left\| \mathbf{y} - D_\delta(\Phi(\mathbf{H}\mathbf{x})) \right\|^2 + \sum_{t=1}^{T} \Psi_\lambda(x_t)\,,$$

we minimize

$$(\forall \mathbf{x} \in \mathbb{R}_+^T) \quad \widetilde{\mathcal{J}}(\mathbf{x}) = \frac{1}{2} \left\| \mathbf{y} - D_\delta(\Phi(\widetilde{\mathbf{H}}\mathbf{x})) \right\|^2 + \sum_{t=1}^{T} \Psi_\lambda(x_t)\,,$$

where $\widetilde{\mathbf{H}} = \mathbf{H}\,\mathrm{Diag}(\boldsymbol{\epsilon})$, $\boldsymbol{\epsilon} \in \{-1, 1\}^T$ is the sign vector of $\overline{\mathbf{x}}$ provided by the oracle, and $\mathrm{Diag}(\boldsymbol{\epsilon})$ is a diagonal matrix with the vector $\boldsymbol{\epsilon}$ on the diagonal. An oracle is built by solving a standard least absolute shrinkage and selection operator (LASSO) problem [Tib96], i.e. we choose for the function Ψ_λ the ℓ_1 norm weighted with the parameter λ, $\lambda|.|$. Note that our oracle is not a true oracle but more a sign estimator as there is a probability that it gets the sign wrong. The availability of an oracle allows us to restrict the minimization of (4.10) to positive valued signals thanks to the new convolution matrix $\widetilde{\mathbf{H}}$. Our oracle decreases significantly the computational time in two ways:

- Since the convergence of the SDP hierarchy occurs for smaller order k, the dimensions of the SDP problem to solve are much lower according to Section 4.5.
- Moreover, since we optimize now on positive variables, we do not need to use the additional variables $(r_t)_{t \in [\![1,T]\!]}$ introduced in Section 4.3.3 to account for symmetries and the presence of absolute values. This results in smaller vectors of moments, hence a lower dimensional SDP problem.

An exact solution is thus retrieved by solving an SDP problem of fair dimension. Finally, the computational cost of our oracle is low since we solve a LASSO using a forward-backward algorithm. It typically takes less than a second which is negligible compared to the computational time of our method as shown in Section 4.6 while providing accurate oracle on the sign of the initial signal.

§ 4.6 Numerical simulations

4.6.1 Experimental set-up

To show the efficiency of our framework, we apply it to the reconstruction of a sparse signal subject to nonlinear distortion and subsampling. We use a piecewise relaxation of ℓ_0 to promote sparsity as detailed in Section 4.2.2. We perform simulations on 60 test cases where the initial sparse signal $\overline{\mathbf{x}}$ has length $T = 100$ with 10 non-zero values. Those values are drawn randomly according to a uniform distribution on $[-1, -0.1] \cup [0.1, 1]$. The position of the non-zero values are also drawn randomly according to uniform distribution on $[\![1, T]\!]$. The length L of the filter is set to 3 and its coefficients are sampled from a Gaussian distribution. This kind of filter is useful to model enlargement due to

measurement from sensors for example. We choose the following saturation function for the nonlinear distortion ϕ

$$(\forall t \in \mathbb{R}) \quad \phi(t) = \frac{t}{\chi + |t|},$$

where χ is set to 0.3. Finally, we perform the relaxation into SDP for relaxation orders 2, 3, and 4. We use GloptiPoly [HLL09] to relax rational problems into SDP problems which are then solved with the solver SDPT3 [TTT99]. All the simulations have been run on a standard computer with an Intel Xeon CPU running at 3.7 GHz and 32 GB of RAM allocated to the process.

4.6.2 Example of rational relaxation: SCAD

To clarify the reformulation of Section 4.3.2, we demonstrate it on the regularizers given in Section 4.2.2. Taking advantage of symmetry as explained in Section 4.3.3, SCAD has three pieces and thus requires to introduce variables $z_t^{(1)}$ and $z_t^{(2)}$ leading to

$$\begin{aligned}
\underset{\mathbf{x},\mathbf{z}}{\text{minimize}} \quad & f_{\mathbf{y}}(\mathbf{x}) + \sum_{t=1}^{T} (1 - z_t^{(1)})\lambda\,|x_t| + z_t^{(2)}\frac{(\gamma+1)\lambda^2}{2} \\
& - z_t^{(1)}(1 - z_t^{(2)})\frac{\lambda^2 - 2\gamma\lambda\,|x_t| + x_t^2}{2(\gamma-1)} \\
\text{s.t.} \quad & (\forall (i,t) \in \{1,2\} \times [\![1,T]\!]) \quad \left(z_t^{(i)}\right)^2 - z_t^{(i)} = 0 \\
& (\forall t \in [\![1,T]\!]) \quad \left(z_t^{(1)} - \frac{1}{2}\right)(|x_t| - \gamma\lambda) \geq 0 \\
& (\forall t \in [\![1,T]\!]) \quad \left(z_t^{(2)} - \frac{1}{2}\right)(|x_t| - \lambda) \geq 0\,.
\end{aligned} \tag{4.18}$$

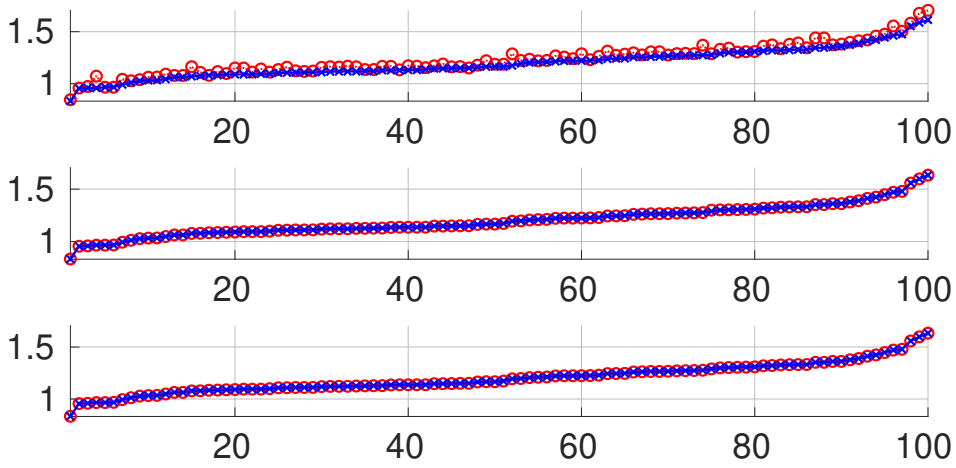
A similar approach applies to Capped ℓ_p, MCP, and CEL0 penalties; the details are omitted for conciseness. Although we use SCAD penalization in all the subsequent simulations, similar results can be obtained with Capped ℓ_p, MCP, and CEL0. Nonetheless, SCAD is more demanding in terms of computation since it has more rational pieces. It consequently provides a worst case scenario for the computational time compared with the other penalizations. The parameter γ for SCAD is set to 2.1 in order to approximate ℓ_0 closely. The value of the parameter λ was determined empirically and set to 0.15.

4.6.3 Acceleration of convergence with the sign oracle

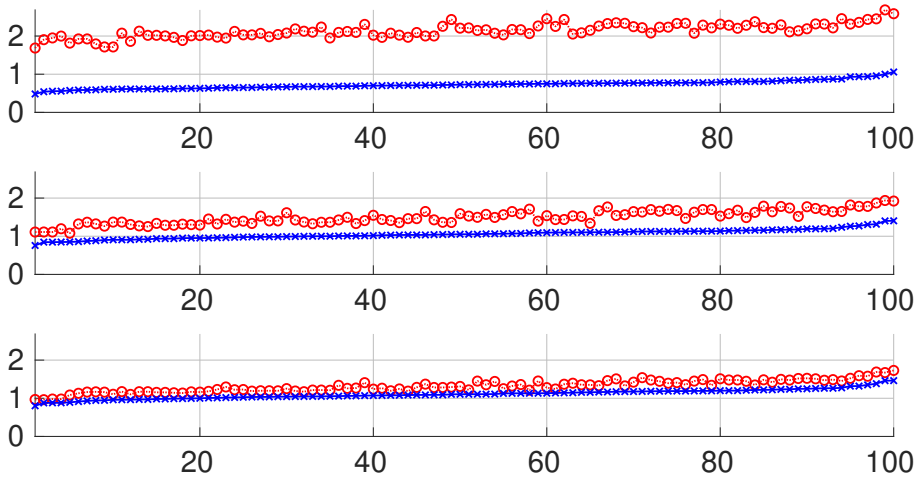
In this section, we want to show how the oracle impacts the convergence of the SDP hierarchy. We first consider the use of a sign oracle in a linear model, i.e. the case when $\phi = \mathrm{Id}$. We then delve into the more challenging case of a nonlinear model. The decimation is set to D_4 in this section. The oracle is build on solving a LASSO problem by using a forward-backward algorithm as described in Section 4.5.3.

4.6.3.1 Linear case

Solving each SDP problem in the hierarchy provides both a lower bound $\mathcal{J}_k^*$, which is the value of the objective function of the SDP at optimality, and an approximate minimizer $\hat{\mathbf{x}}_k$, which is extracted from a minimizer of the SDP problem. We compare here the value of the criterion at $\hat{\mathbf{x}}_k$ with $\mathcal{J}_k^*$. Since increasing the relaxation order k yields larger lower bounds and smaller criterion values, we consider that the convergence of the hierarchy happens when $\mathcal{J}(\hat{\mathbf{x}}_k)$ and $\mathcal{J}_k^*$ are equal. Figure 4.2 compares those two values respectively in the cases with oracle and without the use of our oracle on 100 test



(a) *With oracle:* $k = 2$ *(top),* 3 *(middle)* 4 *(bottom).*



(b) *Without oracle:* $k = 2$ *(top),* 3 *(middle)* 4 *(bottom).*

Figure 4.2: *Comparison between the lower bound* $\mathcal{J}_k^*$ *and the value of the criterion* $\mathcal{J}(\hat{\mathbf{x}}_k)$ *for 100 tests (Linear Case).*

cases. From top to bottom, the two figures are drawn for relaxation orders $k = 2$, $k = 3$, and $k = 4$. Criterion values are represented in red while lower bound are represented in blue. Each point of the x-axis represents the values for a single test case. For the sake of clarity, the values are ordered according to the value of the lower bound. We observe that, without oracle, the convergence is slow and still not reached in general at order $k = 4$. On the other hand, when we use our oracle, convergence appears quickly, i.e. $k = 3$ in most of the test cases.

4.6.3.2 Nonlinear case

Figure 4.3 is similar to Figure 4.2 but in the context of a nonlinear model. The continuous line with cross dots represents the cases without the use of an oracle while the dashed line with circle dots represents the cases with our sign oracle. We observe here that even with a sign oracle, the convergence of the hierarchy does not occur for low values of k

due to the nonlinearity. However, we can notice that the gap between the lower bound and the criterion value at the $\hat{\mathbf{x}}_k$ is greatly reduced when we use our oracle.

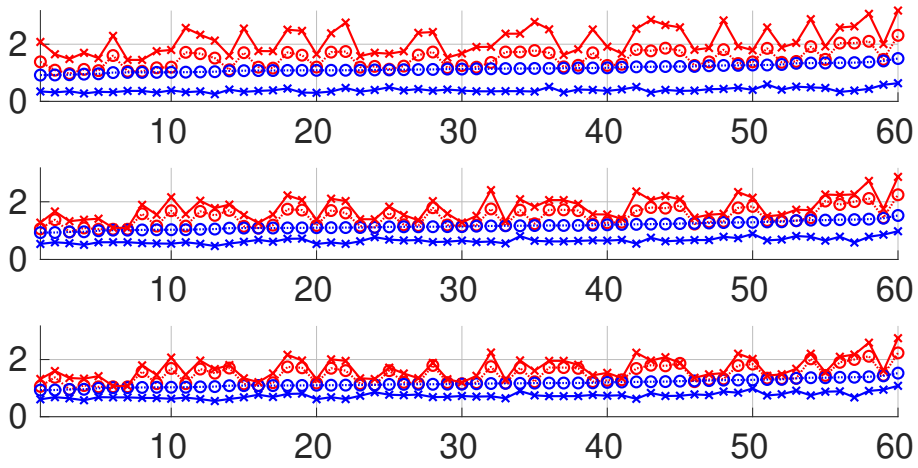


Figure 4.3: *Comparison between the lower bound $\mathcal{J}_k^*$ and the value of the criterion $\mathcal{J}(\hat{\mathbf{x}}_k)$ when the oracle is used. Plain line with cross dots: no oracle used, dashed line with circle dots: use of our oracle (Nonlinear case).*

4.6.4 Reconstruction of sparse signals

4.6.4.1 Global optimality

In this section, we want to demonstrate the quality of the minimizers of Problem (4.18) returned by various methods. Note that we do not use the oracle here. We use the decimation operator D_4 but similar results hold for the other operators. We compare our method to a Forward-Backward algorithm (FB) applied directly to the criterion $\mathcal{J} = f_{\mathbf{y}} + \mathcal{R}_\lambda$ where the gradient step is first performed on the data fitting term and a proximal step is then performed on the penalization. Hence the criterion to minimize is the same for both methods. We initialize the FB algorithm first with the null vector and denote by $\mathbf{x}_{\text{FB0}}$ the resulting solution. Then we perform a warm start of the FB algorithm using the solution obtained from our method as an initializer. The resulting estimate is denoted by $\mathbf{x}_{\text{FB1}}$.

In Figure 4.4, we compare the value of the criterion $\mathcal{J}$ at $\mathbf{x}_{\text{FB0}}$ and $\mathbf{x}_{\text{FB1}}$ with the solution returned by our method for a relaxation order $k = 4$. The solid blue curve with cross dots represents the values of the lower bound $\mathcal{J}_4^*$, the pointed red curve with circle dots represents $\mathcal{J}(\hat{\mathbf{x}}_4)$, the dashed green curve with plus dots represents $\mathcal{J}(\mathbf{x}_{\text{FB0}})$, and the dashed purple curve with plus dots represents $\mathcal{J}(\mathbf{x}_{\text{FB1}})$.

Since the criterion $\mathcal{J}$ is highly nonconvex, the forward-backward algorithm gets stuck in local minimizers. Indeed, changing the initialization point changes the output of the algorithm. We can observe it on Figure 4.4 where the green and purple curves are not superposed. Moreover, we observe that the convergence in the hierarchy has not occurred at order 4 according to Section 4.6.3.2. A solution to improve the quality of the minimizer is to use the solution $\hat{\mathbf{x}}_4$ as a warm start of the FB algorithm as shown by the purple curve.

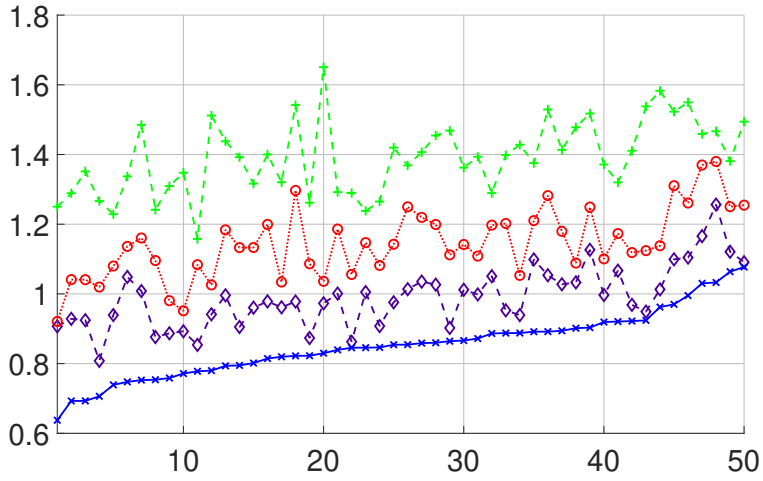


Figure 4.4: *Comparison between the different values of the criterion for the minimizers returned by the different methods. In red* $\mathcal{J}(\hat{\mathbf{x}}_4)$*, in blue* $\mathcal{J}_4^*$*, in green* $\mathcal{J}(\mathbf{x}_{\mathrm{FB0}})$*, and in purple* $\mathcal{J}(\mathbf{x}_{\mathrm{FB1}})$*.*

4.6.4.2 Quality of signal reconstruction

We now look at the quality of the signal reconstruction in terms of mean square error: our method is compared with several other ones to illustrate its interest for faithful recovery of the original signal $\overline{\mathbf{x}}$. In addition to the FB algorithm presented in Section 4.6.4.1, we compare our method with the oracle to iLASSO, a LASSO approach modified to handle the nonlinearity of the model. It consists first in applying the LASSO using a linearization of the nonlinear operator ϕ. Namely, it solves

$$\arg\min_{\mathbf{x}\in\mathbb{R}^T} \|\mathbf{y} - D_\alpha(\mathcal{L}_\phi(\mathbf{h} * \mathbf{x}))\|^2 + \lambda_{\mathrm{LASSO}} \|\mathbf{x}\|_1 \ ,$$

where $\mathcal{L}_\phi$ is a linearization of ϕ and λ_{LASSO} is a parameter set empirically to 0.1. Note that for our choice of ϕ, the linearization $\mathcal{L}_\phi$ is a constant scaling factor given by χ^{-1}. We subsequently apply a modified iterative hard thresholding (IHT) that handles the nonlinearity. Namely, we apply the FB algorithm to find

$$\arg\min_{\mathbf{x}\in\mathbb{R}^T} \|\mathbf{y} - D_\alpha(\Phi(\mathbf{h} * \mathbf{x}))\|^2 + \lambda_{\mathrm{IHT}} \ell_0(\mathbf{x}) \, ,$$

where we perform a gradient step on the data fidelity component and a proximal step on the penalization $\lambda_{\mathrm{IHT}}\ell_0$. This method provides better reconstruction results than the FB algorithm presented in Section 4.6.4.1. We also compare our method to the Iteratively Reweighted ℓ_1 algorithm (IRL1) [CWB08] applied to

$$\arg\min_{\mathbf{x}\in\mathbb{R}^T} \|\mathbf{y} - D_\alpha(\mathcal{L}_\phi(\mathbf{h} * \mathbf{x}))\|^2 + \mathcal{R}_{\lambda_{\mathrm{IRL1}}}(\mathbf{x}) \, ,$$

where $\mathcal{R}_\lambda$ is the SCAD regularization. Both IRL1 and FB algorithms are initialized with the null vector.

Figure 4.5 illustrates the different signals for a single realization using D_2 decimation. From top to bottom, we display the original signal $\overline{\mathbf{x}}$, the subsampled observed signal $\mathbf{y}$, the signal reconstructed respectively with iLASSO $\mathbf{x}_{\mathrm{iLASSO}}$, and the signal reconstructed using our method $\hat{\mathbf{x}}_3$ at the relaxation order $k = 3$. We do not display the signal reconstructed with FB and IRL1 since those algorithms are not well suited for solving (4.10)

and thus provide poor quality reconstruction. We first notice that iLASSO misses many peaks and also detects a peak that does not exist in the original signal while our method detects almost all peaks. One could argue the threshold coefficient λ_{IHT} in iLASSO is too high but, when we decrease it, small artifacts appear. In contrast, our method detects almost all peaks and do not leave any artifacts. We observe that some peaks do not have the same amplitude as the ones in the original signal. This is due to subsampling. Indeed, if a peak is located on an even index, it will be eliminated by the subsampling. However, the convolution with $\mathbf{h}$, that represents the physical limitation of sensors in our example, allows us still to recover the peak since it gets enlarged to odd neighboring. Even though, we lose information about the amplitude of this peak.

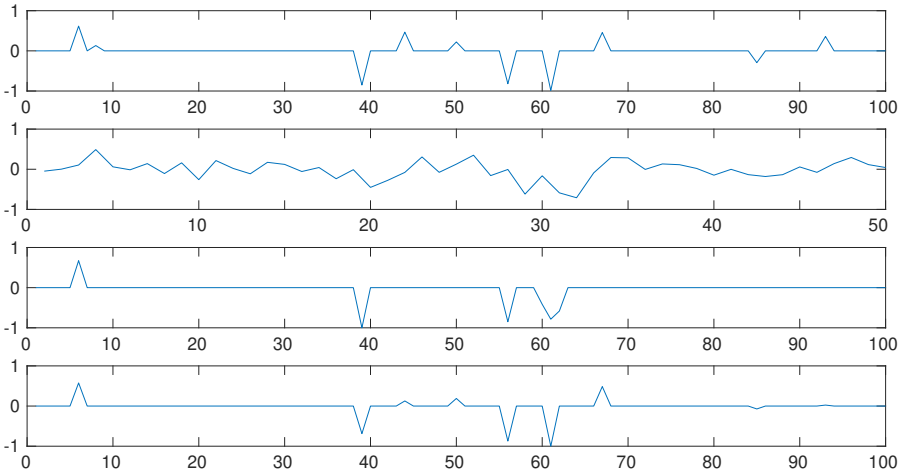


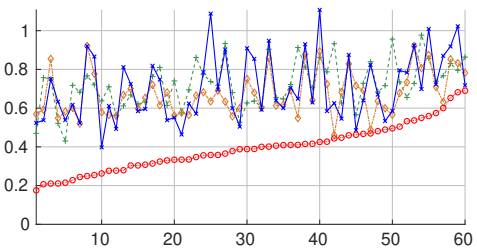
Figure 4.5: *Comparison between iLASSO and our method for signal reconstruction under nonlinear transformation and subsampling. From top to bottom: the original signal* $\overline{\mathbf{x}}$*, the observed signal* $\mathbf{y}$*, and respectively the signal reconstructed with iLASSO* $\mathbf{x}_{\text{iLASSO}}$ *and with our method* $\hat{\mathbf{x}}_3$*.*

Figures 4.6 shows the mean square error $\|\overline{\mathbf{x}} - \mathbf{x}\| / \|\overline{\mathbf{x}}\|$ for D_∞, D_4 and D_2 decimation between the original signal $\overline{\mathbf{x}}$ and: in green $\mathbf{x}_{\text{FB}}$, in orange $\mathbf{x}_{\text{IRL1}}$, in blue $\mathbf{x}_{\text{iLASSO}}$, and in red $\hat{\mathbf{x}}_4$. Those confirm the good reconstruction result shown in the specific example of Figure 4.5.

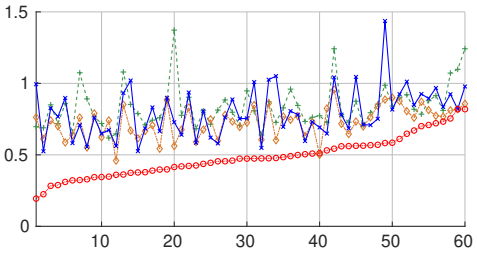
Finally, Table 4.2 shows the average computational times for different decimation operators and relaxation orders. As we expected, the better performance of our method comes at the expense of a higher computational cost than iLASSO, which takes less than 1 second.

Table 4.2: *Computation time of our method (in seconds).*

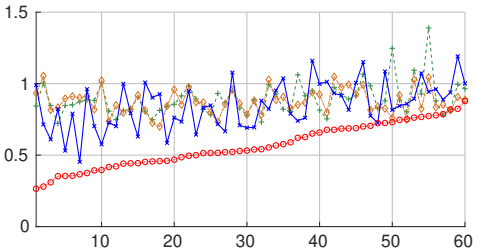
	Without oracle			With oracle		
	D_∞	D_4	D_2	D_∞	D_4	D_2
$k=2$	41	35	29	38	31	25
$k=3$	162	121	87	144	106	74
$k=4$	29991	14575	5801	24362	11062	4084



(a) D_∞ *decimation.*



(b) D_4 *decimation.*



(c) D_2 *decimation.*

Average values	D_∞	D_4	D_2
$\mathbf{x}_{\text{FB}}$	0.72	0.84	0.89
$\mathbf{x}_{\text{IRL1}}$	0.67	0.73	0.87
$\mathbf{x}_{\text{iLASSO}}$	0.71	0.80	0.84
$\hat{\mathbf{x}}_4$	0.39	0.48	0.56

Figure 4.6: *Mean square error between the estimated signal and the original signal* $\overline{\mathbf{x}}$*. In dashed green:* $\mathbf{x}_{\text{FB}}$*, in dashed orange:* $\mathbf{x}_{\text{IRL1}}$*, in blue:* $\mathbf{x}_{\text{iLASSO}}$*, and in dotted red:* $\hat{\mathbf{x}}_4$*. Average values are shown in the table.*

4.6.4.3 Handling higher-dimensional signal

Although our method provides good reconstruction results for medium-size signals, handling higher-dimensional signals is highly demanding in terms of computations as shown in our study of Section 4.5.1 and in Table 4.2. Moreover, we observed that the memory requirements of the SDP solver for its internal process become too important. To tackle this issue, we split the signal into smaller overlapping chunks that are processed independently and then reassembled together. We illustrate the example of Figure 4.7 where we reconstruct a signal of dimension $T = 1000$ using 11 chunks of length 100 with 10 overlapping samples on both extremities. The overlapping sections are averaged in order to obtain the final signal. The decimation operator is D_∞ and the relaxation order is set to 3. We observe that our method yields a better reconstruction than iLASSO with a mean square error of 0.43 against 0.69 for iLASSO.

§ 4.7 Summary

In this chapter, we have proposed a method to globally solve nonconvex problems involving exact relaxation of ℓ_0 in order to reconstruct sparse signal from degraded observations. One of the main advantages of our method is that it is able to deal with nonlinear degradations. We have first reformulated our piecewise rational criterion into a rational optimization problem before applying the framework of Chapter 3 that benefits from the sparsity of the rational functions. We have then discussed the complexity of the obtained SDP relaxation and methods to decrease both the dimension of the latter and the converging order in the hierarchy. Finally, our simulations illustrate the domain of applicability of the method and its high potential for finding a good approximation to a global minimum. Although providing good results for medium-size problems, our method shows computational limitations for larger-scale signals. In the next chapter, we

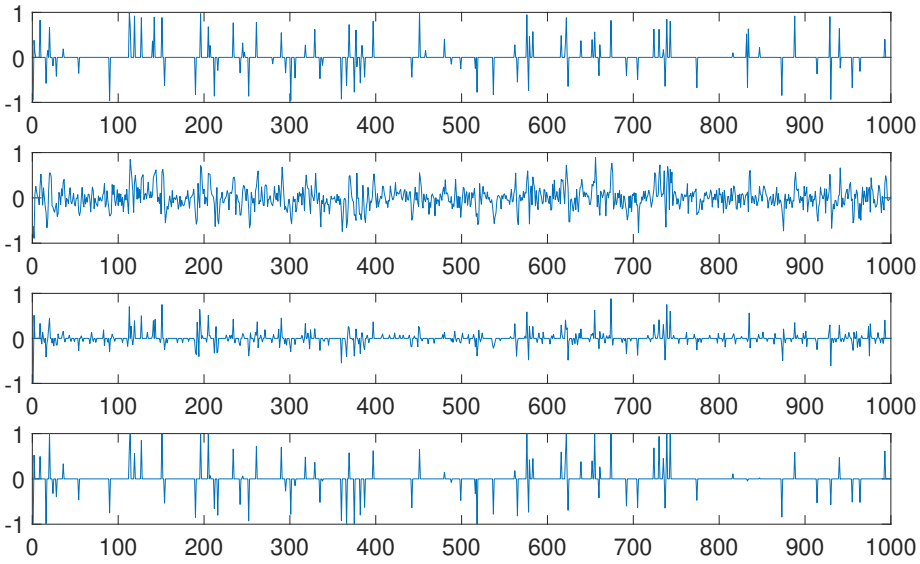


Figure 4.7: *Reconstruction of higher-dimensional signals ($T = 1000$). From top to bottom: the original signal $\overline{\mathbf{x}}$, the observed signal $\mathbf{y}$, and respectively the signal reconstructed with iLASSO $\mathbf{x}_{\text{iLASSO}}$ and with our method $\hat{\mathbf{x}}_3$.*

propose to extend this methodology to non-Gaussian noise.

- Chapter 5 -

Signal Reconstruction for non-Gaussian noise

§ 5.1 Background

The additive white Gaussian noise is a widely used model. Its prominence due to its simplicity and to the central limit theorem makes Gaussian distribution a good approximation to the noise in various models such as the electrical noise of sensors and detectors in the previous chapter. However, this approximation may not be suitable in some specific applications. For instance, Poisson distribution is a more faithful model for the photon shot noise of image sensors [TPAB09]. Another important noise for practical applications that does not follow a Gaussian distribution is caused by the presence of outliers in the observations, for instance due to sensor malfunction.

The goal of this chapter is to extend the methodology exposed in Chapter 4 to handle some non-Gaussian noises. This corresponds to alternative choices of the fit function. Hence, in Section 5.2, we first show how to reconstruct efficiently signals corrupted with Poisson-Gaussian (PG) noise but that show some form of sparsity. Section 5.2.1 introduces the considered data model as well as the general form of the addressed optimization problem. Section 5.2.2 reformulates the original optimization problem into a rational optimization one. Numerical simulations and results are presented in Section 5.2.3 to validate our approach. This work has been done in collaboration with Anna Jezierska from Gdansk University of Technology.

Then in Section 5.3, we adapt the methodology of Chapter 4 to a robust scheme that reduces the impact of possible outliers. In addition, some restrictions on the sought signal are considered under the form of nonconvex unions of subsets. Section 5.3.1 introduces the model for the observed signal. Section 5.3.2 details the reformulation of the optimization problem that we follow to reconstruct the initial signal. Finally, simulation results are presented in Section 5.3.3.

§ 5.2 Poisson-Gaussian noise

Over the last decades, there has been a growing interest for signal reconstruction from measurements corrupted by PG noise. Examples of application areas include fluorescence microscopy [CJPT15], low dose computer tomography [DLZF18, LLK18, ZXZ19], and Visible Light Communication (VLC) [CHYT17]. When PG noise model is considered, most reconstruction methods rely on some approximations to PG statistics. Among them, the weighted least squares approximation is one of the most popular [Gre84, LSYZ15]. In existing works, the log-likelihood is approximated by the log-likelihood of a

Gaussian variable whose variance depends on the data model. The data fit term is then a sum of a weighted least squares plus a logarithm term, the latter being often omitted in order to ensure the convexity of the problem, and to simplify it. Recently in [DLZF18], the authors proposed to keep this term and to handle the resulting nonconvex optimization problem. The latter problem was addressed by employing the alternating direction method of multipliers. However, in this nonconvex setting, it provides only a guarantee to return a local minimizer. We propose in this section a new rational approximation to the PG data fidelity term and then show how to use our methodology from Chapter 4 to globally optimize the corresponding MAP problem and reconstruct the original signal.

5.2.1 Considered model and optimization problem

Observation model

We consider the reconstruction of a discrete positive signal $\overline{\mathbf{x}}$ of length T from an observation vector $\mathbf{y}$. Similarly to the Model (4.1), the signal $\overline{\mathbf{x}}$ is first degraded by a linear operator represented by a matrix with positive coefficients $\mathbf{H}$ and then by a scalar nonlinear function ϕ which takes positive values. Finally, the output signal is corrupted with a PG noise. For instance, matrix $\mathbf{H}$ can model the convolution with a filter impulse response, while ϕ can represent saturation effects such as clipping [TK19, JGX19]. Our model hence reads

$$(\forall t \in [\![1, T]\!]) \quad y_t \sim \mathcal{N}\Big(c + \mathcal{P}(\phi\big((\mathbf{H}\overline{\mathbf{x}})_t\big)), \sigma^2\Big) \,, \tag{5.1}$$

where $(\mathbf{H}\overline{\mathbf{x}})_t$ denotes the t-th component of the vector $\mathbf{H}\overline{\mathbf{x}}$, $\mathcal{P}$ and $\mathcal{N}$ denote respectively Poisson and Gauss distributions, c is a nonnegative constant modelling the average background noise, and σ^2 is the variance of the independent and identically distributed Gaussian noise.

To reconstruct the original signal $\overline{\mathbf{x}}$, a MAP estimator $\hat{\mathbf{x}}$ is computed, which amounts to minimizing the sum of the negative log-likelihood $f_{\mathbf{y}}$ plus a regularization $\mathcal{R}_\lambda$ balanced by a parameter $\lambda > 0$

$$\hat{\mathbf{x}} = \operatorname*{arg\,min}_{\mathbf{x} \in \mathbb{R}^T} f_{\mathbf{y}}(\mathbf{x}) + \mathcal{R}_\lambda(\mathbf{x}) \,. \tag{5.2}$$

Likelihood fit term

The likelihood of PG Model (5.1) is given by

$$\prod_{t=1}^{T} \left(\sum_{n=0}^{+\infty} \frac{e^{-\phi((\mathbf{H}\mathbf{x})_t)} (\phi((\mathbf{H}\mathbf{x})_t))^n}{n!} \frac{e^{-\left(\frac{y_t - n - c}{\sqrt{2}\sigma}\right)^2}}{\sqrt{2\pi}\sigma} \right) \,. \tag{5.3}$$

The corresponding log-likelihood is intricate and surrogates are often used for $f_{\mathbf{y}}$.

When ϕ is the identity function, classical approaches often approximate the negative logarithm of (5.3) with a more tractable function of the form $\sum_{t=1}^{T} g((\mathbf{H}\overline{\mathbf{x}})_t, y_t)$ [LSYZ15, CJPT15, MZCP17, LLK18]. Such a good approximation is the Weighted least squares with a logarithm term (WLOG):

$$g_{\text{wlog}}(x, y) = \frac{1}{2} \frac{(y - x)^2}{x + \sigma^2} + \frac{1}{2} \log(x + \sigma^2) \,.$$

Sparse regularization

The regularization $\mathcal{R}_\lambda$ is chosen according to the prior information available on $\overline{\mathbf{x}}$. We suppose here that, after a suitable linear transformation $\mathbf{D}$, $\overline{\mathbf{x}}$ is a sparse signal, i.e. only a few of its components are nonzero. This extends our work in Chapter 4 where the signal itself was composed of only a few spikes and the operator $\mathbf{D}$ was restricted to the identity. Alternatively, $\mathbf{D}$ can be a gradient operator for a signal with sharp discontinuities. The ℓ_0 regularization function is an effective way to enforce the sparsity of the solution since it penalizes equally the nonzero components of the vector $\mathbf{D}\overline{\mathbf{x}}$. Therefore, we use again a tight continuous approximation of it such as the ones described in Section 4.2.2. However, the pre-composition with the operator $\mathbf{D}$, which is introduced here, adds an extra flexibility to the criterion. Finally, in the following, to build our MAP estimator, we consider the optimization Problem (5.2) with

$$f_{\mathbf{y}}(\mathbf{x}) = \sum_{t=1}^{T} g_{\text{wlog}}(\phi((\mathbf{H}\mathbf{x})_t) + c, y_t) \tag{5.4}$$

$$\mathcal{R}_\lambda(\mathbf{x}) = \sum_{t=1}^{T} \Psi_\lambda((\mathbf{D}\mathbf{x})_t)\,, \tag{5.5}$$

where Ψ_λ is a continuous approximation such as the ones given in Section 4.2.2.

5.2.2 Rational formulation

In order to apply our methodology from Chapter 3, we reformulate Problem (5.2) into a rational problem. The approximation to the ℓ_0 function is handled as described in the Section 4.3.2 Furthermore, we substitute a rational approximation $\widehat{\log}$ for the log function in g_{wlog}. An example of a good rational approximation for the log function is given by the following Padé approximant [DB08]

$$(\forall x \in \mathbb{R}_+^*) \quad \widehat{\log}(x) = (x-1)\frac{x+5}{4x+2}\,, \tag{5.6}$$

such that, for all x in $\mathbb{R}_+^*$, $\log(x) \leq \widehat{\log}(x)$. This approximation is satisfactory on a broad interval and especially accurate on $[0.2, 3.9]$ where the relative error is less than 8%.

Substituting the log function with its rational approximation $\widehat{\log}$, Equation (5.4) becomes a sum of rational functions. As an example, we express Problem (5.2) as a rational optimization problem in the case of interest where $\mathbf{D}$ is the discrete gradient operator and Ψ_λ is the capped ℓ_1 regularization:

$$\begin{aligned} \min_{\mathbf{x},\mathbf{z},\mathbf{r}} \quad & \hat{f}_{\mathbf{y}}(\mathbf{x}) + \sum_{t=1}^{T-1} (r_t(1-z_t) + \lambda z_t) \\ \text{s.t.} \quad & (\forall t \in [\![1, T-1]\!]) \begin{cases} z_t = z_t^2 \\ (z_t - 1/2)(r_t - \lambda) \geq 0 \\ r_t \geq 0 \\ r_t^2 = (x_{t+1} - x_t)^2\,, \end{cases} \end{aligned} \tag{5.7}$$

where $\hat{f}_{\mathbf{y}}$ is the function in (5.4) with the log term replaced by $\widehat{\log}$, $(z_t)_{t\in[\![1,T-1]\!]}$ and $(r_t)_{t\in[\![1,T-1]\!]}$ are extra real-valued variables used to handle respectively the indicator function and the absolute value of the capped ℓ_1 function. Problem (5.7) is then solved using the method from Chapter 3.

Note that the operator $\mathbf{D}$ imposes extra complexity to (5.7). Instead of having a polynomial in the three variables (x_t, z_t, r_t) as in Chapter 4, we now have a polynomial in four variables (x_{t+1}, x_t, z_t, r_t). The choice of the operator $\mathbf{D}$ is thereby important for the complexity of the model: using a higher-order discrete difference operator indeed leads to an increase in the number of variables per polynomial.

5.2.3 Numerical simulations

Throughout this part, $\mathbf{H}$ is a Toeplitz matrix corresponding to the convolution with the lowpass filter with impulse response $\mathbf{h} = [0.25, 0.5, 0.25]$ (except in Table 5.3) and with convolution boundaries using zero padding. The variance σ^2 of the Gaussian noise is set to 0.15 (except in Table 5.2) and c to 0. We solve Problem (5.2) using (5.4) and (5.6) for the data fit term together with a SCAD regularization (with parameter equal to 2.1) in (5.5). We use the software GloptiPoly [HLL09] to perform the relaxation of the rational problem into SDP problems which are then solved using SDPT3 [TTT99]. We set the relaxation order in the SDP hierarchy to 3, i.e. we consider moments up to degree 6. We compare our method to proximal methods applied on the following convex problem:

$$\underset{\mathbf{x}\in\mathbb{R}^T}{\text{minimize}} \sum_{t=1}^{T} g((\mathbf{H}\mathbf{x})_t, y_t) + \lambda \ell_1(\mathbf{D}\mathbf{x}) \;, \tag{5.8}$$

where g is one of the common convex approximations to the negative log-likelihood mentioned in Section 5.2.1. We use the Peak Signal to Noise Ratio (PSNR) between the original signal $\overline{\mathbf{x}}$ and the estimator $\hat{\mathbf{x}}$ to assess the quality of the reconstruction.

5.2.3.1 Sparse signals reconstruction

The linear operator $\mathbf{D}$ is first set to identity. We performed simulations on 100 randomly generated sparse signals of length $T = 200$ with 20 nonzero elements. The positions of the latter are drawn uniformly between 1 and T.

Linear case

We first consider the case when $\phi = \mathrm{Id}$. We compare our method to the classic FB applied to Problem (5.8) as the considered function g is differentiable. More precisely, we compare our method with the results obtained with FB applied for the four approximations Generalized Anscombe Transform (GAST), Weighted Least Squares (WLS), WLOG, and SPOI (Shifted Poisson) [CJPT15, MZCP17] of the likelihood (5.3). The value of the regularization parameter λ has been tuned empirically to 5.5 for our method and 0.5, 2.5, 0.9 and 1 for FB respectively on GAST, WLS, WLOG, and SPOI.

Table 5.1 shows statistics on the PSNR of the estimated signal over 100 runs for each tested methods. We set the maximum number of iterations to 10000 for FB. Furthermore, we show in Table 5.2 the impact of the variance σ^2 on the quality of the signal reconstructed with our method.

We finally show that our results hold for different impulse responses $\mathbf{h}$. We drew 50 different signals $\overline{\mathbf{x}}$ randomly of length $T = 200$ with 20 nonzero elements as previous. We also drew 4 filter finite impulse responses $\mathbf{h}$ of length 3 following a standard normal distribution. The impulse responses were then normalized to sum up to 1. We give the expression of the obtained vectors $\mathbf{h}$:

$$\mathbf{h}_1 = \begin{pmatrix} 0.15 \\ 0.53 \\ 0.32 \end{pmatrix} \;,\; \mathbf{h}_2 = \begin{pmatrix} 0.39 \\ 0.54 \\ 0.07 \end{pmatrix} \;,\; \mathbf{h}_3 = \begin{pmatrix} 0.15 \\ 0.23 \\ 0.62 \end{pmatrix} \;,\; \mathbf{h}_4 = \begin{pmatrix} 0.67 \\ 0.12 \\ 0.21 \end{pmatrix} \;.$$

Table 5.1: *Statistics on the PSNR (in dB) between the original sparse signal and the estimated signal (100 realizations).*

	Average	Median	Minimum	Maximum
FB (GAST)	12.0	12.0	10.5	13.8
FB (WLS)	7.3	7.1	4.1	10.9
FB (WLOG)	8.8	8.7	7.1	11.8
FB (SPOI)	10.7	11.2	6.9	13.5
Our method	12.5	12.5	11.3	14.1

Table 5.2: *Impact of the variance σ^2 on the reconstruction quality*

σ^2	0.15	0.25	0.5	0.85
Average PSNR on 100 realizations (in dB)	12.5	11.3	10.5	8.1

For each filter, we generated an observation vector $\mathbf{y}$ for each signal $\overline{\mathbf{x}}$ and we applied our algorithm to obtain an estimate of $\overline{\mathbf{x}}$. Table 5.3 shows statistics on the PSNR between the original signal and its estimate. We observe that the results are similar for each impulse response and are also similar to the ones reported with the impulse response $\mathbf{h} = [0.25, 0.5, 0.25]$.

Table 5.3: *Statistics on the PSNR (in dB) between the original sparse signal and the estimated signal for four different filters (50 realizations)*

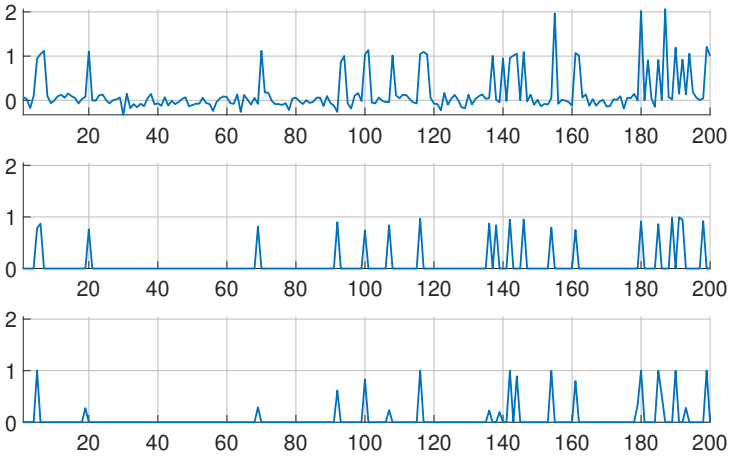
	Average	Median	Minimum	Maximum
$\mathbf{h}_1$	12.1	12.1	11.2	13.7
$\mathbf{h}_1$	12.4	12.3	11.2	13.4
$\mathbf{h}_1$	12.7	12.7	10.8	13.6
$\mathbf{h}_1$	12.8	12.9	11.3	14.4

Nonlinear case

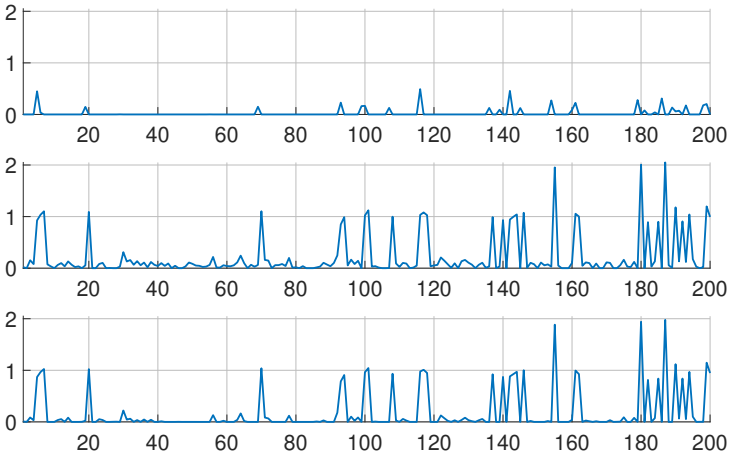
We now choose $\phi(x) = \frac{x}{\delta+|x|}$ with $\delta = 0.3$. Since ϕ is nonconvex, we linearize it around 0 in (5.8) in order to apply FB. Figure 5.1 shows an example of the reconstruction in the nonlinear case where our method performs better than FB. Moreover, we observe that in both linear and nonlinear cases, the convergence in the SDP hierarchy occurs and that our method returns a global solution to (5.2). Indeed, the relative gap between the lower bound returned by solving the SDP problem of order 3 and the value of the criterion in (5.2) is in the order of magnitude of 10^{-6}.

Impact of logarithmic term

In this paragraph, we study the impact of the log term by dropping it in (5.4) and compare it with the previous criterion. We use the same methodology developed in Section 5.2.2 and the same experimental settings of Section 5.2.3.1 in the nonlinear case.



(a) *From top to bottom: the observed signal* $\mathbf{y}$*, the original signal* $\overline{\mathbf{x}}$*, and the signal reconstructed with our method.*



(b) *From top to bottom, the estimator returned by FB for respectively GAST, WLS, and SPOI approximations.*

Figure 5.1: *Example of sparse signal reconstruction with PG noise and a nonlinear model.*

In Figure 5.2, we zoom in the first 60 samples of the signals to compare them. We observe that without the log term, the WLS approximation always overestimate the amplitudes of the peaks. This is consistent with [LLK18] where the authors evidence that WLS approximation, as well as SPOI and GAST, introduce bias since the observations $\mathbf{y}$ must be nonnegative. Conversely, we also note that after adding the log term, the amplitude of the peaks are slightly underestimated. However, many small artifacts are removed. Table 5.4 shows that adding the log term results in an 1.4 dB PSNR improvement in average.

5.2.3.2 Visible light communication signal reconstruction

We now observe $\mathbf{y}$ according to Model (5.1) when $\phi(x) = \frac{x}{\delta+|x|}$ and $\mathbf{D}$ is the discrete gradient operator. We consider 100 binary signals $\overline{\mathbf{x}}$ such that there are only 20 transitions between the 0 and 1 states. The location of the transitions are drawn uniformly

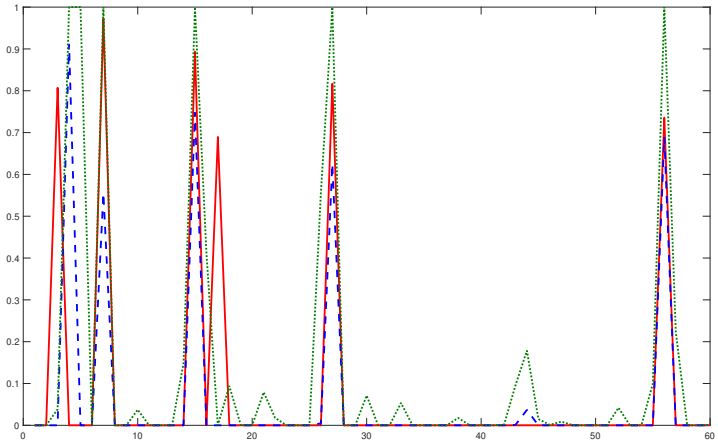


Figure 5.2: *Comparison of the reconstructed signal using our method. In red, the original signal* $\overline{\mathbf{x}}$*, in dashed blue, the signal reconstructed using* g_{wlog}*, and in dotted green, the result by using* g_{wls}*.*

Table 5.4: *Comparison on the PSNR (in dB) for WLS and WLOG approximations using our method (100 random realizations).*

	Average	Median	Minimum	Maximum
WLS	11.0	10.8	9.2	13.4
WLOG	12.0	11.9	10.5	15.1

between 1 and T. This model is inspired from VLC where a digital signal is transmitted by a LED that is alternatively switching between high and low intensity [CHYT17].

As the proximal operator of $\ell_1 \circ \mathbf{D}$ does not have a closed form, our method is now compared to the forward-backward primal-dual algorithm (FBPD) [KP15] instead of FB. Moreover, we use the GAST approximation for the log-likelihood in FBPD as it gives the best reconstruction results. Algorithm 1 shows the final FBPD algorithm where sgn denotes the sign function, max the element-wise maximum, $\boldsymbol{\lambda}$ the vector composed of solely λ, $\beta = \max_{t\in[\![1,T]\!]} \beta_t$ the overall Lipschitz constant of the GAST approximation, and $\mathbf{d}$ the vector $(g'((\mathbf{Hx})_t, y_t))_{t\in[\![1,T]\!]}$. We recall the expression of g, g', and β_t for the GAST approximation [MZCP17]:

$$
\begin{aligned}
g(x, y_t) &= 2\Big(\sqrt{y_t + \sigma^2 + 3/8} - \sqrt{x + \sigma^2 + 3/8}\Big)^2 \\
g'(x, y_t) &= 2 - 2\frac{\sqrt{y_t + \sigma^2 + 3/8}}{\sqrt{x + \sigma^2 + 3/8}} \\
\beta_t &= \left(\frac{3}{8} + \sigma^2\right)^{-3/2} \sqrt{y_t + \sigma^2 + \frac{3}{8}}\,.
\end{aligned}
$$

Note that the last line of Algorithm 1 is actually the expression of $\operatorname{prox}_{\theta(\lambda\ell_1)^*}(\overline{\mathbf{z}})$ according to Moreau's formula (cf. Proposition 5 in Appendix B).

The value of the regularization parameters λ is tuned empirically to 2 for our method and to 1.1 for FBPD. Figure 5.3 illustrates the obtained signals for a single test. We observe that our method provides a better estimator of the original signal. This is

confirmed in Table 5.5 that shows different statistics on the PSNR between the original signal $\overline{\mathbf{x}}$ and the estimated one $\hat{\mathbf{x}}$.

Algorithm 1: Forward-backward primal-dual algorithm to solve (5.8)

Input: Set $(\mathbf{x}^{(0)}, \mathbf{z}^{(0)}) \in \mathbb{R}^T \times \mathbb{R}^{T-1}$

Input: Set $(\tau, \theta) \in \mathbb{R}_+ \times \mathbb{R}_+$ such that $\tau\left(\frac{\beta}{2} + \theta \left\|\mathbf{D}^\top \mathbf{D}\right\|\right) < 1$

Output: $\mathbf{x}$, solution of (5.8)

for $k = 0,1,\ldots$ **do**

$\mathbf{x}^{(k+1)} \leftarrow \mathbf{x}^{(k)} - \tau(\mathbf{H}^\top \mathbf{d} + \mathbf{D}^\top \mathbf{z}^{(k)})$;

$\overline{\mathbf{z}} \leftarrow \mathbf{z}^{(k)} + \theta \mathbf{D}(2\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)})$;

$\mathbf{z}^{(k+1)} \leftarrow \overline{\mathbf{z}} - \theta \max(\theta^{-1}(\overline{\mathbf{z}} - \boldsymbol{\lambda}), \mathbf{0}) \odot \operatorname{sgn}(\overline{\mathbf{z}})$;

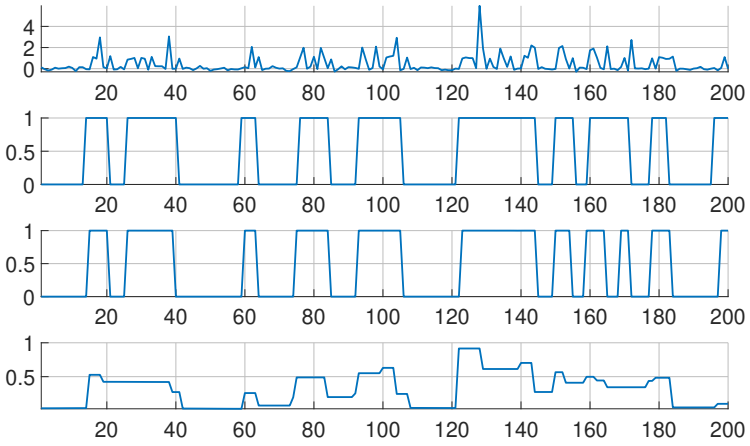


Figure 5.3: *Example of VLC signals reconstruction. From top to bottom: the observed signal* $\mathbf{y}$*, the original signal* $\overline{\mathbf{x}}$*, the estimator obtained with our method, and the estimator returned by FBPD.*

Table 5.5: *Statistics on PSNR (in dB) between the original VLC signal and the estimated signal (100 realizations).*

	Average	Median	Minimum	Maximum
FBPD	8.16	8.18	5.87	9.94
Our method	9.20	9.12	6.38	12.22

§ 5.3 Impulse noise

In many applications, the noise is modelled by a zero-mean normal distribution which is added to the noiseless signal as in Chapter 4. As a consequence, minimization of a mean square error appears as a ubiquitous technique in signal processing. However, in practice, it is common that outliers are present in the observations, so altering the noise

distribution. This results in poor performance of least squares based estimators. Indeed, outliers produce large errors making their corresponding weight prevalent in least squares fitting. Consequently, even very few of them can significantly decrease the performance of the estimator.

With the emergence of big data, manually discarding outliers is not a suitable solution. Moreover, it can be difficult to decide which data are outliers, especially in high dimensional problems. Hence, many robust fit functions have been proposed in order to reduce the impact of outliers on the estimate.

A standard approach in robust estimation is to cap the ℓ_2 function in order to keep the least squares behavior for small error values around zero and to apply a constant term in order to penalize equally errors and outliers above a given threshold [LHY16]. However, the convexity of the fit function is lost and the resulting optimization problem becomes intricate. A convex surrogate is the ℓ_1 norm, or a smoothed version of it, which reduces the influence of the outliers as it gives a steadily increasing weight to the errors [Nik02]. Nonetheless, it results in a shrinkage of low errors towards 0, which is not desirable as it induces biased estimates. In order to keep benefits both of least squares for low errors and of the least absolute values for high errors, the Huber function has been proposed [Hub64]. The latter also has the advantage of being convex, which is an enjoyable property for optimization. Smoother version of Huber functions are also used such as the pseudo-Huber function [CBFAB94]. Other M-estimators have been proposed such as Tukey's function for instance [MMYSB19]. Furthermore, transpositions of robust estimators for sparse signals estimation [ZKOM18] and for multivariate signals [MMYSB19, DP18] have also been proposed.

Exploiting the properties of the original signal is an important feature in inverse problems, that can also contribute to improve robustness. For instance, in the previous chapters, we considered that the original signal was sparse of sparse under a linear transformation. Here, we concentrate on an assumption corresponding to a union of subsets model. Such a model has been a topic of interest in signal processing, especially when the subsets are affine spaces. For instance, it has appeared in compressed sensing where one wants to reconstruct a signal having only a given number of nonzero components from linear observations [Blu11]. Other examples that can be expressed as union of subspaces include reconstruction of a stream of Dirac impulses where both the location and the amplitude are unknown, determination of overlapping echoes with unknown delay and amplitude, or reconstruction of a signal whose Fourier transform is known to be located in a union of sub-bands [LD08]. Union of subspaces are also useful in matrix completion to express nonlinear connections between elements [OWNB17].

Working in a union of subsets often leads to a challenging problem as linearity and convexity are lost. By making additional assumptions on the shapes of the subspaces or on the model, some methods have been shown to be successful in solving specific problems [EM09, Blu11, HIS16, AH18]. Nevertheless, more general forms are still difficult to solve and the proposed methods do no extend easily to the union of general subsets.

A key observation is that many unions of subsets, can be expressed as polynomial constraints. Similarly, many robust fit function, such as the capped ℓ_2 function, are piecewise polynomial. The versatility of rational functions thus allows us to encompass both contexts together. In the remaining of this chapter, we address a problem of robust signal reconstruction on a union of subsets. We add extra nonconvex constraints that force the amplitude of the signal to be greater than a threshold or identically zero. Although, this constraint is usually difficult to handle, it can be addressed under the form of polynomials constraints. We propose to reformulate this nonconvex problem on a union of subsets as a rational optimization problem that is then solved by using the method of Chapter 3 in order to guarantee global optimality.

5.3.1 Observation model

Model with outliers

Similarly to the model of Chapter 4, we first consider a degraded version $\overline{\mathbf{y}}$ of an original signal $\overline{\mathbf{x}}$ of size T such that

$$\overline{\mathbf{y}} = \phi(\mathbf{H}\overline{\mathbf{x}}) + \mathbf{w}\,, \tag{5.9}$$

where $\mathbf{H}$ is a $T \times T$ matrix corresponding to a linear operator, $\mathbf{w}$ is a zero-mean white Gaussian noise, and ϕ is a rational function, i.e. a ratio of two polynomials, that acts component-wise.

The novelty of this section is to add a significant perturbation on some of the samples of $\mathbf{y}$. This models for instance the possibility of a sensor malfunction. It follows that the observation vector includes a certain number of outliers, i.e. values that differ significantly from the others. Finally, the observation model becomes

$$(\forall t \in [\![1, T]\!]) \quad y_t = \begin{cases} \overline{y}_t & \text{with probability } 1-\delta\,, \\ \overline{y}_t + n_t & \text{with probability } \delta\,. \end{cases}$$

In the above equation, $\overline{\mathbf{y}}$ comes from Model (5.9), δ is a small real between 0 and 1, and n_t is the realization of a noise with high amplitude compared to values of $\overline{\mathbf{y}}$.

Signal model as a union of subsets

Similarly to many methods for inverse problem, ours relies on both data fidelity and some assumptions about the original signal $\overline{\mathbf{x}}$. We consider the following prior knowledge on $\overline{\mathbf{x}}$: its components are either zero or have absolute value above a given positive threshold λ, i.e.

$$(\forall t \in [\![1, T]\!]) \quad x_t = 0 \quad \text{or} \quad |x_t| \geq \lambda\,. \tag{5.10}$$

Constraints (5.10) imply that each sample of the signal $\overline{\mathbf{x}}$ belongs to the union of three convex subsets, namely $\{0\} \cup]-\infty, -\lambda] \cup [\lambda, +\infty[$. The benefit of such a model is to reduce the space where we search for a solution. Let us emphasize that the union we consider here is composed of subsets that are not necessarily linear subspaces. Constraints (5.10) are nonconvex and thus result in a difficult optimization problem.

Our assumption on the signal model may be related to the standard union of subspaces approach from compressed sensing, where the dimension of the subspaces is low compared to the underlying space dimension. This accounts for sparsity. On the other hand, the union of subspaces model does not necessarily restrict the number of non-zero components nor impose sparsity on the components [BNFR19]. Hence, the method we propose can be applied to recover sparse signals as well as dense signals in the sense that only a few elements are null.

5.3.2 Problem formulation

5.3.2.1 Robust reconstruction criteria

To reconstruct the original signal $\overline{\mathbf{x}}$, we minimize a criterion matching the data to the model. A classic approach for dealing with Model (5.9) is to minimize the mean square error $\|\mathbf{y} - \phi(\mathbf{H}\mathbf{x})\|^2$ with respect to $\mathbf{x}$. However, the presence of outliers skews the solutions. To tackle this issue, several robust surrogates for the squared norm have been proposed. The latter are all written under the form $\sum_{t=1}^{T} \Psi_\theta(y_t - \phi((\mathbf{H}\mathbf{x})_t))$ where the

function $\Psi_\theta : \mathbb{R} \to \mathbb{R}$ may depend on a positive real parameter θ. We obtain the following criteria:

- the previously mentionned ℓ_2 norm: $\Psi_\theta(x) = x^2$,
- the ℓ_1 norm: $\Psi_\theta(x) = |x|$,
- Huber function [Hub64]:

$$\Psi_\theta(x) = \frac{1}{2}x^2 \mathbb{1}_{\{|x| \leq \theta\}}(x) + \theta\left(|x| - \frac{1}{2}\theta\right)\mathbb{1}_{\{|x|>\theta\}}(x)\,,$$

- the pseudo-Huber function [CBFAB94]: $\Psi_\theta(x) = \sqrt{x^2 + \theta^2}$,
- and the capped ℓ_2 function:

$$\Psi_\theta(x) = \mathbb{1}_{\{|x| \geq 1/\sqrt{\theta}\}}(x) + \theta x^2 \mathbb{1}_{\{|x|<1/\sqrt{\theta}\}}(x)\,.$$

Figure 5.4 displays the considered robust fit functions Ψ_θ. The capped ℓ_2 and Huber functions aim at reducing the impact of the outliers by decreasing the penalization on the high errors while preserving mean squares on the low errors. The parameter θ represents here the threshold value on the error between these two penalties. It must be set carefully depending on the level of noise $\mathbf{w}$ and of the outliers.

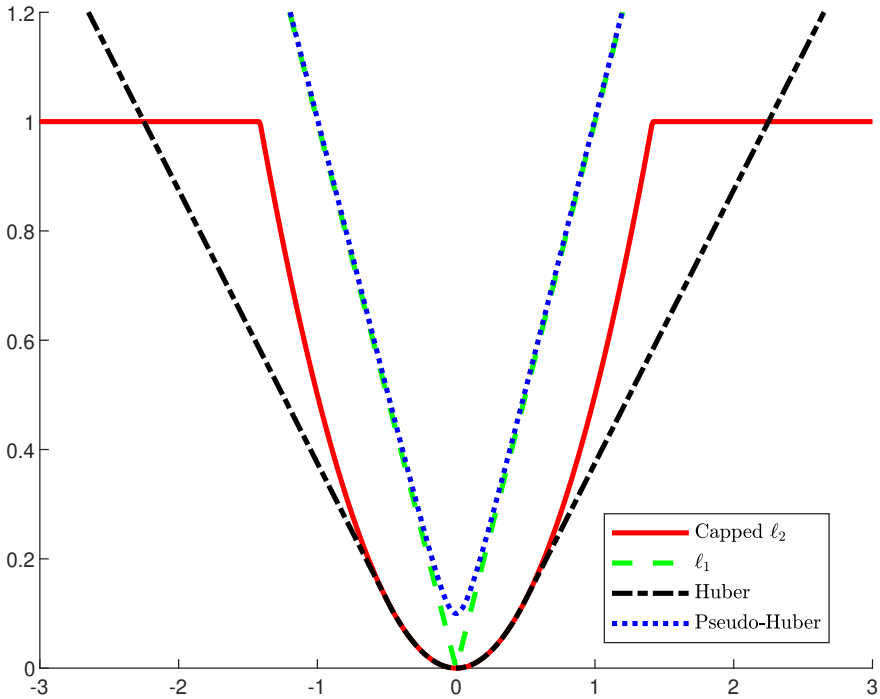


Figure 5.4: *Considered robust fit functions Ψ_θ ($\theta = 0.5$ for the capped ℓ_2 and Huber functions, $\theta = 0.1$ for the pseudo-Huber function).*

Considering the fit criterion to be minimized and Constraints (5.10) simultaneously, we finally obtain the following optimization problem:

$$\begin{aligned} &\underset{\mathbf{x}\in\mathbb{R}^T}{\text{minimize}} && \sum_{t=1}^{T} \Psi_\theta(y_t - \phi((\mathbf{H}\mathbf{x})_t)) \\ &\text{s.t.} && (\forall t \in [\![1, T]\!]) \quad x_t = 0 \text{ or } |x_t| \geq \lambda\,, \end{aligned} \tag{5.11}$$

where $(\mathbf{Hx})_t$ denotes the t-th component of the vector $\mathbf{Hx}$. Problem (5.11) is difficult since the constraints are nonconvex. Moreover, the objective function is also nonconvex for the capped ℓ_2 fit. However, we propose to use the methodology of Chapter 4 to solve Problem (5.11) for any above choice of the function Ψ_θ. We hence reformulate the optimization problem as a rational one before solving it. We start with the reformulation of the constraints.

5.3.2.2 Reformulation as a rational optimization problem

Constraints expressed as polynomial inequalities

The constraints in (5.11) can be expressed as polynomials ones by introducing extra binary variables ζ_t for all t in $[\![1, T]\!]$. We can indeed substitute the product $\zeta_t x_t$ for x_t in (5.11) and, letting $\zeta_t = 0$ account for vanishing x_t, the constraints in (5.11) reduce to the inequalities $|x_t| \geq \lambda$ for all t in $[\![1, T]\!]$. Finally, binary values can be imposed by the polynomial constraints $\zeta_t = \zeta_t^2$ for all t in $[\![1, T]\!]$. We thus obtain the following polynomial optimization problem

$$\begin{array}{lll} \underset{(\mathbf{x},\boldsymbol{\zeta})\in\mathbb{R}^T\times\mathbb{R}^T}{\text{minimize}} & \displaystyle\sum_{t=1}^{T} \Psi_\theta(y_t - \phi((\mathbf{H}(\mathbf{x}\odot\boldsymbol{\zeta}))_t)) & \\ \text{s.t.} & (\forall t \in [\![1, T]\!]) \quad |x_t| \geq \lambda & \\ & (\forall t \in [\![1, T]\!]) \quad \zeta_t = \zeta_t^2\,. & \end{array} \tag{5.12}$$

Note that the constraints $|x_t| \geq \lambda$ are still nonconvex and make Problem (5.12) difficult to solve with standard methods. However, under the transformation of the objective function of the next paragraph, it leads to the minimization of a polynomial function subject to polynomial constraints. Therefore, it can be solved with the tools of Chapter 3.

Nevertheless this formulation doubles the number of optimization variables and similarly to the example in Section 4.3.1, Problem (5.12) cannot be solved by state-of-the-art polynomial optimization solvers in a fair amount of time, even for signals of small size T. Instead, we suggest to relax the equality constraints in (5.10) into an inequality and we obtain the following constraints

$$(\forall t \in [\![1, T]\!]) \quad |x_t| \leq \epsilon \text{ or } |x_t| \geq \lambda\,, \tag{5.13}$$

where $\epsilon < \lambda$ is a small positive real. Notice that this constraint can also be used to expressed signals that are in two different bands as in [LD08] for instance. We now write the above constraints as polynomial inequalities and we obtain

$$\begin{array}{ll} (\forall t \in [\![1, T]\!]) & (\epsilon - r_t)(\lambda - r_t) \geq 0 \\ (\forall t \in [\![1, T]\!]) & r_t^2 = x_t^2\,,\ r_t \geq 0\,. \end{array}$$

The absolute value in the constraints of (5.13) is handled by adding the extra variable $\mathbf{r}$ as detailed in Section 4.3.3.

Objective function expressed as a rational function

In the following, we study the reformulation for the capped ℓ_2 function since it is the most challenging case. The method can be easily adapted for other fit functions from Section 5.3.2.1.

As witnessed by the characteristic functions that appear in the definition of the capped ℓ_2 function, the objective in (5.11) is piecewise polynomial. Following Section 4.3.2, a characteristic function can be replaced by a binary variable z that takes

identical values. For instance, if Ψ_θ is the capped ℓ_2 function as defined in Section 5.3.2.1, we introduce a binary variable z which takes value 0 when $|x|$ is smaller than $1/\sqrt{\theta}$, and 1 otherwise. This can be written as

$$(z - 1/2)\left(|x| - 1/\sqrt{\theta}\right) \geq 0\,. \tag{5.14}$$

Substituting similarly the characteristic functions for all t in $[\![1, T]\!]$ and the original constraints in Problem (5.11), we finally obtain the following rational optimization problem

$$\begin{aligned} \underset{(\mathbf{x},\mathbf{r},\mathbf{v},\mathbf{z})\in\mathbb{R}^{4T}}{\text{minimize}} \quad & \sum_{t=1}^{T} \; z_t + (1 - z_t)\theta(y_t - \phi((\mathbf{H}\mathbf{x})_t))^2 \\ \text{s.t. } (\forall t \in [\![1, T]\!]) \quad & (\epsilon - r_t)(\lambda - r_t) \geq 0 \\ & r_t^2 = x_t^2\,,\; r_t \geq 0 \\ & (z_t - 1/2)\left(v_t - 1/\sqrt{\theta}\right) \geq 0 \\ & v_t^2 = (y_t - \phi((\mathbf{H}\mathbf{x})_t))^2\,,\; v_t \geq 0 \\ & z_t = z_t^2\,, \end{aligned} \tag{5.15}$$

where the extra variable $\mathbf{v}$ is used to handle the absolute value in Constraint (5.14). The objective function is now a rational function in $\mathbf{x}$ and $\mathbf{z}$ while the constraints are polynomial inequalities in $\mathbf{x}$, $\mathbf{r}$, $\mathbf{v}$, and $\mathbf{z}$. Problem (5.15) can then be solved using the method from Chapter 3.

Note that the radical in the pseudo-Huber function can be handled similarly to the absolute value. Indeed, we can replace $\sqrt{x^2 + \theta^2}$ by an additional variable u and add the polynomial constraints

$$\begin{cases} u^2 = x^2 + \theta^2 \\ u \geq 0\,. \end{cases}$$

5.3.3 Numerical results

For each test, we generate an initial signal $\overline{\mathbf{x}}$ of size $T = 50$ satisfying Constraints (5.10): a given percentage of elements of $\overline{\mathbf{x}}$, called the degree of sparsity, chosen randomly are set to 0 while the others are set to have their absolute value uniformly selected between $\lambda = 0.7$ and 1. We choose $\mathbf{H}$ as a convolution matrix associated to a finite impulse response filter $\mathbf{h}$ of length 3 whose elements are drawn uniformly in $[0, 1]$, i.e. $\mathbf{H}$ is Toeplitz-band with $\mathbf{h}$ defining the elements on the band. The white noise $\mathbf{w}$ has a standard deviation of 0.15. The vector $\mathbf{y}$ contains 1% of outliers and their location have been drawn randomly with equal probability among the T samples. The impulse noise $\mathbf{n}$ has a fixed amplitude set to twice the maximum of $\overline{\mathbf{y}}$ and a random sign with equal probability. The parameter ϵ is set to 10^{-3} and the value of θ for the capped ℓ_2 and Huber functions is tuned to 0.4. The nonlinearity ϕ is chosen as a saturation expressed by the rational function

$$(\forall x \in \mathbb{R}) \quad \phi(x) = \frac{x}{0.3 + |x|}\,.$$

We use GloptiPoly [HLL09] together with the SDP solver SDPT3 [TTT99] to solve the rational problem. In all our simulations, we compute the order 3 SDP relaxation of Lasserre's hierarchy. The problem is formulated using the approach of [CPM19] in order to use the structure of $\mathbf{H}$ and reduce the computational burden. All the simulations have been run on an Intel i7 CPU running at 1.90 GHz with 16 GB of RAM.

We compare our approach for the different fit functions listed in Section 5.3.2.1. We run 50 simulations and compare the different relative errors $\|\overline{\mathbf{x}} - \hat{\mathbf{x}}\| / \|\overline{\mathbf{x}}\|$ between the initial signal $\overline{\mathbf{x}}$ and the signal $\hat{\mathbf{x}}$ estimated with our method.

5.3.3.1 Convergence of the SDP hierarchy

We first look at the convergence of Lasserre's hierarchy. For the ℓ_2, the capped ℓ_2, and Huber functions, the convergence has occurred at relaxation order 3. This is certified by the sufficient rank condition given in Section 3.1.3 and implemented in GloptiPoly, which additionally certifies that the optimal point is unique. This is an important feature since, in contrast to many nonconvex optimization methods, we have the theoretical guarantee that the obtained solutions are exactly the global minimizers of Problem (5.15).

On the other hand, for the ℓ_1 and the pseudo-Huber functions, the convergence does not always occur at order 3. This is shown for the ℓ_1 function in Figure 5.5 where we draw for the 50 tests, the obtained lower bound in plain blue and the value of the criterion at the computed solution in dotted red. We observe a gap between the two curves that shows that a higher relaxation order would be required for convergence of the hierarchy. Moreover, a consequence for these two fit functions is that the imposed constraints are not always satisfied by the candidate approximate optimal point.

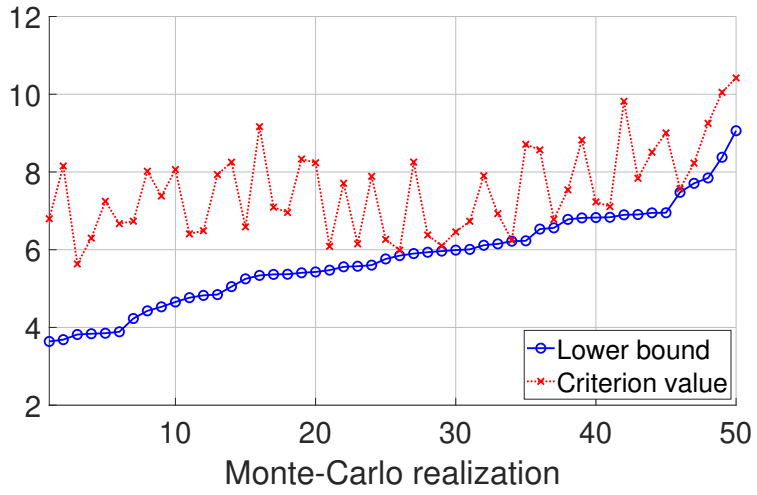


Figure 5.5: *Convergence study of Lasserre's hierarchy for the ℓ_1 fit function on 50 tests: In plain blue the value of the lower bound, in dotted red the value of the criterion at the solution. For the sake of clarity, the results are ordered according to the values of the lower bound.*

The convergence at a low relaxation order is important for the global minimum guarantee as well as for the applicability of the method by limiting the SDP to a fair size. Therefore in the following section, we focus our attention on the capped ℓ_2 and Huber functions against the ℓ_2 fit.

5.3.3.2 Comparison of the robust approach with least squares

We compare here the ℓ_2 with the capped ℓ_2 and Huber fit functions for three different degrees of sparsity of the original signal $\overline{\mathbf{x}}$. Table 5.6 shows the relative error for the different fit functions. We observe that both robust fit functions yield a smaller error than least squares but the capped ℓ_2 gives the smallest error. This confirms the interest of our methodology, which is able to deal with nonconvex penalty functions. Furthermore,

Table 5.6: *Statistics on the relative error between* $\overline{\mathbf{x}}$ *and* $\hat{\mathbf{x}}$ *with different degrees of sparsity for 50 tests.*

	Degree of sparsity	80%	50%	30%
ℓ_2	Average	0.87	0.66	0.54
	Median	0.85	0.66	0.55
Huber	Average	0.65	0.48	0.35
	Median	0.69	0.47	0.33
Capped ℓ_2	Average	0.43	0.36	0.30
	Median	0.34	0.34	0.28

similar results are observed for both sparse and dense signals. Table 5.7 shows both the True Positive Rate (TPR) and the False Positive Rate (FPR)for both ℓ_2 and the capped ℓ_2 functions. We use a threshold value of 0.1 for the peak detection. Note that since Constraints (5.10) are enforced when solving (5.15), the threshold value can be taken in $]\epsilon, \lambda[$ indifferently. We notice that for the capped ℓ_2 fit, the average TPR is close to 1, which means that almost all the peaks are well detected, and the FPR is close to zero, i.e. we do not detect peaks at samples originally equal to zero. This is in contrast with the results for the ℓ_2 fit.

Table 5.7: *Statistics on the TPR and FPR of peak detection between* $\overline{\mathbf{x}}$ *and* $\hat{\mathbf{x}}$ *with different degrees of sparsity for 50 tests.*

Fit function		Capped ℓ_2			ℓ_2		
Degree of sparsity		80%	50%	30%	80%	50%	30%
TPR	Average	0.96	0.94	0.94	0.84	0.81	0.82
	Median	1.00	0.93	0.95	0.83	0.80	0.83
FPR	Average	0.01	0.10	0.09	0.15	0.22	0.22
	Median	0.04	0.07	0.10	0.12	0.20	0.20

The poor results above for the ℓ_2 fit are due to the outliers together with the convolution matrix $\mathbf{H}$ which makes the estimation inaccurate in the neighborhood of the outliers as illustrated in Figure 5.6. The latter shows a comparison between a robust and a least squares recovery on a single test. The red curve represents the initial signal $\overline{\mathbf{x}}$ and the blue and green curves are the estimated signal using respectively the capped ℓ_2 and the ℓ_2 functions. For the capped ℓ_2 function, we observe that the locations of the non-zero samples are well recovered and the amplitudes are close to the ones of $\overline{\mathbf{x}}$. At locations far from any outliers, comparable results are observed for both the capped ℓ_2 and ℓ_2 fit functions. However, close to an outlier value, the estimated signal is prone to many errors using ℓ_2 in contrast with its robust counterpart. This illustrates the benefit of the capped ℓ_2 for robust reconstruction.

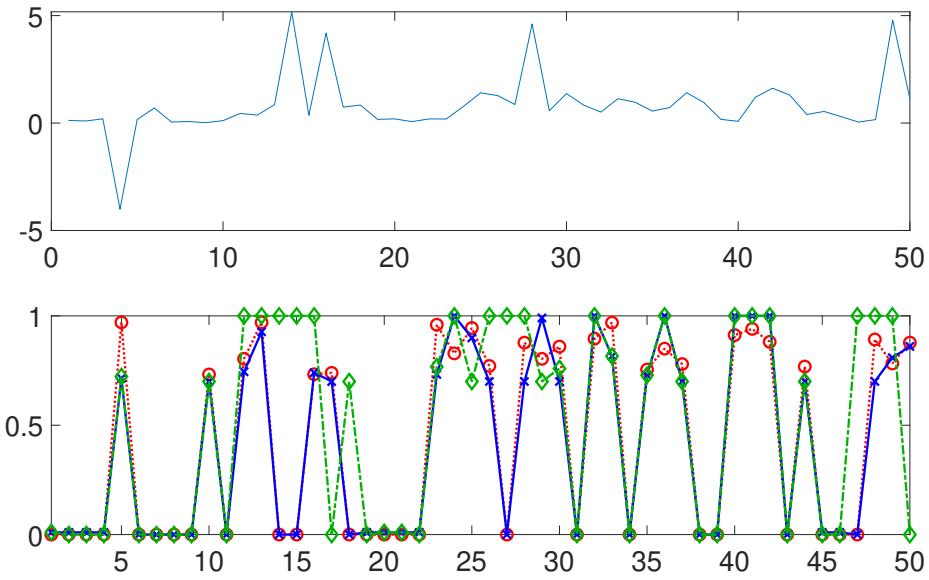


Figure 5.6: *Reconstruction of a signal with degree of sparsity of 50%. Top: the corrupted observation* $\mathbf{y}$*, Bottom: in red circle dotted curve, the initial signal* $\overline{\mathbf{x}}$ *and respectively in blue cross plain curve and green dotted curve, the estimated signal* $\hat{\mathbf{x}}$ *using the capped* ℓ_2 *and the* ℓ_2 *fit functions.*

§ 5.4 Summary

In this chapter, we have extended the reconstruction method of Chapter 4 to different types of noise as well as different signal models.

We have first considered the reconstruction of a signal which is sparse in a transformed domain and subject to Poisson-Gaussian noise. We have proposed a rational approximation to the log-likelihood as a fit function and have added a linear operator to the nonconvex approximation to the ℓ_0 regularizations. The final rational problem has then been solved globally using the framework of Chapter 3. The improvement obtained with our proposed approach is shown on two different applications.

We have also tackled the issue of robust estimation in the presence of outliers. Instead of the mean squares used in Chapter 4, we have proposed to use robust fit functions that may be nonconvex but are piecewise rational. Moreover, we have considered a signal model that is expressed as a union of subsets constraints. We have showed that the resulting inverse problem can be reformulated into a polynomial optimization problem using additional variables and thus globally solved using the methodology of Chapter 3. Finally our simulations have shown the good quality of the signal reconstructed in the presence of outliers.

- Chapter 6 -

Semi-definite programming problems

Part of the work in this chapter has been done in collaboration with Luis Briceño Arias from Universidad Técnica Federico Santa María during its visit in our lab. He suggests some of the formulations of the SDP problems ans some associated algorithms to solve them.

§ 6.1 Background

Semi-Definite Programming (SDP) aims at minimizing a linear objective function over the intersection of the cone of positive semi-definite matrices $\mathbb{S}^n_+$ with an affine space defined by m linear constraints. SDP problems have been studied in many works [dK02] (and references therein) for the last few decades as they often appear as relaxations of many more intricate problems. For instance, SDP problems appear in polynomial optimization as shown in Chapter 3, but also in graph theory [GW95], in the quadratic assignment and the traveling salesman problems [Sot11], in sparse principal component analysis [ZdG11], in floorplanning of large scale integration circuits [AL11], or in MIMO detection [LMS+10].

The current state-of-the-art SDP solvers are based on interior point methods [Ren01, dK02] which enjoy a polynomial-time complexity with respect to the dimension of the problem. Popular implementations of the latter methods include CSDP [BY07], DSDP [BYZ00], SDPA [FNYF07], SDPT3 [TTT99], and SeDuMi [Stu99]. Interior point methods are however limited in terms of scaling. For instance, if the sparsity of involved matrices is not considered, computation becomes too heavy when the dimension n of the semi-definite constraint is greater than 10^4 or when the number m of linear constraints is greater than 10^3. More specifically, iterations of interior point methods involve solving a linear system of size $m \times m$ which is highly demanding in terms of computation and memory when m is large. Many alternative methods have been proposed to solve such SDP problems: the bundle method [HR00], the mirror descend algorithm [LNM06], or augmented Lagrangian methods [MPRW09, ZST10, HM11, OCPB16]. Nevertheless, those methods are able to overcome interior point methods only on some specific SDP problems. A recent approach based on Burer-Monteiro factorisation [BM03, WW18] has shown some encouraging results. However, the latter factorisation leads to nonconvex problems and retrieving their global minimizers is possible only under some conditions [WW18, BVB19].

Our goal in this chapter is to explore proximal methods based on fixed point strategies [CP20] to solve SDP problems. We are especially interested in solving SDP problems arising from our relaxations (3.7) and (3.12) of rational problems, that are problems

where both the dimensions of the matrices n and number of linear constraints m are high. Previous works [MPRW09, NW12] have explored the benefit of proximal methods for SDP problems with a large m, which is a bottleneck for interior point methods. However, the dimension n has always been kept low, which does not correspond to our applications of the previous chapters as shown in Table 4.1.

The remaining of this section provides a short overview on SDP problems, their different forms, and the challenge they rise in comparison to LP problems. In Section 6.2, we reformulate both SDP primal and dual problems as unconstrained optimization problems. We then apply forthright several standard proximal methods using different splitting of the objective function. The latter methods are recalled in Appendix B. In Section 6.3, we modify the initial SDP problem with several regularizations in order to favour convergence of the proximal methods. We first apply the proximal algorithms on the augmented Lagrangian formulation of the dual SDP problem before using a quadratic regularization on the objective of the dual SDP problem in order to improve the decrease of the objective function. Similar approaches are tested on the primal problem as well. Finally, we substitute the semi-definite constraint with a barrier function in the spirit of interior point methods. All the developed algorithms are then compared on numerical simulations in Section 6.4.

Note that the SDP relaxations (3.8) of our rational optimization problems are in the dual form and thus we are particularly interested in the solution of this problem as they are the moments of the measure solution to Problem (3.11). Consequently, the dual solution is the focus of our algorithms.

6.1.1 Notation

The notation of this chapter is independent of the one used in other chapters. Furthermore, we introduce the following specific notation to this chapter: for a non empty closed convex set $\mathcal{X}$ in a Hilbert space $\mathcal{H}$, $\iota_{\mathcal{X}}$, $\Pi_{\mathcal{X}}$, and $\mathcal{N}_{\mathcal{X}}(x)$ denote respectively the indicator function, the projection, and the normal cone at x defined as

$$\iota_{\mathcal{X}}(x) = \begin{cases} 0, & \text{if } x \in \mathcal{X} \\ +\infty, & \text{otherwise} \end{cases}$$

$$\Pi_{\mathcal{X}}(x) = \operatorname*{arg\,min}_{y \in \mathcal{X}} \|y - x\|$$

$$\mathcal{N}_{\mathcal{X}}(x) = \begin{cases} \{u \in \mathcal{H} \mid (\forall y \in \mathcal{X}) \quad \langle u \mid y - x \rangle_{\mathcal{H}} \leq 0\}, & \text{if } x \in \mathcal{X} \\ \emptyset, & \text{otherwise}. \end{cases}$$

The barrier function on $\mathbb{R}^{n \times n}$ is denoted by ld and defined as

$$(\forall \mathbf{M} \in \mathbb{R}^{n \times n}) \quad \mathrm{ld}(\mathbf{M}) = -\log\det(\mathbf{M}).$$

We define the following inner products

$$(\forall (\mathbf{a}, \mathbf{b}) \in \mathbb{R}^n \times \mathbb{R}^n) \quad \langle \mathbf{a} \mid \mathbf{b} \rangle_{\mathbb{R}^n} = \sum_{i=1}^{n} a_i b_i$$

$$(\forall (\mathbf{A}, \mathbf{B}) \in \mathbb{S}^n \times \mathbb{S}^n) \quad \langle \mathbf{A} \mid \mathbf{B} \rangle_{\mathbb{S}^n} = \mathrm{Tr}(\mathbf{AB}).$$

Similarly to $\mathbb{S}^n_+$, we define $\mathbb{S}^n_-$ the cone of negative semi-definite $n \times n$ matrices. Some background on optimization can be found in Appendix B. The superscript * indicates either the convex conjugate of a function or the adjoint operator, ∂ denotes the subdifferential of a proper function, prox the proximal operator of a proper lower-semicontinuous convex function, and $J_{\mathcal{M}}$ the resolvent of a monotone operator $\mathcal{M}$. Definitions of the latter operators are also given in Appendix B.

6.1.2 SDP canonical formulations

The canonical primal and dual forms of an SDP problem are respectively defined as follows

$$\begin{cases} \underset{\mathbf{X}\in\mathbb{S}^n}{\text{minimize}} & \langle \mathbf{C} \mid \mathbf{X} \rangle_{\mathbb{S}^n} \\ \text{subject to} & A(\mathbf{X}) = \mathbf{b} \\ & \mathbf{X} \in \mathbb{S}^n_+ \end{cases} \quad (P) \qquad \begin{cases} \underset{\mathbf{y}\in\mathbb{R}^m}{\text{maximize}} & \langle \mathbf{b} \mid \mathbf{y} \rangle_{\mathbb{R}^m} \\ \text{subject to} & A^*(\mathbf{y}) + \mathbf{S} = \mathbf{C} \\ & \mathbf{S} \in \mathbb{S}^n_+ \end{cases} \quad (D)$$

where $\mathbf{C}$ is a symmetric matrix in $\mathbb{S}^n$, $\mathbf{b}$ is a vector in $\mathbb{R}^m$, A is a linear application from $\mathbb{S}^n$ to $\mathbb{R}^m$ and A^* is its adjoint operator. Note that, by the Riesz representation theorem, the linear application A can be represented by the set of matrices $(\mathbf{A}_i)_{i\in[\![1,m]\!]}$ of $\mathbb{S}^n$ such that

$$(\forall \mathbf{X} \in \mathbb{S}^n) \quad A(\mathbf{X}) = \left(\langle \mathbf{A}_i \mid \mathbf{X} \rangle_{\mathbb{S}^n}\right)_{i\in[\![1,m]\!]} .$$

Its adjoint operator is then

$$(\forall \mathbf{y} \in \mathbb{R}^m) \quad A^*(\mathbf{y}) = \sum_{i=1}^{m} y_i \mathbf{A}_i .$$

If both the primal and the dual feasible sets are non empty and if A is surjective, i.e. Robinson's constraint qualification condition is verified, the triplet $(\mathbf{X}, \mathbf{y}, \mathbf{S})$ is a primal-dual solution of (P) and (D) if and only if

$$\begin{cases} A^*(\mathbf{y}) + \mathbf{S} & = \mathbf{C} \ , \ \mathbf{S} \in \mathbb{S}^n_+ \\ A(\mathbf{X}) & = \mathbf{b} \ , \ \mathbf{X} \in \mathbb{S}^n_+ \\ \langle \mathbf{X} \mid \mathbf{S} \rangle_{\mathbb{S}^n_+} & = 0 \end{cases}$$

For SDP problems, Robinson's qualification condition is equivalent to Slater's condition, which is verified if there exists a strictly feasible point [GOL98]. The duality gap is then expressed by

$$\text{duality gap} = \langle \mathbf{C} \mid \mathbf{X} \rangle_{\mathbb{S}^n} - \langle \mathbf{b} \mid \mathbf{y} \rangle_{\mathbb{R}^m} = \langle \mathbf{S} \mid \mathbf{X} \rangle_{\mathbb{S}^n} .$$

The complementary slackness condition hence gives information on the duality gap: the constraint qualification condition implies that strong duality holds.

Note that Problem (3.8) does not take exactly the form of (D) but Section 6.1.4 provides reformulations of (3.8) into the standard dual form.

6.1.3 Link between primal and dual solutions

As stated above, we look for the dual solution of SDP problems in order to solve rational optimization problems of Chapter 3. In this section, we show that the dual solution can actually be extracted from the primal solution using the Lagrangian duality.

Let us write the Lagrangian $\mathcal{L}$ of the SDP primal problem (P)

$$(\forall \mathbf{X} \in \mathbb{S}^n)(\forall \mathbf{y} \in \mathbb{R}^m) \quad \mathcal{L}(\mathbf{X}, \mathbf{y}) = \langle \mathbf{C} \mid \mathbf{X} \rangle_{\mathbb{S}^n} + \langle \mathbf{y} \mid \mathbf{b} - A(\mathbf{X}) \rangle_{\mathbb{R}^m} + \iota_{\mathbb{S}^n_+}(\mathbf{X}) .$$

At optimality, the Karush-Kuhn-Tucker conditions (KKT conditions) state that

$$0 \in \partial_{\mathbf{X}} \mathcal{L}(\mathbf{X}, \mathbf{y}) = \mathbf{C} - A^*(\mathbf{y}) + \mathcal{N}_{\mathbb{S}^n_+}(\mathbf{X}) .$$

Setting $\mathbf{Z} = A^*(\mathbf{y}) - \mathbf{C}$, we obtain

$$\begin{aligned} 0 \in \mathbf{C} - A^*(\mathbf{y}) + \mathcal{N}_{\mathbb{S}^n_+}(\mathbf{X}) &\iff 0 \in \mathcal{N}_{\mathbb{S}^n_+}(\mathbf{X}) - \mathbf{Z} \\ &\iff \mathbf{X} \in \mathcal{N}_{\mathbb{S}^n_+}^{-1}(\mathbf{Z}) = \mathcal{N}_{\mathbb{S}^n_-}(\mathbf{Z}) \\ &\iff \mathbf{X} + \mathbf{Z} \in (\mathcal{N}_{\mathbb{S}^n_-} + \mathrm{Id})(\mathbf{Z}) \\ &\iff \mathbf{Z} \in J_{\mathcal{N}_{\mathbb{S}^n_-}}(\mathbf{X} + \mathbf{Z}) \\ &\iff \mathbf{Z} = \Pi_{\mathbb{S}^n_-}(\mathbf{X} + \mathbf{Z})\,. \end{aligned}$$

The variable $\mathbf{Z}$ is hence a fixed point of the operator $\Pi_{\mathbb{S}^n_-}(\mathbf{X} + .)$. Moreover, if AA^* is invertible, we have

$$\begin{aligned} \mathbf{Z} = A^*(\mathbf{y}) - \mathbf{C} &\implies A(\mathbf{Z} + \mathbf{C}) = AA^*(\mathbf{y}) \\ &\implies \mathbf{y} = (AA^*)^{-1}(A(\mathbf{Z} + \mathbf{C}))\,. \end{aligned}$$

Therefore, an optimal solution $\mathbf{y}_*$ of the dual problem (D) can be retrieved from an optimal solution $\mathbf{X}_*$ of the corresponding primal problem (P) by using Algorithm 2. The convergence of the latter algorithm results from the firm non-expansiveness of the operator $\Pi_{\mathbb{S}^n_-}(\mathbf{X} + .)$ [BC11]. However, in practice, Algorithm 2 converges slowly and we will favour methods that return directly $\mathbf{y}_*$.

Algorithm 2: Retrieve SDP dual solution from SDP primal solution

Input: $\mathbf{X}_*$, optimal solution of problem (P)
Input: Set $\mathbf{Z} \in \mathbb{S}^n$
Output: $\mathbf{y}_*$, solution of (D)

```
for k = 0,1,... do
    Z^(k+1) = Π_{S^n_-}(X_* + Z^(k)) ;
y_* = (AA*)^{-1}(A(Z^(k) + C)) ;
```

6.1.4 Free variables

Some SDP problems may include unrestricted variables, or equivalently linear constraints on the dual problem. Those variables are referred as free variables and appear naturally in many problems. For instance, the variable γ in (2.3) is a free variable and the linear constraints on the moments in Problems (3.6) and (3.12) generate free variables.

Assuming p is an integer smaller than m, the primal and dual SDP problems hence take the following more general forms

$$\begin{cases} \underset{(\mathbf{X},\mathbf{z})\in\mathbb{S}^n\times\mathbb{R}^p}{\text{minimize}} & \langle \mathbf{C} \mid \mathbf{X}\rangle_{\mathbb{S}^n} + \langle \mathbf{f} \mid \mathbf{z}\rangle_{\mathbb{R}^p} \\ \text{subject to} & A(\mathbf{X}) + D(\mathbf{z}) = \mathbf{b} \\ & \mathbf{X} \in \mathbb{S}^n_+ \end{cases} \quad (P_{\text{fr}}) \qquad \begin{cases} \underset{\mathbf{y}\in\mathbb{R}^m}{\text{maximize}} & \langle \mathbf{b} \mid \mathbf{y}\rangle_{\mathbb{R}^m} \\ \text{subject to} & \mathbf{C} - A^*(\mathbf{y}) \in \mathbb{S}^n_+ \\ & D^*(\mathbf{y}) - \mathbf{f} = 0 \end{cases} \quad (D_{\text{fr}})$$

where $\mathbf{z}$ is the vector of free variables in $\mathbb{R}^p$, $\mathbf{f}$ is a vector of $\mathbb{R}^p$, and D is a linear application from $\mathbb{R}^p$ to $\mathbb{R}^m$.

Free variables are not a theoretical issue but a rather important computational one. Indeed, interior point methods solve a positive definite linear system at each iteration using Cholesky factorisation customised to benefit from sparsity. However, free variables make the system indefinite and consequently harder to solve.

We present below four common methods to deal with free variables, each one aiming at allowing the use of Cholesky factorisation. They all reduce Problem (P_{fr}) and (D_{fr}) to their standard form in order to use the same algorithms to solve it.

- **Splitting Method:** The free variable $\mathbf{z}$ is split into two positive variables $\mathbf{z}_+$ and $\mathbf{z}_-$ such that $\mathbf{z} = \mathbf{z}_+ - \mathbf{z}_-$. Problems ($P_{\text{fr}}$) and ($D_{\text{fr}}$) can then be written in the canonical forms of (P) and (D) with matrices in $\mathbb{S}_+^{n+2p}$. Albeit simple, this method is numerically unstable. Indeed, the converted SDP problem has a continuum of optimal solutions and its dual has no interior feasible point. Moreover, $\mathbf{z}_+$ and $\mathbf{z}_-$ may be unbounded individually whereas their difference $\mathbf{z}$ is bounded. These are serious issues in interior point methods. Splitting is the default method used in SeDuMi because of its cheap cost. However, when numerical instability appears, Andersen method can be used.

- **Kobayashi Method:** This method [KNK07] eliminates $\mathbf{z}$ using a well-chosen basis that preserves sparsity of A. Compared to the splitting method, the dimension of the matrix space does not change and this method prevents numerical difficulties. Nonetheless, the choice of the basis is highly important but can be difficult in practice.

- **Andersen Method:** [AB08] The free variables are arranged inside a Lorentz cone $\mathcal{K}_{\text{Lorentz}}$ together with an extra unconstrained variable z_0

$$\mathcal{K}_{\text{Lorentz}} = \left\{ (z_0, \mathbf{z}) \in \mathbb{R}^{p+1} \middle| z_0 \geq \|\mathbf{z}\|_2 \right\} .$$

  Problems (P_{fr}) and (D_{fr}) can thus be written in the canonical forms by using the additional linear matrix inequality

$$\begin{bmatrix} z_0 & \mathbf{z}^\top \\ \mathbf{z} & z_0 \operatorname{Id}_p \end{bmatrix} \in \mathbb{S}_+^{p+1} .$$

  This method is numerically more stable. However, the dimension of the matrix space increased since instead of $\mathbb{S}^n$, we now work in $\mathbb{S}^n \times \mathbb{R}^p$ and consequently the computational time to solve the problem also increases.

- **Mészáros Method:** This method [Més98] is mainly used in interior point algorithms to handle free variables directly by perturbing the indefinite linear system to be solved at each iteration in order to make it quasi-definite. The method also preserves the sparsity of the operator A and D, which is of special interest for large scale problems. The main downside is that, since the solutions of the linear system are perturbed, they need to be handled carefully in the following steps to ensure the convergence of the interior point method. A variant of this method is used in the solver SDPT3.

Note that those methods to deal with free variables in SDP are highly inspired from the ones used in solving LP problems.

In the following, we suppose that the SDP relaxation (3.8) has been preprocessed by one of the above method and thus focus on solving the standard forms (P) and (D).

6.1.5 Comparison with LP

Although LP and SDP problems share a common form, they have significant differences. The feasible set of a LP problem is a polyhedron which has nice properties. For instance, its projection is also a polyhedron and its image by a linear application is always closed. Hence, LP problems are either infeasible or solvable, that is we can find a primal-dual solution. Moreover strong duality always holds in LP: the primal optimal value and the dual optimal value are identical. In other words, the duality gap is identical to zero at optimality. Thereby, we can apply the separating hyperplane theorem on the compact

set $\{\mathbf{b}\}$ and on the closed set $A(\mathbb{R}_+)$. If we can find a separating hyperplane, then the problem is infeasible, otherwise it is feasible and strong duality holds.

On the other hand, the feasible set of a SDP is called a spectrahedron which is also a closed convex set. However, its projection, albeit convex, is not closed in general. In the same way, its image by a linear application is not necessarily closed. As a consequence, Slater's condition may not hold. This is the root of some strange cases that do not happen in LP.

Let us take the following examples [Tod01] (detailed computations can be found in Section C.1):

- Problem (P) is feasible and bounded but does not have a solution while Problem (D) has a solution:

$$\mathbf{A}_1 = \begin{bmatrix} 0 & -1 \\ -1 & 2 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 2 & -1 \\ -1 & 0 \end{bmatrix}, \mathbf{b} = 2$$

  The optimal value of Problem (P) is -2 but it is never reached.

- Problem (P) has a solution, Problem (D) is feasible but does not have solution:

$$\mathbf{A}_1 = \begin{bmatrix} -1 & 0 \\ 0 & 0 \end{bmatrix}, \mathbf{A}_2 = \begin{bmatrix} 0 & 0 \\ 0 & -1 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$$

  Note that there is no duality gap since both primal and dual optimal values are zero.

- Problem (D) has a solution whereas Problem (P) is infeasible:

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \mathbf{A}_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$$

- Problem (P) and Problem (D) have a solution but there is a duality gap:

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{A}_2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 2 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$$

  The optimal value of the primal is 1 while the optimal value of the dual is 0.

Therefore, solving a general SDP may be difficult. Fortunately, for SDP problems obtained from relaxations of polynomial optimization problem, it has been proved [JH15] that strong duality always holds and as a consequence, those weird cases do not arise. Moreover, we assume that the considered SDP problems are feasible. Indeed, we search for faster algorithms to process high dimensional problems. Methods to detect infeasibility such as the homogeneous self-dual embedding [dKRT97, LSZ00] or the big M approach [MA89] are available in the literature and can be used to package the developed algorithms.

Therefore, in the following, we only focus on solving feasible SDP problems in their canonical forms where strong duality holds.

§ 6.2 Proximal methods on standard forms

6.2.1 Primal problem

Our goal in this section is to apply standard proximal methods, such as Chambolle-Pock or Douglas-Rachford algorithms (cf. Appendix B.4) as well as some enhanced versions

of those methods, on different formulations of the standard SDP problem. Hence, we reformulate both SDP primal and dual problems as unconstrained optimization problems and apply proximal methods forthright using different splitting of the objective function.

6.2.1.1 Chambolle-Pock algorithm

We write the primal SDP problem (P) as an unconstrained problem

$$(P) \quad \Longleftrightarrow \quad \min_{\mathbf{X}\in\mathbb{S}^n} \underbrace{\langle \mathbf{C} \mid \mathbf{X}\rangle_{\mathbb{S}^n} + \iota_{\mathbb{S}^n_+}(\mathbf{X})}_{=f(\mathbf{X})} + \underbrace{\iota_{\{\mathbf{b}\}}(A(\mathbf{X}))}_{=g\circ A(\mathbf{X})} . \tag{6.1}$$

Applying Fermat's rule, we write (6.1) as a monotone inclusion problem:

Find $\mathbf{X}$ in $\mathbb{S}^n$ such that

$$0 \in \partial f(\mathbf{X}) + \partial(g \circ A)(\mathbf{X}) . \tag{6.2}$$

As the subdifferential is a monotone operator (see Property 1 in Appendix B) and A is a linear operator, Problem (6.2) can be solved with Chambolle-Pock algorithm (see Appendix B.4). The algorithm is shown in Algorithm 3 where the blue expressions are the results of the computations of the resolvents.

Algorithm 3: ChamPockP, Chambolle-Pock algorithm applied on (6.1)

Input: Set $(\mathbf{X}^{(0)}, \mathbf{y}^{(0)}) \in \mathbb{S}^n \times \mathbb{R}^m$
Input: Set $\theta \in [0,1]$ and $(\tau, \sigma) \in \mathbb{R}_+ \times \mathbb{R}_+$
Output: $\mathbf{X}$, solution of (P) and $\mathbf{y}$, solution of (D)

Set $\overline{\mathbf{X}}^{(0)} = \mathbf{X}^{(0)}$;
for $k = 0,1,\ldots$ **do**
  $\mathbf{y}^{(k)} \leftarrow \text{prox}_{\sigma g^*}\left(\mathbf{y}^{(k)} + \sigma A\overline{\mathbf{X}}^{(k)}\right) = \mathbf{y}^{(k)} + \sigma\left(A\overline{\mathbf{X}}^{(k)} - \mathbf{b}\right)$;
  $\mathbf{X}^{(k+1)} \leftarrow \text{prox}_{\tau f}\left(\mathbf{X}^{(k)} - \tau A^*\mathbf{y}^{(k+1)}\right) = \Pi_{\mathbb{S}^n_+}\left(\mathbf{X}^{(k)} - \tau\left(A^*\mathbf{y}^{(k+1)} - \mathbf{C}\right)\right)$;
  $\overline{\mathbf{X}}^{(k+1)} \leftarrow \mathbf{X}^{(k+1)} + \theta\left(\mathbf{X}^{(k+1)} - \mathbf{X}^{(k)}\right)$;

In order to obtain feasible iterates and based on [BAR18], we propose to add an extra projection on the hyperplane of the affine constraints at the end of each iteration in the algorithm. It may also help the convergence to occur with a lower number of iterations. The projection is shown in red in Algorithm 4.

Algorithm 4: ChamPock+$\mathbf{\Pi}$, Chambolle-Pock algorithm applied on (6.1) with projection

Input: Set $(\mathbf{X}^{(0)}, \mathbf{y}^{(0)}) \in \mathbb{S}^n \times \mathbb{R}^m$
Input: Set $\theta \in [0,1]$ and $(\tau, \sigma) \in \mathbb{R}_+ \times \mathbb{R}_+$
Output: $\mathbf{X}$, solution of (P) and $\mathbf{y}$, solution of (D)

Set $\overline{\mathbf{X}}^{(0)} = \mathbf{X}^{(0)}$;
for $k = 0,1,\ldots$ **do**
  $\mathbf{y}^{(k)} \leftarrow \text{prox}_{\sigma g^*}\left(\mathbf{y}^{(k)} + \sigma A\overline{\mathbf{X}}^{(k)}\right)$;
  $\mathbf{P}^{(k+1)} \leftarrow \text{prox}_{\tau f}\left(\mathbf{X}^{(k)} - \tau A^*\mathbf{y}^{(k+1)}\right)$;
  $\mathbf{X}^{(k+1)} \leftarrow \mathbf{P}^{(k+1)} - A^*(AA^*)^{-1}\left(A\mathbf{P}^{(k+1)} - \mathbf{b}\right)$;
  $\overline{\mathbf{X}}^{(k+1)} \leftarrow \mathbf{X}^{(k+1)} + \theta\left(\mathbf{P}^{(k+1)} - \mathbf{X}^{(k)}\right)$;

6.2.1.2 Douglas-Rachford algorithm

The primal SDP problem (P) can also be reformulated as a problem with only linear constraints

$$(P) \quad \Longleftrightarrow \quad \begin{cases} \min\limits_{\mathbf{X}\in\mathbb{S}^n} & \langle \mathbf{C} \mid \mathbf{X}\rangle_{\mathbb{S}^n} + \iota_{\mathbb{S}^n_+}(\mathbf{X}) \\ \text{s.t.} & A(\mathbf{X}) = \mathbf{b} \end{cases} \tag{6.3}$$

We write the Lagrangian $\mathcal{L}$ of Problem (6.3)

$$(\forall (\mathbf{X},\mathbf{y}) \in \mathbb{S}^n \times \mathbb{R}^m) \quad \mathcal{L}(\mathbf{X},\mathbf{y}) = \langle \mathbf{C} \mid \mathbf{X}\rangle_{\mathbb{S}^n} + \iota_{\mathbb{S}^n_+}(\mathbf{X}) + \langle \mathbf{y} \mid \mathbf{b} - A(\mathbf{X})\rangle_{\mathbb{R}^m}$$

By writing the KKT conditions, we search for a point $(\mathbf{X},\mathbf{y})$ of $\mathbb{S}^n \times \mathbb{R}^m$ at optimality such that

$$\begin{cases} \mathbf{0} \in & \mathbf{C} + \mathcal{N}_{\mathbb{S}^n_+}(\mathbf{X}) + A^*(\mathbf{y}) \\ \mathbf{0} = & \mathbf{b} - A(\mathbf{X})\,. \end{cases}$$

By defining the constants $\mathbf{C}$ and $\mathbf{b}$ as translations, the above conditions can be written as a monotone inclusion problem:

$$\begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix} \in \underbrace{\begin{bmatrix} \mathcal{N}_{\mathbb{S}^n_+} + \mathbf{C} & \mathbf{0} \\ \mathbf{0} & \mathbf{b} \end{bmatrix}}_{=\mathcal{A}} \begin{pmatrix} \mathbf{X} \\ \mathbf{y} \end{pmatrix} + \underbrace{\begin{bmatrix} \mathbf{0} & A^* \\ -A & \mathbf{0} \end{bmatrix}}_{=\mathcal{B}} \begin{pmatrix} \mathbf{X} \\ \mathbf{y} \end{pmatrix} \tag{6.4}$$

In Algorithm 5, we apply Douglas-Rachford algorithm on the monotone inclusion problem (6.4). Note that the main variables in the algorithms are $(\mathbf{Z},\mathbf{w})$ while we are interested only in the intermediate variables $(\mathbf{X},\mathbf{y})$.

Algorithm 5: DougRachP, Douglas-Rachford algorithm applied on (6.4)

Input: Set $(\mathbf{Z}^{(0)},\mathbf{w}^{(0)}) \in \mathbb{S}^n \times \mathbb{R}^m$
Input: Set $\gamma \in \mathbb{R}_+$
Output: $\mathbf{X}$, solution of (P) and $\mathbf{y}$, solution of (D)
for *k = 0,1,...* **do**
  $\begin{pmatrix} \mathbf{X}^{(k)} \\ \mathbf{y}^{(k)} \end{pmatrix} \leftarrow J_{\gamma\mathcal{B}} \begin{pmatrix} \mathbf{Z}^{(k)} \\ \mathbf{w}^{(k)} \end{pmatrix}$;
  $\begin{pmatrix} \mathbf{Z}^{(k+1)} \\ \mathbf{w}^{(k+1)} \end{pmatrix} \leftarrow J_{\gamma\mathcal{A}} \left(2 \begin{pmatrix} \mathbf{X}^{(k)} \\ \mathbf{y}^{(k)} \end{pmatrix} - \begin{pmatrix} \mathbf{Z}^{(k)} \\ \mathbf{w}^{(k)} \end{pmatrix} \right) + \begin{pmatrix} \mathbf{Z}^{(k)} \\ \mathbf{w}^{(k)} \end{pmatrix} - \begin{pmatrix} \mathbf{X}^{(k)} \\ \mathbf{y}^{(k)} \end{pmatrix}$;

The two resolvents $J_{\gamma\mathcal{B}}$ and $J_{\gamma\mathcal{A}}$ are given by

$$\begin{pmatrix} \mathbf{X} \\ \mathbf{y} \end{pmatrix} = J_{\gamma\mathcal{B}} \begin{pmatrix} \mathbf{Z} \\ \mathbf{w} \end{pmatrix} \Longleftrightarrow \gamma^{-1} \begin{pmatrix} \mathbf{Z}-\mathbf{X} \\ \mathbf{w}-\mathbf{y} \end{pmatrix} = \begin{pmatrix} A^*\mathbf{y} \\ -A\mathbf{X} \end{pmatrix} \Longleftrightarrow \begin{cases} \mathbf{y} = (\mathrm{Id} + \gamma^2 AA^*)^{-1}(\mathbf{w} + \gamma A\mathbf{Z}) \\ \mathbf{X} = \mathbf{Z} - \gamma A^*\mathbf{y} \end{cases}$$

$$\begin{pmatrix} \mathbf{Z} \\ \mathbf{w} \end{pmatrix} = J_{\gamma\mathcal{A}} \begin{pmatrix} \mathbf{X} \\ \mathbf{y} \end{pmatrix} \Longleftrightarrow \gamma^{-1} \begin{pmatrix} \mathbf{X}-\mathbf{Z} \\ \mathbf{y}-\mathbf{w} \end{pmatrix} \in \begin{pmatrix} \mathcal{N}_{\mathbb{S}^n_+}(\mathbf{Z}) + \mathbf{C} \\ \mathbf{b} \end{pmatrix} \Longleftrightarrow \begin{cases} \mathbf{Z} = \Pi_{\mathbb{S}^n_+}(\mathbf{X} - \gamma\mathbf{C}) \\ \mathbf{w} = \mathbf{y} - \gamma\mathbf{b} \end{cases}.$$

Note that we can interchange the two resolvents, i.e. applying first $J_{\gamma\mathcal{A}}$ and then $J_{\gamma\mathcal{B}}$ in Algorithm 5.

6.2.2 Dual problem

6.2.2.1 Chambolle-Pock algorithm

In a similar manner to the primal standard form, we reformulate the dual SDP problem (D) first as an unconstrained minimization problem

$$(D) \quad \Longleftrightarrow \quad \min_{(\mathbf{y},\mathbf{S})\in\mathbb{R}^m\times\mathbb{S}^n} \underbrace{\langle -\mathbf{b} \mid \mathbf{y}\rangle_{\mathbb{R}^m} + \iota_{\mathbb{S}^n_+}(\mathbf{S})}_{=f(\mathbf{y},\mathbf{S})} + \underbrace{\iota_{\{\mathbf{C}\}}(A^*\mathbf{y} + \mathbf{S})}_{=g\circ L^*(\mathbf{y},\mathbf{S})}, \tag{6.5}$$

where $L^* = \begin{bmatrix} A^* & \mathrm{Id} \end{bmatrix}$ and hence $L = \begin{bmatrix} A \\ \mathrm{Id} \end{bmatrix}$. We notice the close similarities with the unconstrained formulation written for the primal problem (P) in (6.1). Applying Fermat's rule, we write (6.5) as a monotone inclusion problem:

Find $(\mathbf{y}, \mathbf{S})$ in $\mathbb{R}^m \times \mathbb{S}^n$ such that

$$\mathbf{0} \in \partial f(\mathbf{y}, \mathbf{S}) + \partial (g \circ L^*)(\mathbf{y}, \mathbf{S})\,. \tag{6.6}$$

Problem (6.6) is a monotone inclusion problem that can be solved using Chambolle-Pock algorithm as shown in Algorithm 6.

Algorithm 6: ChamPockD, Chambolle-Pock algorithm applied on (6.5)

Input: Set $(\mathbf{y}^{(0)}, \mathbf{S}^{(0)}, \mathbf{X}^{(0)}) \in \mathbb{R}^m \times \mathbb{S}^n \times \mathbb{S}^n$
Input: Set $\theta \in [0;1]$ and $(\tau, \sigma) \in \mathbb{R}_+ \times \mathbb{R}_+$
Output: $(\mathbf{y}, \mathbf{S})$, solution of (D) and $\mathbf{X}$, solution of (P)

Set $(\overline{\mathbf{y}}^{(0)}, \overline{\mathbf{S}}^{(0)}) = (\mathbf{y}^{(0)}, \mathbf{S}^{(0)})$;
for $k = 0,1,\ldots$ **do**
  $\mathbf{X}^{(k+1)} \leftarrow \mathrm{prox}_{\sigma g^*}\left(\mathbf{X}^{(k)} + \sigma(A^*\overline{\mathbf{y}}^{(k)} + \overline{\mathbf{S}}^{(k)})\right) = \mathbf{X}^{(k)} + \sigma\left(A^*\overline{\mathbf{y}}^{(k)} + \overline{\mathbf{S}}^{(k)} - \mathbf{C}\right)$;
  $\begin{pmatrix} \mathbf{y}^{(k+1)} \\ \mathbf{S}^{(k+1)} \end{pmatrix} \leftarrow \mathrm{prox}_{\tau f}\left(\begin{pmatrix} \mathbf{y}^{(k)} \\ \mathbf{S}^{(k)} \end{pmatrix} - \tau L \mathbf{X}^{(k+1)}\right) = \begin{pmatrix} \mathbf{y}^{(k)} - \tau(A\mathbf{X}^{(k+1)} + \mathbf{b}) \\ \Pi_{\mathbb{S}^n_+}\left(\mathbf{S}^{(k)} - \tau \mathbf{X}^{(k+1)}\right) \end{pmatrix}$;
  $\begin{pmatrix} \overline{\mathbf{y}}^{(k+1)} \\ \overline{\mathbf{S}}^{(k+1)} \end{pmatrix} \leftarrow \begin{pmatrix} \mathbf{y}^{(k+1)} \\ \mathbf{S}^{(k+1)} \end{pmatrix} + \theta\left(\begin{pmatrix} \mathbf{y}^{(k+1)} \\ \mathbf{S}^{(k+1)} \end{pmatrix} - \begin{pmatrix} \mathbf{y}^{(k)} \\ \mathbf{S}^{(k)} \end{pmatrix}\right)$;

6.2.2.2 Douglas-Rachford algorithm

We can reformulate the function $g \circ L^*$ in problem (6.5) as an indicator function

$$g \circ L^*(\mathbf{y}, \mathbf{S}) = \iota_{\{\mathbf{C}\}}(A^*\mathbf{y} + \mathbf{S}) = \iota_{\mathcal{D}}(\mathbf{y}, \mathbf{S})\,,$$

where $\mathcal{D} = \left\{(\mathbf{y}, \mathbf{S}) \in \mathbb{R}^m \times \mathbb{S}^n \mid L^* \begin{pmatrix} \mathbf{y} \\ \mathbf{S} \end{pmatrix} = \mathbf{C}\right\}$. Problem (6.5) hence reads

$$\mathbf{0} \in \partial f(\mathbf{y}, \mathbf{S}) + \partial \iota_{\mathcal{D}}(\mathbf{y}, \mathbf{S})\,.$$

As ∂f and $\partial \iota_{\mathcal{D}}$ are monotone operators, we can apply Douglas-Rachford algorithm to the latter problem and we obtain Algorithm 7.

Algorithm 7: DougRachD, Douglas-Rachford algorithm applied on (6.5)

Input: Set $(\mathbf{w}^{(0)}, \mathbf{Z}^{(0)}) \in \mathbb{R}^m \times \mathbb{S}^n$
Input: Set $\gamma \in \mathbb{R}_+$
Output: $(\mathbf{y}, \mathbf{S})$, solution of (D)

for $k = 0,1,\ldots$ **do**
  $\begin{pmatrix} \mathbf{y}^{(k)} \\ \mathbf{S}^{(k)} \end{pmatrix} \leftarrow \Pi_{\mathcal{D}} \begin{pmatrix} w^{(k)} \\ Z^{(k)} \end{pmatrix}$;
  $\begin{pmatrix} \mathbf{w}^{(k+1)} \\ \mathbf{Z}^{(k+1)} \end{pmatrix} \leftarrow \mathrm{prox}_{\gamma f}\left(2\begin{pmatrix} \mathbf{y}^{(k)} \\ \mathbf{S}^{(k)} \end{pmatrix} - \begin{pmatrix} \mathbf{w}^{(k)} \\ \mathbf{Z}^{(k)} \end{pmatrix}\right) + \begin{pmatrix} \mathbf{w}^{(k)} \\ \mathbf{Z}^{(k)} \end{pmatrix} - \begin{pmatrix} \mathbf{y}^{(k)} \\ \mathbf{S}^{(k)} \end{pmatrix}$;

We compute the projection $\Pi_{\mathcal{D}}$ and the proximal operator $\mathrm{prox}_{\gamma f}$:

$$\begin{pmatrix}\mathbf{y}\\ \mathbf{S}\end{pmatrix} = \Pi_{\mathcal{D}}\begin{pmatrix}\mathbf{w}\\ \mathbf{Z}\end{pmatrix} = \begin{pmatrix}\mathbf{w}\\ \mathbf{Z}\end{pmatrix} - L(A^*A + \mathrm{Id})^{-1}(A^*\mathbf{w} + \mathbf{Z} - \mathbf{C})$$

$$\begin{pmatrix}\mathbf{w}\\ \mathbf{Z}\end{pmatrix} = \mathrm{prox}_{\gamma f}\begin{pmatrix}\mathbf{y}\\ \mathbf{S}\end{pmatrix} \iff \gamma^{-1}\begin{pmatrix}\mathbf{y}-\mathbf{w}\\ \mathbf{S}-\mathbf{Z}\end{pmatrix} \in \begin{pmatrix}-\mathbf{b}\\ \mathcal{N}_{\mathbb{S}^n_+}(\mathbf{Z})\end{pmatrix} \iff \begin{cases}\mathbf{w} = \mathbf{y} + \gamma\mathbf{b}\\ \mathbf{Z} = \Pi_{\mathbb{S}^n_+}(\mathbf{S})\end{cases}$$

Notice that the computation of $\Pi_{\mathcal{D}}$ requires to inverse a $n^2 \times n^2$ matrix while in Algorithm 5, the computation of the first resolvent requires to inverse a $m \times m$ matrix.

6.2.2.3 FISTA

We can eliminate the dual variable $\mathbf{S}$ and keep only the dual variable $\mathbf{y}$ in (D), and we obtain

$$\begin{cases}\underset{\mathbf{y}\in\mathbb{R}^m}{\text{maximize}} & \langle \mathbf{b} \mid \mathbf{y}\rangle_{\mathbb{R}^m}\\ \text{subject to} & A^*(\mathbf{y}) + \mathbf{S} = \mathbf{C} - A^*(\mathbf{y}) \in \mathbb{S}^n_+\end{cases}$$

We then express it as an unconstrained problem using an indicator function:

$$\underset{\mathbf{y}\in\mathbb{R}^m}{\text{minimize}} \underbrace{\langle -\mathbf{b} \mid \mathbf{y}\rangle_{\mathbb{R}^m}}_{=f(\mathbf{y})} + \underbrace{\iota_{\mathbb{S}^n_+}(\mathbf{C} - A^*(\mathbf{y}))}_{=g\circ A^*(\mathbf{y})}. \tag{6.7}$$

We apply Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) [BT09], an over-relaxed version of FB which is given in Appendix B, to solve Problem (6.7) as shown in Algorithm 8. Notice that the work of Chambolle and Dossal [CD15] provides a sufficient condition on the sequence $(\theta_k)_{k\in\mathbb{N}}$ to ensure the convergence of the iterates of FISTA to a minimizer.

Algorithm 8: FISTA, Fast Iterative Shrinkage-Thresholding Algorithm to solve (D)

Input: Set $\mathbf{y}^{(0)} \in \mathbb{R}^m$ and define $\mathbf{v}^{(0)} = \mathbf{y}^{(0)}$
Output: $\mathbf{y}$, solution of (D)

for $k = 1,2,\ldots$ **do**
- $\theta_k \leftarrow \frac{2}{k+1}$;
- $\mathbf{z} \leftarrow (1-\theta_k)\mathbf{y}^{(k-1)} + \theta_k\mathbf{v}^{(k-1)}$;
- $\mathbf{y}^{(k)} \leftarrow \mathrm{prox}_{\alpha_k g\circ A^*}(\mathbf{z} - \alpha_k\nabla f(\mathbf{z}))$;
- $\mathbf{v}^{(k)} \leftarrow \mathbf{y}^{(k-1)} + \frac{1}{\theta_k}(\mathbf{y}^{(k)} - \mathbf{y}^{(k-1)})$;

To compute $\mathrm{prox}_{\alpha_k g\circ A^*}$, we need to solve the following optimization problem:

Find $\hat{\mathbf{y}}$ in $\mathbb{R}^m$ such that

$$\hat{\mathbf{y}} = \mathrm{prox}_{\alpha_k g\circ A^*}(\tilde{\mathbf{y}}) = \underset{\mathbf{y}\in\mathbb{R}^m}{\arg\min}\, g(A^*(\mathbf{y})) + \phi(\mathbf{y}),$$

where $\phi(\mathbf{y}) = \frac{1}{2}\|\mathbf{y} - \tilde{\mathbf{y}}\|^2$ for $\mathbf{y}$ in $\mathbb{R}^m$.

We actually solve its dual problem

$$\hat{\mathbf{X}} = \underset{\mathbf{X}\in\mathbb{S}^n}{\arg\max} -g^*(\mathbf{X}) - \phi^*(-A(\mathbf{X})) = \underset{\mathbf{X}\in\mathbb{S}^n}{\arg\min}\, g^*(\mathbf{X}) + \phi^*(-A(\mathbf{X})),$$

where $\phi^*(\mathbf{y}) = \mathbf{y}^\top\tilde{\mathbf{y}} + \frac{1}{2}\|\mathbf{y}\|^2$ for all $\mathbf{y}$ in $\mathbb{R}^m$. The dual problem is also solved using FISTA as shown in Algorithm 9. We set the step-size γ_k to $\|AA^*\|^{-1}$ since the gradient of the differentiable function $(\phi^* \circ (-A))$ is $\|AA^*\|$-Lipschitz

$$(\forall \mathbf{X} \in \mathbb{S}^n) \quad \nabla(\phi^* \circ (-A))(\mathbf{X}) = -A^*(\tilde{\mathbf{y}} - A\mathbf{X}).$$

Algorithm 9: Inner loop to compute $\text{prox}_{g\circ A^*}$ in Algorithm 8

Input: A point $\tilde{\mathbf{y}}$ where to compute $\text{prox}_{g\circ A^*}$
Input: Set $\mathbf{X}^{(0)} \in \mathbb{S}^n$ and define $\mathbf{V}^{(0)} = \mathbf{X}^{(0)}$
Output: $\hat{\mathbf{y}}$, value of $\text{prox}_{\alpha_k g\circ A^*}$ at point $\tilde{\mathbf{y}}$

for $k = 0,1,\ldots$ **do**
- $\theta_k \leftarrow \frac{2}{k+1}$;
- $\mathbf{Z} \leftarrow (1-\theta_k)\mathbf{X}^{(k-1)} + \theta_k \mathbf{V}^{(k-1)}$;
- $\mathbf{y}^{(k)} \leftarrow \tilde{\mathbf{y}} - A(\mathbf{Z})$;
- $\tilde{\mathbf{X}}^{(k)} \leftarrow \mathbf{Z} + \gamma_k B^{-1} A^* \mathbf{y}^{(k)}$;
- $\mathbf{X}^{(k)} \leftarrow \tilde{\mathbf{X}}^{(k)} - \gamma_k B^{-1}(\mathbf{C} - \Pi_{\mathbb{S}^n_+}(\mathbf{C} - \gamma_k^{-1} B\tilde{\mathbf{X}}^{(k)}))$;
- $\mathbf{V}^{(k)} \leftarrow \mathbf{X}^{(k-1)} + \frac{1}{\theta_k}(\mathbf{X}^{(k)} - \mathbf{X}^{(k-1)})$;

6.2.2.4 Rescaled forward-backward primal-dual algorithm

Rescaled Forward-Backward Primal-Dual algorithm (RFBPD) [KP15] is a variant of Chambolle-Pock algorithm that adds a scaling as well as an extra relaxation step on the dual variable, and applies Moreau's formula to compute the resolvent of the operator affected by the convex conjugate. We apply it on (6.7) as shown in Algorithm 10.

Algorithm 10: RFBPD, Rescaled forward-backward primal-dual algorithm to solve (D)

Input: Set $\mathbf{y}^{(0)} \in \mathbb{R}^m$ and $\mathbf{X}^{(0)} \in \mathbb{S}^n_+$
Input: Set $(\tau, \sigma) \in \mathbb{R}^*_+ \times \mathbb{R}^*_+$
Output: $\mathbf{y}$: solution of the problem (D)

for $k = 0,1,\ldots$ **do**
- $\mathbf{p}^{(k)} \leftarrow \text{prox}_{\tau f}(\mathbf{y}^{(k)} - \tau\sigma A\mathbf{X}^{(k)}) = \mathbf{y}^{(k)} - \tau\sigma A\mathbf{X}^{(k)} - \tau\mathbf{b}$;
- $\mathbf{W}^{(k)} \leftarrow \mathbf{X}^{(k)} + A^*(2\mathbf{p}^{(k)} - \mathbf{y}^{(k)})$;
- $\mathbf{Q}^{(k)} \leftarrow (\text{Id} - \text{prox}_{\sigma^{-1}g})\mathbf{W}^{(k)} = \mathbf{W}^{(k)} - \mathbf{C} + \Pi_{\mathbb{S}^n_+}(\mathbf{C} - \mathbf{W}^{(k)})$;
- Set $\lambda^{(k)}$ in $]0;2[$;
- $\mathbf{y}^{(k+1)} \leftarrow \mathbf{y}^{(k)} + \lambda^{(k)}(\mathbf{p}^{(k)} - \mathbf{y}^{(k)})$;
- $\mathbf{X}^{(k+1)} \leftarrow \mathbf{X}^{(k)} + \lambda^{(k)}(\mathbf{Q}^{(k)} - \mathbf{X}^{(k)})$

$\mathbf{X} \leftarrow \sigma\mathbf{X}$;

6.2.2.5 Combettes-Eckstein algorithm

In this section, we still consider Problem (D) under the form (6.7) and we apply the idea from [CE16] to solve it. We first express it as the monotone inclusion problem (6.8)

$$\mathbf{0} \in \partial f(\mathbf{y}) + \partial(g \circ A^*)(\mathbf{y})\,. \tag{6.8}$$

Using the chain rule, we can expand equation (6.8) into

$$\mathbf{0} \in \partial f(\mathbf{y}) + A(\partial g)A^*(\mathbf{y})\,.$$

Its dual problem is as follows [BC11]

$$\mathbf{0} \in -A^*(\partial f^*)(-A\mathbf{X}) + \partial g^*(\mathbf{X})\,.$$

Applying the KKT conditions, we obtain

$$\begin{cases} -A(\mathbf{X}) \in \partial f(\mathbf{y}) \\ A^*(\mathbf{y}) \in \partial g^*(\mathbf{X})\,. \end{cases}$$

We call $\mathcal{Z}$ the set of points $(\mathbf{y}, \mathbf{X})$ in $\mathbb{R}^m \times \mathbb{S}^n$ that verifies those KKT conditions. Thus we have

$$\begin{cases} (\mathbf{y}, -A(\mathbf{X})) \in \operatorname{gra} \partial f \\ (A^*(\mathbf{y}), \mathbf{X}) \in \operatorname{gra} \partial g \,, \end{cases}$$

where gra denotes the graph of an operator and is defined in Appendix B. Since ∂f and ∂g are monotone operators, we have by definition

$$\begin{cases} (\forall (\mathbf{u}, \mathbf{v}) \in \operatorname{gra} \partial f) & \langle \mathbf{u} - \mathbf{y} \mid \mathbf{v} + A(\mathbf{X}) \rangle_{\mathbb{R}^m} & \geq 0 \\ (\forall (\mathbf{U}, \mathbf{V}) \in \operatorname{gra} \partial g) & \langle \mathbf{U} - A^*(\mathbf{y}) \mid \mathbf{V} - \mathbf{X} \rangle_{\mathbb{S}^n} & \geq 0 \,. \end{cases}$$

Thus,

$$\begin{aligned} & \langle \mathbf{u} - \mathbf{y} \mid \mathbf{v} + A(\mathbf{X}) \rangle_{\mathbb{R}^m} + \langle \mathbf{U} - A^*(\mathbf{y}) \mid \mathbf{V} - \mathbf{X} \rangle_{\mathbb{S}^n} \geq 0 \\ \iff \quad & \langle \mathbf{u} \mid \mathbf{v} \rangle_{\mathbb{R}^m} + \langle \mathbf{u} \mid A(\mathbf{X}) \rangle_{\mathbb{S}^n} - \langle \mathbf{y} \mid \mathbf{v} \rangle_{\mathbb{R}^m} - \langle \mathbf{y} \mid A(\mathbf{X}) \rangle_{\mathbb{R}^m} \\ & + \langle \mathbf{U} \mid \mathbf{V} \rangle_{\mathbb{S}^n} - \langle \mathbf{U} \mid \mathbf{X} \rangle_{\mathbb{S}^n} - \langle A^*(\mathbf{y}) \mid \mathbf{V} \rangle_{\mathbb{S}^n} + \langle A^*(\mathbf{y}) \mid \mathbf{X} \rangle_{\mathbb{S}^n} \geq 0 \\ \iff \quad & \langle \mathbf{u} \mid \mathbf{v} \rangle_{\mathbb{R}^m} + \langle \mathbf{U} \mid \mathbf{V} \rangle_{\mathbb{S}^n} \geq \langle \mathbf{y} \mid \mathbf{v} + A(\mathbf{V}) \rangle_{\mathbb{S}^n} + \langle \mathbf{X} \mid \mathbf{U} - A^*(\mathbf{u}) \rangle_{\mathbb{S}^n} \,. \end{aligned}$$

Let us set the following notation

- $\eta = \langle \mathbf{u} \mid \mathbf{v} \rangle_{\mathbb{R}^m} + \langle \mathbf{U} \mid \mathbf{V} \rangle_{\mathbb{S}^n} \in \mathbb{R}$
- $s = (\mathbf{v} + A(\mathbf{V}), \mathbf{U} - A^*(\mathbf{u})) \in \mathbb{R}^m \times \mathbb{S}^n$

The pair (η, s) hence defines a hyperplane in $\mathbb{R}^m \times \mathbb{S}^n$ whose normal is the vector $\mathbf{s}$. Note that η and $\mathbf{s}$ both depends on the points $(\mathbf{u}, \mathbf{v})$ and $(\mathbf{U}, \mathbf{V})$. This hyperplane divides the space in two half-spaces, one of which named H contains the set $\mathcal{Z}$

$$H = \{ (\mathbf{y}, \mathbf{X}) \in \mathbb{R}^m \times \mathbb{S}^n \mid \eta \geq \langle (\mathbf{y}, \mathbf{X}) \mid \mathbf{s} \rangle_{\mathbb{S}^n \times \mathbb{R}^m} \} \,.$$

Algorithm 12 follows the general Fejérian scheme exposed in Algorithm 11 using the above hyperplanes [Com01].

Algorithm 11: General Fejérian scheme

Input: Set $\mathcal{H}$ a Hilbert space and $\mathcal{E}$ a non-empty closed and convex set of $\mathcal{H}$
Input: Set $x^{(0)} \in \mathcal{H}$
Input: Set $\epsilon \in]0; 1[$ and $(\lambda_n)_{n \in \mathbb{N}} \in [\epsilon, 2 - \epsilon]^{\mathbb{N}}$
Output: x, a point in $\mathcal{E}$

1 **for** *k = 0,1,...* **do**
2 Generate an affine half space $H^{(k)}$ containing $\mathcal{E}$;
3 Project $x^{(k)}$ onto $H^{(k)}$: $p^{(k)} = \Pi_{H_n}(x^{(k)})$;
4 Perform over-relaxation: $x^{(k+1)} = x^{(k)} + \lambda^{(k)} \left(p^{(k)} - x^{(k)} \right)$;

It selects first two sequences of points $(\mathbf{u}^{(k)}, \mathbf{v}^{(k)})_{k \in \mathbb{N}}$ and $(\mathbf{U}^{(k)}, \mathbf{V}^{(k)})_{k \in \mathbb{N}}$ respectively in $\operatorname{gra} \partial f$ and $\operatorname{gra} \partial g$ before building the corresponding hyperplane defined by $\eta^{(k)}$ and $\mathbf{s}^{(k)}$. It then projects the current iterate $(\mathbf{y}^{(k)}, \mathbf{X}^{(k)})$ into the half-space $H^{(k)}$, with a potential over-relaxation. Algorithm 12 adds an extra subspace K that is used to limit the number of possible hyperplanes we could pick. The projection of $\mathbf{s}^{(k)}$ onto the subspace K is yet used as the normal to the hyperplane.

Algorithm 12: CombEck, Combettes-Eckstein algorithm to solve (6.7)

Input: Set $K = K_v \times K_m$ a closed vector subspace of $\mathbb{R}^m \times \mathbb{S}^n$ that contains the space of primal-dual solutions
Input: Set $(\mathbf{y}^{(0)}, \mathbf{X}^{(0)}) \in K_v \times K_m$
Input: Set $\epsilon \in]0;1[$ and $(\lambda_n)_{n\in\mathbb{N}} \in [\epsilon, 2-\epsilon]^{\mathbb{N}}$
Output: (y, X), a primal dual solution of (P) and (D)

for $k = 0,1,\ldots$ **do**
  Pick $(\mathbf{u}^{(k)}, \mathbf{v}^{(k)}) \in \mathrm{gra}(\partial f)$
    $\mathbf{u}^{(k)} \leftarrow \mathrm{prox}_{\gamma f}(\mathbf{y}^{(k)} - \gamma A\mathbf{X}^{(k)})$;
    $\mathbf{v}^{(k)} \leftarrow \gamma^{-1}(\mathbf{y}^{(k)} - \mathbf{u}^{(k)}) - A\mathbf{X}^{(k)}$;
  Pick $(\mathbf{U}^{(k)}, \mathbf{V}^{(k)}) \in \mathrm{gra}(\partial g)$
    $\mathbf{U}^{(k)} \leftarrow \mathrm{prox}_{\sigma g}(A^*\mathbf{y}^{(k)} + \sigma\mathbf{X}^{(k)})$;
    $\mathbf{V}^{(k)} \leftarrow \mathbf{X}^{(k)} + \sigma^{-1}(A^*\mathbf{y}^{(k)} - \mathbf{U}^{(k)})$;
  $s \leftarrow (\mathbf{v}^{(k)} + A(\mathbf{V}^{(k)}), \mathbf{U}^{(k)} - A^*(\mathbf{u}^{(k)}))$;
  $(\mathbf{t}^{(k)}, \mathbf{T}^{(k)}) \leftarrow \Pi_K(\mathbf{s}^{(k)})$;
  $\tau^{(k)} \leftarrow \|\mathbf{t}^{(k)}\|^2 + \|\mathbf{T}^{(k)}\|^2$;
  if $\tau^{(k)} > 0$ **then**
    $\eta^{(k)} \leftarrow \langle \mathbf{u}^{(k)} \mid \mathbf{v}^{(k)} \rangle_{\mathbb{R}^m} + \langle \mathbf{U}^{(k)} \mid \mathbf{V}^{(k)} \rangle_{\mathbb{S}^n}$;
    $\theta^{(k)} \leftarrow \frac{\lambda^{(k)}}{\tau^{(k)}} \max\left\{0, \langle (\mathbf{y}^{(k)}, \mathbf{X}^{(k)}) \mid (\mathbf{t}^{(k)}, \mathbf{T}^{(k)}) \rangle_{\mathbb{R}^m \times \mathbb{S}^n} - \eta^{(k)}\right\}$;
  else
    $\theta^{(k)} \leftarrow 0$;
  $\mathbf{y}^{(k+1)} \leftarrow \mathbf{y}^{(k)} - \theta^{(k)}\mathbf{t}^{(k)}$;
  $\mathbf{X}^{(k+1)} \leftarrow \mathbf{X}^{(k)} - \theta^{(k)}\mathbf{T}^{(k)}$;

§ 6.3 Proximal methods to solve altered forms

In order to try to accelerate the convergence of the algorithms, we modify the standard SDP problem by several means. In Sections 6.3.1, 6.3.2 and 6.3.4, we add a quadratic term respectively to the Lagrangian and to the dual objective function. The goal is here to accelerate the minimization of the target function while solving an equivalent problem. On the other hand, in Section 6.3.3, we relax the semi-definite constraints in order to ease the original optimization problem.

6.3.1 Augmented Lagrangian

The dual SDP problem (D) can be reformulated as a minimization problem with only linear constraints

$$(D) \iff \begin{cases} \min\limits_{(\mathbf{y},\mathbf{S})\in\mathbb{R}^m\times\mathbb{S}^n} & \langle -\mathbf{b} \mid \mathbf{y} \rangle_{\mathbb{R}^m} + \iota_{\mathbb{S}^n_+}(\mathbf{S}) \\ \text{s.t.} & A^*(\mathbf{y}) + \mathbf{S} = \mathbf{C}\,. \end{cases} \tag{6.9}$$

We build the augmented Lagrangian $\mathcal{L}_\rho$ of Problem (6.9) by adding to the standard Lagrangian a quadratic term controlled by a strictly positive real parameter ρ:

$$\mathcal{L}_\rho(\mathbf{y}, \mathbf{S}, \mathbf{X}) = \langle -\mathbf{b} \mid \mathbf{y} \rangle_{\mathbb{R}^m} + \iota_{\mathbb{S}^n_+}(\mathbf{S}) + \langle \mathbf{X} \mid A^*(\mathbf{y}) + \mathbf{S} - \mathbf{C} \rangle_{\mathbb{S}^n} + \frac{\rho}{2} \|A^*(\mathbf{y}) + \mathbf{S} - \mathbf{C}\|^2 \,.$$

Note that the extra quadratic term depends only on the conic constraint and does not affect the minimizer nor the minimum of Problem (6.9).

Let us compute its partial subdifferentials

$$\begin{aligned}\partial_{\mathbf{y}}\,\mathcal{L}_\rho(\mathbf{y},\mathbf{S},\mathbf{X}) &= -\mathbf{b}+A(\mathbf{X})+\rho A(A^*(\mathbf{y})+\mathbf{S}-\mathbf{C})\\ \partial_{\mathbf{S}}\,\mathcal{L}_\rho(\mathbf{y},\mathbf{S},\mathbf{X}) &= \mathcal{N}_{\mathbb{S}^n_+}(\mathbf{S})+\mathbf{X}+\rho(A^*(\mathbf{y})+\mathbf{S}-\mathbf{C})\\ \partial_{\mathbf{X}}\,\mathcal{L}_\rho(\mathbf{y},\mathbf{S},\mathbf{X}) &= A^*(\mathbf{y})+\mathbf{S}-\mathbf{C}\,.\end{aligned}$$

By writing the KKT conditions and splitting the operator, we obtain

$$\begin{pmatrix}\mathbf{0}\\ \mathbf{0}\\ \mathbf{0}\end{pmatrix} \in \left(\underbrace{\begin{bmatrix}-\mathbf{b}-\rho A\mathbf{C} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathcal{N}_{\mathbb{S}^n_+}-\rho\mathbf{C} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{C}\end{bmatrix}}_{=\mathcal{A}} + \rho \underbrace{\begin{bmatrix}AA^* & A & \mathbf{0}\\ A^* & \mathrm{Id} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{0}\end{bmatrix}}_{=\mathcal{B}_1} + \underbrace{\begin{bmatrix}\mathbf{0} & \mathbf{0} & A\\ \mathbf{0} & \mathbf{0} & \mathrm{Id}\\ -A^* & -\mathrm{Id} & \mathbf{0}\end{bmatrix}}_{=\mathcal{B}_2} \right) \begin{pmatrix}\mathbf{y}\\ \mathbf{S}\\ \mathbf{X}\end{pmatrix} \tag{6.10}$$

6.3.1.1 Douglas-Rachford algorithm

In Algorithm 13, we perform Douglas-Rachford splitting on operator $\mathcal{A}$ and $\mathcal{B}$ of Problem (6.10) where $\mathcal{B}=\mathcal{B}_1+\mathcal{B}_2$.

Algorithm 13: DougRachD+AL, Douglas-Rachford algorithm applied on (6.10)

Input: Set $(\mathbf{w}^{(0)},\mathbf{Z}^{(0)},\mathbf{V}^{(0)})\in\mathbb{R}^m\times\mathbb{S}^n\times\mathbb{S}^n$
Input: Set $\gamma\in\mathbb{R}_+$
Output: $\mathbf{X}$, solution of (P) and $(\mathbf{y},\mathbf{S})$, solution of (D)

1 **for** *k = 0,1,...* **do**

2 $$\begin{pmatrix}\mathbf{y}^{(k)}\\ \mathbf{S}^{(k)}\\ \mathbf{X}^{(k)}\end{pmatrix} \leftarrow J_{\gamma\mathcal{A}}\begin{pmatrix}\mathbf{w}^{(k)}\\ \mathbf{Z}^{(k)}\\ \mathbf{V}^{(k)}\end{pmatrix};$$

3 $$\begin{pmatrix}\mathbf{w}^{(k+1)}\\ \mathbf{Z}^{(k+1)}\\ \mathbf{V}^{(k+1)}\end{pmatrix} \leftarrow J_{\gamma\mathcal{B}}\left(2\begin{pmatrix}\mathbf{y}^{(k)}\\ \mathbf{S}^{(k)}\\ \mathbf{X}^{(k)}\end{pmatrix}-\begin{pmatrix}\mathbf{w}^{(k)}\\ \mathbf{Z}^{(k)}\\ \mathbf{V}^{(k)}\end{pmatrix}\right)+\begin{pmatrix}\mathbf{w}^{(k)}\\ \mathbf{Z}^{(k)}\\ \mathbf{V}^{(k)}\end{pmatrix}-\begin{pmatrix}\mathbf{y}^{(k)}\\ \mathbf{S}^{(k)}\\ \mathbf{X}^{(k)}\end{pmatrix};$$

The two resolvents $J_{\gamma\mathcal{B}}$ and $J_{\gamma\mathcal{A}}$ are given by (cf. Appendix C.2.1 for the detailed calculations)

$$J_{\gamma\mathcal{B}}\begin{pmatrix}\mathbf{w}\\ \mathbf{Z}\\ \mathbf{V}\end{pmatrix} = \begin{pmatrix}(\mathrm{Id}+\kappa AA^*)^{-1}\left(\mathbf{w}-\kappa A(\mathbf{Z})-\frac{1}{\gamma+\gamma^{-1}+\rho}A(\mathbf{V})\right)\\ \frac{1}{\gamma+\gamma^{-1}+\rho}\left(\mathbf{Z}+A^*(\mathbf{y})+(\gamma^{-1}+\rho)V\right)\\ -A^*\mathbf{y}+\gamma^{-1}(\mathbf{X}-\mathbf{V})\end{pmatrix}.$$

$$J_{\gamma\mathcal{A}}\begin{pmatrix}\mathbf{y}\\ \mathbf{S}\\ \mathbf{X}\end{pmatrix} = \begin{pmatrix}\mathbf{y}+\gamma(\mathbf{b}+\rho A(\mathbf{C}))\\ \Pi_{\mathbb{S}^n_+}(\mathbf{S}+\gamma\rho\mathbf{C})\\ \mathbf{X}-\gamma\mathbf{C}\end{pmatrix}$$

6.3.1.2 Forward-backward half-forward algorithm

In Problem (6.10), we notice that operator $\mathcal{B}_1$ is cocoercive, that operator $\mathcal{B}_2$ is Lipschitz continuous and monotone and that operator $\mathcal{A}$ is maximally monotone.

Indeed, we set $U=\begin{pmatrix}\mathbf{y}\\ \mathbf{S}\\ \mathbf{X}\end{pmatrix}$ and $W=\begin{pmatrix}\mathbf{w}\\ \mathbf{Z}\\ \mathbf{V}\end{pmatrix}$

- Cocoercivity of $\mathcal{B}_1$:

$$\begin{aligned}\|\mathcal{B}_1(U)-\mathcal{B}_1(W)\|^2 &= \|A(A^*(\mathbf{y}-\mathbf{w})+(\mathbf{S}-\mathbf{Z}))\|^2+\|A^*(\mathbf{y}-\mathbf{w})+(\mathbf{S}-\mathbf{Z})\|^2\\ &\leq (1+\|A\|^2)\,\|A^*(\mathbf{y}-\mathbf{w})+(\mathbf{S}-\mathbf{Z})\|^2\\ &\leq (1+\|A\|^2)\langle \mathcal{B}_1(U)-\mathcal{B}_1(W)\mid U-W\rangle_{\mathbb{R}^m\times\mathbb{S}^n\times\mathbb{S}^n}\end{aligned}$$

  Operator $\mathcal{B}_1$ is therefore cocoercive with a cocoercivity coefficient $(1+\|A\|^2)^{-1}$.

- Lipschitz continuity of $\mathcal{B}_2$:

$$\begin{aligned}\|\mathcal{B}_2(U)-\mathcal{B}_2(W)\|^2 &= \|A(\mathbf{X}-\mathbf{V})\|^2+\|(\mathbf{X}-\mathbf{V})\|^2+\|-A^*(\mathbf{y}-\mathbf{w})-(\mathbf{S}-\mathbf{Z})\|^2\\ &\leq (\|A\|^2+1)\,\|U-W\|^2\end{aligned}$$

  Operator $\mathcal{B}_2$ is $\sqrt{(1+\|A\|^2)}$-Lipschitz.

We thus apply the Forward-Backward Half-Forward Algorithm (FBHF) [BAD18] given in Appendix B on Problem (6.10) as shown in Algorithm 14. We choose the closed convex set $\mathcal{C}=\mathbb{R}^m\times\mathbb{S}^n_+\times\mathcal{D}$ to perform the projection, where $\mathcal{D}$ is the hyperplane defined in Section 6.2.2.2. According to Appendix B, the coefficient χ is here given by

$$\chi=\frac{4}{1+\|A\|^2+\sqrt{\|A\|^4+18\,\|A\|^2+17}}$$

Algorithm 14: FBHF applied on (6.10)

Input: Set $(\mathbf{y}^{(0)},\mathbf{S}^{(0)},\mathbf{X}^{(0)})\in\mathbb{R}^m\times\mathbb{S}^n\times\mathbb{S}^n$
Input: Set $\gamma\in[\epsilon;\chi-\epsilon]$
Output: $\mathbf{X}$, solution of (P) and $(\mathbf{y},\mathbf{S})$, solution of (D)

1 **for** *k = 0,1,...* **do**

2 $$\begin{pmatrix}\mathbf{y}^{(k+1)}\\ \mathbf{S}^{(k+1)}\\ \mathbf{X}^{(k+1)}\end{pmatrix}\leftarrow J_{\gamma\mathcal{A}}\left(\begin{pmatrix}\mathbf{w}^{(k)}\\ \mathbf{Z}^{(k)}\\ \mathbf{V}^{(k)}\end{pmatrix}-\gamma(\mathcal{B}_1+\mathcal{B}_2)\begin{pmatrix}\mathbf{w}^{(k)}\\ \mathbf{Z}^{(k)}\\ \mathbf{V}^{(k)}\end{pmatrix}\right);$$

3 $$\begin{pmatrix}\mathbf{w}^{(k+1)}\\ \mathbf{Z}^{(k+1)}\\ \mathbf{V}^{(k+1)}\end{pmatrix}\leftarrow \Pi_{\mathcal{C}}\left(\begin{pmatrix}\mathbf{y}^{(k+1)}\\ \mathbf{S}^{(k+1)}\\ \mathbf{X}^{(k+1)}\end{pmatrix}+\gamma\left(\mathcal{B}_2\begin{pmatrix}\mathbf{w}^{(k)}\\ \mathbf{Z}^{(k)}\\ \mathbf{V}^{(k)}\end{pmatrix}-\mathcal{B}_2\begin{pmatrix}\mathbf{y}^{(k+1)}\\ \mathbf{S}^{(k+1)}\\ \mathbf{X}^{(k+1)}\end{pmatrix}\right)\right);$$

The projection $\Pi_{\mathcal{C}}$ is simply

$$\Pi_{\mathcal{C}}\begin{pmatrix}\mathbf{y}\\ \mathbf{S}\\ \mathbf{X}\end{pmatrix}=\begin{pmatrix}\mathbf{y}\\ \Pi_{\mathbb{S}^+_n}(\mathbf{S})\\ \Pi_{\mathcal{D}}(\mathbf{X})\end{pmatrix}.$$

The resolvent $J_{\gamma\mathcal{A}}$ as computed in Section 6.3.1.1 and the projection onto $\mathcal{D}$ was given in Section 6.2.2.2.

6.3.1.3 ADMM and ADAL

Using (6.9), we reformulate Problem (D) into

$$(D)\iff(6.9)\iff\begin{cases}\min\limits_{(\mathbf{y},\mathbf{S})\in\mathbb{R}^m\times\mathbb{S}^n} & \langle-\mathbf{b}\mid\mathbf{y}\rangle_{\mathbb{R}^m}+\iota_{\mathbb{S}^n_+}\circ\tau_{\mathbf{C}}(\mathbf{Z})\\ \text{s.t.} & A^*(\mathbf{y})=\mathbf{Z}\end{cases}\tag{6.11}$$

where $\tau_{\mathbf{C}}$ is the affine application such that

$$(\forall \mathbf{Z} \in \mathbb{S}^n) \quad \tau_{\mathbf{C}}(\mathbf{Z}) = \mathbf{C} - \mathbf{Z}\,.$$

The augmented Lagrangian $\mathcal{L}_\rho$ of Problem (6.11) is given by

$$\begin{aligned}\mathcal{L}_\rho(\mathbf{y}, \mathbf{Z}, \mathbf{X}) &= \langle -\mathbf{b} \mid \mathbf{y} \rangle_{\mathbb{R}^m} + \iota_{\mathbb{S}^n_+} \circ \tau_{\mathbf{C}}(\mathbf{Z}) + \langle \mathbf{X} \mid A^*(\mathbf{y}) - \mathbf{Z} \rangle_{\mathbb{S}^n} + \frac{\rho}{2} \left\| A^*(\mathbf{y}) - \mathbf{Z} \right\|^2 \\ &= \langle -\mathbf{b} \mid \mathbf{y} \rangle_{\mathbb{R}^m} + \iota_{\mathbb{S}^n_+} \circ \tau_{\mathbf{C}}(\mathbf{Z}) + \frac{\rho}{2} \left\| A^*(\mathbf{y}) - \mathbf{Z} + \frac{\mathbf{X}}{\rho} \right\|^2 - \frac{1}{2\rho} \left\| \mathbf{X} \right\|^2 .\end{aligned}$$

In Algorithm 15, we apply Alternative Direction Method of Multipliers (ADMM) on the previous augmented Lagrangian $\mathcal{L}_\rho$, that is we first minimize $\mathcal{L}_\rho$ with respect to $(\mathbf{y}, \mathbf{Z})$ alternatively and then apply a gradient ascent with respect to $\mathbf{X}$. It is well-known ADMM is equivalent to Douglas-Rachford splitting applied on the dual [EB92]. The computation of the subproblems is done in Appendix C.2.2.

Algorithm 15: ADMM to solve problem (D)

Input: Set $\mathbf{X}^{(0)} \in \mathbb{S}^n$ and $\mathbf{Z}^{(0)} \in \mathbb{S}^n$
Input: Set $\rho \in \mathbb{R}_+^*$
Output: y: solution of the problem (D)

1 **for** $k = 0,1,\ldots$ **do**
2 $\quad \mathbf{y}^{(k+1)} \leftarrow (AA^*)^{-1} \left(A(\mathbf{Z}^{(k)} - \rho^{-1}\mathbf{X}^{(k)}) + \rho^{-1}\mathbf{b} \right)$;
3 $\quad \mathbf{Z}^{(k+1)} \leftarrow \mathbf{C} - \Pi_{\mathbb{S}^n_+}(\mathbf{C} - A^*(\mathbf{y}^{(k+1)}) - \rho^{-1}\mathbf{X}^{(k)})$;
4 $\quad \mathbf{X}^{(k+1)} \leftarrow \mathbf{X}^{(k)} + \rho(A^*(\mathbf{y}^{(k+1)}) - \mathbf{Z}^{(k+1)})$;

Improvement
Building the Gram matrix of the set $(\mathbf{A}_i)_{i \in [\![1,m]\!]}$ in the linear system at line 2 of Algorithm 15 and solving it directly may be costly when m is high. If the matrices $(\mathbf{A}_i)_{i \in [\![1,m]\!]}$ are sparse, sparse Cholesky factorisation and multi-frontal solvers can be used together with parallelisation techniques to speed-up the computation time [OCPB16, ZFP+17]. Iterative methods such has conjugate gradient method can also be used to solve the linear system in a fast and tractable manner [WGY10, YST15, OCPB16]. Moreover, empirical update of the parameter ρ can be used when stagnation is observed in order to accelerate the convergence [WGY10]. Those upgrades have been implemented in an alternative of ADMM named Alternative Direction Augmented Lagrangian method (ADAL) by Wen et al. [WGY10].

6.3.2 Augmented quadratic objective function

In order to speed up the convergence of the previous methods, we want to find a problem equivalent to (D) but with a quadratic objective function instead of a linear one. To derive such a problem, we start from the assumption that the dual variable $\mathbf{y}$ is bounded, which is always satisfied in practice:

$$(\exists K \in \mathbb{R}_+) \quad \|\mathbf{y}\| \leq K\,.$$

Then Cauchy-Schwarz inequality gives

$$\langle -\mathbf{b} \mid \mathbf{y} \rangle_{\mathbb{R}^m} \geq -\|\mathbf{b}\|\, K \quad \Longleftrightarrow \quad \langle -\mathbf{b} \mid \mathbf{y} \rangle_{\mathbb{R}^m} + \|\mathbf{b}\|\, K \geq 0\,.$$

Hence, the dual SDP problem (D) is equivalent, in the sense they have the same minimizer $\mathbf{y}$, to the following problem

$$\begin{cases} \underset{\mathbf{y}\in\mathbb{R}^m}{\text{minimize}} & \frac{1}{2}(\langle -\mathbf{b} \mid \mathbf{y}\rangle_{\mathbb{R}^m} + \|\mathbf{b}\| K)^2 \\ \text{subject to} & A^*\mathbf{y} + \mathbf{S} = \mathbf{C} \\ & \mathbf{S} \in \mathbb{S}^n_+ \end{cases} \tag{6.12}$$

We write problem (6.12) as in Section 6.2.2.2

$$\min_{(\mathbf{y},\mathbf{S})\in\mathbb{R}^m\times\mathbb{S}^n} \underbrace{\frac{1}{2}(\langle -\mathbf{b} \mid \mathbf{y}\rangle_{\mathbb{R}^m} + \|\mathbf{b}\| K)^2 + \iota_{\mathbb{S}^n_+}(\mathbf{S})}_{=f(\mathbf{y},\mathbf{S})} + \underbrace{\iota_{\{\mathbf{C}\}}(A^*\mathbf{y}+\mathbf{S})}_{=\iota_{\mathcal{D}}(\mathbf{y},\mathbf{S})}, \tag{6.13}$$

where $\mathcal{D} = \left\{(\mathbf{y},\mathbf{S}) \in \mathbb{R}^m \times \mathbb{S}^n \mid \begin{pmatrix} A^* & \mathrm{Id} \end{pmatrix} \begin{pmatrix} \mathbf{y} \\ \mathbf{S} \end{pmatrix} = \mathbf{C}\right\}$. We then apply Douglas-Rachford algorithm to Problem (6.13) as shown in Algorithm 16.

Algorithm 16: DougRach+Q, Douglas-Rachford algorithm applied on (6.13)

Input: Set $\mathbf{w}^{(0)} \in \mathbb{R}^m$
Input: Set $\mathbf{Z}^{(0)} \in \mathbb{S}^n$
Input: Set $\gamma \in \mathbb{R}_+$
Output: $(\mathbf{y},\mathbf{S})$, solution of (D)
for $k = 0,1,\ldots$ **do**
  $\begin{pmatrix} \mathbf{y}^{(k)} \\ \mathbf{S}^{(k)} \end{pmatrix} \leftarrow \Pi_{\mathcal{D}} \begin{pmatrix} \mathbf{w}^{(k)} \\ \mathbf{Z}^{(k)} \end{pmatrix}$;
  $\begin{pmatrix} \mathbf{w}^{(k+1)} \\ \mathbf{Z}^{(k+1)} \end{pmatrix} \leftarrow \mathrm{prox}_{\gamma f}\left(2\begin{pmatrix} \mathbf{y}^{(k)} \\ \mathbf{S}^{(k)} \end{pmatrix} - \begin{pmatrix} \mathbf{w}^{(k)} \\ \mathbf{Z}^{(k)} \end{pmatrix}\right) + \begin{pmatrix} \mathbf{w}^{(k)} \\ \mathbf{Z}^{(k)} \end{pmatrix} - \begin{pmatrix} \mathbf{y}^{(k)} \\ \mathbf{S}^{(k)} \end{pmatrix}$;

We have already computed the projection $\Pi_{\mathcal{D}}$ in Section 6.2.2.2. The new operator $\mathrm{prox}_{\gamma f}$ is given by (cf. Appendix C.2.3 for detailed computations)

$$\mathrm{prox}_{\gamma f}\begin{pmatrix} \mathbf{y} \\ \mathbf{S} \end{pmatrix} = \begin{pmatrix} \mathbf{y} - \frac{\gamma(\langle \mathbf{y}|\mathbf{b}\rangle_{\mathbb{R}^m} - \|\mathbf{b}\|K)\mathbf{b}}{1+\gamma\|\mathbf{b}\|^2} \\ \Pi_{\mathbb{S}^n_+}(\mathbf{S}) \end{pmatrix}.$$

6.3.3 Objective function with a barrier

6.3.3.1 Primal-dual Newton algorithm

Even if their iterations are cheap, the previous proposed proximal methods usually require a high number of such iterations before convergence occurs. In contrast, interior point methods converge after a very few iterations, which are however costly for medium and large scale problems.

In this section, we propose a second order method based on interior point methods. Hence, we relax the positive semi-definiteness constraint on the primal variable $\mathbf{X}$ of (P) by adding a barrier to the objective function

$$\begin{cases} \underset{\mathbf{X}\in\mathbb{S}^n}{\text{minimize}} & \langle \mathbf{C} \mid \mathbf{X}\rangle_{\mathbb{S}^n} + \mu\,\mathrm{ld}(\mathbf{X}) \\ \text{subject to} & A(\mathbf{X}) = \mathbf{b} \end{cases}. \tag{P_μ}$$

The barrier function is controlled by a strictly positive real parameter μ. We start with a given $\mu^{(0)}$, not too small in order to be able to solve Problem (P_μ). Afterwards, we decrease μ and use the solution of (P_μ) found with the previous value $\mu^{(l)}$ to solve (P_μ) with the new value $\mu^{(l+1)}$. As μ goes to zero, we obtain a solution of the initial problem (P).

Solving the inner problem (P_μ)
We write the Lagrangian of (P_μ)

$$\mathcal{L}_\mu(\mathbf{X}, \mathbf{y}) = \langle \mathbf{C} \mid \mathbf{X} \rangle_{\mathbb{S}^n} + \mu \operatorname{ld}(\mathbf{X}) + \langle \mathbf{y} \mid \mathbf{b} - A(\mathbf{X}) \rangle_{\mathbb{R}^m} ,$$

and we compute its gradient

$$\begin{aligned} \nabla_{\mathbf{X}} \mathcal{L}_\mu(\mathbf{X}, \mathbf{y}) &= \mathbf{C} - \mu \mathbf{X}^{-1} - A^*(\mathbf{y}) \\ \nabla_{\mathbf{y}} \mathcal{L}_\mu(\mathbf{X}, \mathbf{y}) &= \mathbf{b} - A(\mathbf{X}) . \end{aligned}$$

We write the stationary KKT conditions at optimality

$$\begin{aligned} \nabla_{\mathbf{X}} \mathcal{L}_\mu(\mathbf{X}, \mathbf{y}) &= \mathbf{0} \\ \nabla_{\mathbf{y}} \mathcal{L}_\mu(\mathbf{X}, \mathbf{y}) &= \mathbf{0} . \end{aligned}$$

We thereby look for the zeros of the gradient of the Lagrangian $\mathcal{L}_\mu$. We can reformulate it as the following inclusion problem

$$\mathbf{0} \in (\nabla_{\mathbf{X}} \mathcal{L}_\mu \times (-\nabla_{\mathbf{y}} \mathcal{L}_\mu))(\mathbf{X}, \mathbf{y}) . \tag{6.14}$$

Note that since the Lagrangian $\mathcal{L}_\mu$ is convex in X and concave in $\mathbf{y}$, we look at $-\nabla_{\mathbf{y}} \mathcal{L}_\mu$ instead of $\nabla_{\mathbf{y}} \mathcal{L}_\mu$. The operator $\nabla_{\mathbf{X}} \mathcal{L}_\mu \times (-\nabla_{\mathbf{y}} \mathcal{L}_\mu)$ is hence monotone and we apply Newton's iteration to solve (6.14), namely

$$\mathbf{z}^{(k+1)} = \mathbf{z}^{(k)} - \gamma^{(k)} (M^{(k)})^{-1} \nabla \mathcal{L}_\mu(\mathbf{z}^{(k)}) ,$$

where $z^{(k)} = (\mathbf{X}^{(k)}, \mathbf{y}^{(k)})$ is the primal-dual variable and $M^{(k)} = M_{(\mathbf{X}^{(k)}, \mathbf{y}^{(k)})}$ is the Hessian matrix with

$$M_{(\mathbf{X}, \mathbf{y})} = \begin{bmatrix} \mu \mathbf{X}^{-1} \otimes \mathbf{X}^{-1} & -A^* \\ A & 0 \end{bmatrix} .$$

Note that we can add a preconditioner $\lambda \operatorname{Id}$ to $M_{(\mathbf{X}, \mathbf{y})}$:

$$M_\lambda = \begin{bmatrix} \mu \mathbf{X}^{-1} \otimes \mathbf{X}^{-1} & -A^* \\ A & \lambda \operatorname{Id} \end{bmatrix} .$$

We compute the inverse of M_λ:

$$\begin{aligned} &(\forall (\mathbf{W}, \mathbf{v}) \in \mathbb{S}^n \times \mathbb{R}^m)(\forall (\mathbf{Z}, \mathbf{u}) \in \mathbb{S}^n \times \mathbb{R}^m) \\ &M_\lambda \begin{pmatrix} \mathbf{W} \\ \mathbf{v} \end{pmatrix} = \begin{pmatrix} \mathbf{Z} \\ \mathbf{u} \end{pmatrix} \implies \begin{cases} \mathbf{W} = (\mu^{-1} \mathbf{X} \otimes \mathbf{X})(\mathbf{Z} + A^*(\mathbf{v})) \\ \mathbf{v} = (A(\mu^{-1} \mathbf{X} \otimes \mathbf{X}) A^* + \lambda \operatorname{Id})^{-1} (\mathbf{u} - A(\mu^{-1} \mathbf{X} \otimes \mathbf{X}) \mathbf{Z}) \end{cases} \end{aligned}$$

Line-search
For the choice of the step size $\gamma^{(k)}$, we perform a line-search following [Sal17]

$$\gamma \left\| (M^{(k)})^{-1} (\nabla \mathcal{L}_\mu(J^{(k)}(\mathbf{z}^{(k)}, \gamma)) - \nabla \mathcal{L}_\mu(\mathbf{z}^{(k)})) \right\|_k \leq \delta \left\| J^{(k)}(\mathbf{z}^{(k)}, \gamma) - \mathbf{z}^{(k)} \right\|_k .$$

where $\|.\|_k$ is the norm associated to the scalar product $\langle . \mid M^{(k)} . \rangle$ and $J^{(k)}$ is defined as

$$J^{(k)}(\mathbf{z}^{(k)}, \gamma) = \mathbf{z}^{(k)} - \gamma (M^{(k)})^{-1} \nabla \mathcal{L}_\mu(\mathbf{z}^{(k)}) .$$

Note that we can reformulate the line-search criterion by squaring each side

$$\begin{aligned} \gamma^2 \langle \nabla \mathcal{L}_\mu(J^{(k)}) - \nabla \mathcal{L}_\mu(\mathbf{z}^{(k)}) \mid (M^{(k)})^{-1} (\nabla \mathcal{L}_\mu(J^{(k)}) - \nabla \mathcal{L}_\mu(\mathbf{z}^{(k)})) \rangle \\ \leq \delta^2 \langle J^{(k)} - \mathbf{z}^{(k)} \mid M^{(k)} (J^{(k)} - \mathbf{z}^{(k)}) \rangle . \end{aligned}$$

If the line-search criterion fails, we update γ by multiplying it with a parameter θ in $]0, 1[$. Typical values for θ and δ are respectively 0.9 and 0.7.

Outer loop and update of μ

The update of μ is a sensitive issue. Indeed, if the variation of μ is too quick, the algorithm will not converge. On the other hand, if its variation is too slow, the convergence will not happen in polynomial time and the method will not be applicable in practice. Moreover, the optimal update of μ depends on the current primal-dual solution of (P_μ).

Inspired by interior point method, we tried the three following criteria:

- $\mu^{(l+1)} = \mu^{(l)}\,(1-\tau)$ where τ is a fix parameter
- $\mu^{(l+1)} = \mu^{(l)}\left(1-\frac{0.1}{l}\right)$
- $\mu^{(l+1)} = (1-\alpha)\mu^{(l)}$ with $\alpha = \frac{2}{1+\sqrt{1+13\frac{n}{2}}}$ depending on the dimension n of the semi-definite constraint of (D).

The last option is more efficient but results are still far from the ones obtained from interior point methods where μ is updated according to the chosen search direction. Algorithm 17 shows the overall algorithm with the inner and outer loops.

Algorithm 17: Newton-PD, Newton algorithm to solve the barrier problem (P_μ)

Input: Set $(\mathbf{X}^{(0)}, \mathbf{y}^{(0)}) \in \mathbb{S}^n \times \mathbb{R}^m$
Input: Set $\mu^{(0)} \in \mathbb{R}_+^*$, $\gamma \in \mathbb{R}_+^*$, $\theta \in]0,1[$ and $\delta \in \mathbb{R}_+^*$
Output: $\mathbf{X}$, solution of (P) and $\mathbf{y}$, solution of (D)

for $l = 0,1,\ldots$ **do**
  $\mathbf{z}^{(0)} \leftarrow (\mathbf{X}^{(l)}, \mathbf{y}^{(l)})$;
  for $k = 0,1,\ldots$ **do**
    Compute $(M^{(k)})^{-1}$ and $\nabla\,\mathcal{L}_\mu(\mathbf{z}^{(k)})$;
    while $\gamma^2\langle \nabla\,\mathcal{L}_\mu(J^{(k)}) - \nabla\,\mathcal{L}_\mu(\mathbf{z}^{(k)}) \mid (M^{(k)})^{-1}(\nabla\,\mathcal{L}_\mu(J^{(k)}) - \nabla\,\mathcal{L}_\mu(\mathbf{z}^{(k)}))\rangle$
     $\leq \delta^2\langle J^{(k)} - \mathbf{z}^{(k)} \mid M^{(k)}(J^{(k)} - \mathbf{z}^{(k)})\rangle$ **do**
      $\gamma \leftarrow \theta\gamma$;
    $\gamma^{(k)} \leftarrow \gamma$;
    $\mathbf{z}^{(k+1)} \leftarrow \mathbf{z}^{(k)} - \gamma^{(k)}(M^{(k)})^{-1}\nabla\,\mathcal{L}_\mu(\mathbf{z}^{(k)})$;
    $(\mathbf{X}^{(l+1)}, \mathbf{y}^{(l+1)}) \leftarrow \mathbf{z}^{(k+1)}$;
  $\mu^{(l+1)} \leftarrow (1-\alpha)\mu^{(l)}$;

6.3.3.2 Dual Newton algorithm

We can perform the same steps of Section 6.3.3.1 but on the dual problem only. We first express the dual of (P_μ):

$$\max_{\mathbf{y}\in\mathbb{R}^m} \langle \mathbf{b} \mid \mathbf{y}\rangle_{\mathbb{R}^m} + \min_{\mathbf{X}\in\mathbb{S}^n} \langle \mathbf{C} \mid \mathbf{X}\rangle_{\mathbb{S}^n} + \mu\,\mathrm{ld}(\mathbf{X}) - \langle \mathbf{y} \mid A(\mathbf{X})\rangle_{\mathbb{R}^m}\,.$$

We solve the minimization with respect to $\mathbf{X}$ and we obtain

$$\mathbf{0} = \mathbf{C} - A^*(\mathbf{y}) - \mu\mathbf{X}^{-1} \iff \mathbf{X} = \mu(\mathbf{C} - A^*(\mathbf{y}))^{-1}\,.$$

So we have,

$$\begin{aligned}&\max_{\mathbf{y}\in\mathbb{R}^m} \langle \mathbf{b} \mid \mathbf{y}\rangle_{\mathbb{R}^m} + \mu\langle \mathbf{C} - A^*(\mathbf{y}) \mid (\mathbf{C} - A^*(\mathbf{y}))^{-1}\rangle_{\mathbb{S}^n} - \mu\log(\det(\mu(\mathbf{C} - A^*(\mathbf{y}))^{-1}))\\ &= n\mu(1-\log(\mu)) - \min_{\mathbf{y}\in\mathbb{R}^m} -\langle \mathbf{b} \mid \mathbf{y}\rangle_{\mathbb{R}^m} + \mu\log\det(\mathbf{C} - A^*(\mathbf{y}))\,.\end{aligned}$$

We set

$$\psi_\mu(\mathbf{y}) = -\langle \mathbf{b} \mid \mathbf{y} \rangle_{\mathbb{R}^m} + \mu \log\det(\mathbf{C} - A^*(\mathbf{y}))\,,$$

and compute its gradient and Hessian

$$\begin{aligned}\nabla\psi_\mu(\mathbf{y}) &= -\mathbf{b} - \mu A(\mathbf{C} - A^*(\mathbf{y}))^{-1}\\ \nabla^2\psi_\mu(\mathbf{y}) &= \mu A((\mathbf{C} - A^*(\mathbf{y}))^{-1} \otimes (\mathbf{C} - A^*(\mathbf{y}))^{-1})A^*\,.\end{aligned}$$

We finally apply Newton algorithm to solve the following dual problem $\arg\min_{\mathbf{y}\in\mathbb{R}^m} \psi_\mu(\mathbf{y})$:

$$\mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} - (A((\mathbf{C} - A^*(\mathbf{y}^{(k)}))^{-1} \otimes (\mathbf{C} - A^*(\mathbf{y}^{(k)}))^{-1})A^*)^{-1}(A(\mathbf{C} - A^*(\mathbf{y}^{(k)}))^{-1} - \mathbf{b}\mu^{-1})\,.$$

Afterwards, we can add a line-search and an update of parameter μ such as the ones of Section 6.3.3.1 as shown in Algorithm 18 where $J^{(k)}$ is defined similarly to Section 6.3.3.1:

$$J^{(k)}(\mathbf{y}^{(k)}, \gamma) = \mathbf{y}^{(k)} - \gamma(M^{(k)})^{-1}\nabla\psi_\mu(\mathbf{y}^{(k)})\,.$$

Algorithm 18: Newton-D, Newton algorithm to solve the dual problem of (P_μ)

Input: Set $\mathbf{y}^{(0)} \in \mathbb{R}^m$
Input: Set $\mu^{(0)} \in \mathbb{R}_+^*$, $\gamma \in \mathbb{R}_+^*$, $\theta \in]0,1[$ and $\delta \in \mathbb{R}_+^*$
Output: $\mathbf{y}$, solution of (D)

for $l = 0,1,\ldots$ **do**
  $\mathbf{y}_{\text{tmp}}^{(0)} \leftarrow \mathbf{y}^{(l)}$;
  for $k = 0,1,\ldots$ **do**
    $\mathbf{S}_{\text{inv}}^{(k)} \leftarrow (\mathbf{C} - A^*(\mathbf{y}_{\text{tmp}}^{(k)}))^{-1}$;
    $M^{(k)} \leftarrow (A\mathbf{S}_{\text{inv}}^{(k)} \otimes \mathbf{S}_{\text{inv}}^{(k)}A^*)^{-1}$;
    while
     $\gamma^2\langle \nabla\psi_\mu(J^{(k)}) - \nabla\psi_\mu(\mathbf{y}_{\text{tmp}}^{(k)}) \mid (M^{(k)})^{-1}(\nabla\psi_\mu(J^{(k)}) - \nabla\psi_\mu(\mathbf{y}_{\text{tmp}}^{(k)}))\rangle \leq$
     $\delta^2\langle J^{(k)} - \mathbf{y}_{\text{tmp}}^{(k)} \mid M^{(k)}(J^{(k)} - \mathbf{y}_{\text{tmp}}^{(k)})\rangle$ **do**
      $\gamma \leftarrow \theta\gamma$;
    $\gamma^{(k)} \leftarrow \gamma$;
    $\mathbf{y}_{\text{tmp}}^{(k+1)} \leftarrow \mathbf{y}_{\text{tmp}}^{(k)} - \gamma^{(k)}(M^{(k)})^{-1}(A\mathbf{S}_{\text{inv}}^{(k)} - \mathbf{b}(\mu^{(l)})^{-1})$;
    $\mathbf{y}^{(l+1)} \leftarrow \mathbf{y}_{\text{tmp}}^{(k+1)}$;
  $\mu^{(l+1)} \leftarrow (1-\alpha)\mu^{(l)}$;

6.3.4 Objective function with a regularization

We look at the following problem

$$\begin{cases}\underset{\mathbf{X}\in\mathbb{S}^n}{\text{minimize}} & \langle \mathbf{C} \mid \mathbf{X}\rangle_{\mathbb{S}^n} + \dfrac{\epsilon}{2}\|\mathbf{X}\|^2\\ \text{subject to} & A(\mathbf{X}) = \mathbf{b}\\ & \mathbf{X} \in \mathbb{S}_+^n\end{cases} \qquad (P_\epsilon)$$

where ϵ is a strictly positive real parameter.

In a similar way of Section 6.3.3, we first solve the inner problem (P_ϵ) and then update the parameter ϵ until it comes close to zero for a given tolerance.

Solving the inner problem (P_ϵ)
Let us write the dual of problem (P_ϵ):

$$\begin{aligned}&\max_{\mathbf{y}\in\mathbb{R}^m}\min_{\mathbf{X}\in\mathbb{S}^n}\langle\mathbf{C}\mid\mathbf{X}\rangle_{\mathbb{S}^n}+\frac{\epsilon}{2}\|\mathbf{X}\|^2+\iota_{\mathbb{S}^n_+}(\mathbf{X})+\langle\mathbf{y}\mid\mathbf{b}-A(\mathbf{X})\rangle_{\mathbb{R}^m}\\&=-\min_{\mathbf{y}\in\mathbb{R}^m}\left(\iota_{\mathbb{S}^n_+}+\frac{\epsilon}{2}\|.\|^2\right)^*(A^*(\mathbf{y})-\mathbf{C})-\langle\mathbf{b}\mid\mathbf{y}\rangle_{\mathbb{R}^m}\end{aligned}$$

We set the function ψ_ϵ such that

$$(\forall\mathbf{y}\in\mathbb{R}^m)\quad\psi_\epsilon(\mathbf{y})=\left(\iota_{\mathbb{S}^n_+}+\frac{\epsilon}{2}\|.\|^2\right)^*(A^*(\mathbf{y})-\mathbf{C})-\langle\mathbf{b}\mid\mathbf{y}\rangle_{\mathbb{R}^m},$$

and compute its gradient

$$(\forall\mathbf{y}\in\mathbb{R}^m)\quad\nabla\Psi_\epsilon(\mathbf{y})=-\mathbf{b}+A\Pi_{\mathbb{S}^n_+}\left(\frac{1}{\epsilon}(A^*(\mathbf{y})-\mathbf{C})\right).$$

We finally compute $\mathbf{y}$ by applying gradient descent algorithm:

$$\mathbf{y}^{(k+1)}=\mathbf{y}^{(k)}-\gamma^{(k)}\left(A\Pi_{\mathbb{S}^n_+}\left(\frac{1}{\epsilon}(A^*(\mathbf{y}^{(k)})-\mathbf{C})\right)-\mathbf{b}\right)$$

Note that an update of the algorithm to solve the inner problem would be to apply Newton's method instead of gradient descent. However, this requires to compute the derivative of the projection on the cone $\mathbb{S}^n_+$. Such derivative can be computed [MS06] but it raises the complexity of the problem.

Outer loop and issue
Finding a suitable updating rule for ϵ is a hard problem. Indeed, the decrease has to be fast enough to make the convergence occurs after a reasonable time. On the other hand, if the decrease is too fast, Problem (P_ϵ) may be hard to solve. We tried rules inspired from interior point methods but we did not succeed to obtain a suitable update rule for ϵ.

§ 6.4 Numerical comparisons

6.4.1 Set-up

We have run the simulations on a standard computer with an Intel I7 CPU running at 3.60 GHz and 32 GB of RAM. We used SeDuMi [Stu99] as implementation of interior point methods. Since interior point methods are more accurate than first-order methods, we use the value of the criterion returned by SeDuMi as our reference value.

We generate random SDP problems that have strong duality and that are not sparse. Unlike SeDuMi, our algorithms do not yet detect infeasibility nor exploit the sparsity of problem data. These features can be added following the examples of CDCS [ZFP+17] or SCS [OCPB16].

Solving linear systems
Many algorithms presented involve solving a linear system. We can solve it by caching LU decomposition of the linear operator since it is unchanged at each iteration. However we prefer performing a single step of conjugate gradient method which is cheaper for high-dimensional matrices as advised in [OCPB16]. Performing more steps of conjugate gradients methods per iteration does not really accelerate the global convergence of the algorithms.

Convergence and stopping criterion

For all the tested algorithms, we use the following stopping criterion

$$\frac{\xi^{(k+1)} - \xi^{(k)}}{\xi^{(k)}} \leq \epsilon\,,$$

where ϵ is a given tolerance and ξ is the variable output by the algorithm. Depending on algorithm, ξ may be either $\mathbf{X}$, $\mathbf{y}$, the concatenation of $\mathbf{X}$ and $\mathbf{y}$, of $\mathbf{y}$ and $\mathbf{S}$, of $\mathbf{X}$ and $\mathbf{y}$, of $\mathbf{y}$, $\mathbf{S}$ and $\mathbf{X}$ or of $\mathbf{Z}$ and $\mathbf{w}$.

Another possible criterion, which is not used here, is the DIMACS [WGY10] for primal-dual algorithms

$$\max\{\text{pinf}, \text{dinf}, \text{gap}\} \leq \epsilon\,,$$

where

- $\text{pinf} = \frac{\|A(\mathbf{X})-\mathbf{b}\|}{1+\|\mathbf{b}\|}$ is the renormalized primal infeasibility
- $\text{dinf} = \frac{\|\mathbf{C}-A^*(\mathbf{y})-\mathbf{S}\|}{1+\|\mathbf{C}\|}$ is the renormalized dual infeasibility
- $\text{gap} = \frac{|\langle \mathbf{b}|\mathbf{y}\rangle_{\mathbb{R}^m} - \langle \mathbf{C}|\mathbf{X}\rangle_{\mathbb{S}^n}|}{1+\langle \mathbf{b}|\mathbf{y}\rangle_{\mathbb{R}^m} + \langle \mathbf{C}|\mathbf{X}\rangle_{\mathbb{S}^n}}$ is the renormalized duality gap.

It measures the distance to both primal and dual feasible sets as well as the duality gap. In order to avoid extremely long running time, we set a maximal number of iterations to 100,000.

We list here the different algorithms to solve SDP problems developed in the previous sections

1. **ADMM**: ADMM on dual problem (D) (Section 6.3.1)
2. **ADAL**: ADAL, a variant **ADMM** with automatic update of the augmented Lagrangian's parameter and early exit of the algorithm when stagnation is detected (Section 6.3.1)
3. **CombEck**: Combettes-Eckstein algorithm applied on dual problem (D) (Section 6.2.2)
4. **ChamPockP**: Chambolle-Pock algorithm applied on primal problem (P) (Section 6.2.1)
5. **ChamPockP+Π**: Chambolle-Pock algorithm applied on primal problem (P) with an additional projection (Section 6.2.1)
6. **ChamPockD**: Chambolle-Pock algorithm applied on dual problem (D) (Section 6.2.2)
7. **DougRachP**: Douglas-Rachford algorithm applied on primal problem (P) (Section 6.2.1)
8. **DougRachP+I**: **DougRachP** where the two resolvents are interchanged (Section 6.2.1)
9. **DougRachD**: Douglas-Rachford algorithm on dual (D) (Section 6.2.2)

10. **DougRachD+AL**: Douglas-Rachford algorithm applied on the first order optimality condition of the augmented Lagrangian of the dual problem (D) (Section 6.3.1)

11. **DougRachD+Q**: Douglas-Rachford algorithm on transformed dual problem (D) to get a quadratic objective function (Section 6.3.2)

12. **FBHF**: FBHF applied on dual problem (D) without the projection on $\mathcal{C}$ (Section 6.3.1)

13. **FBHF+Π**: FBHF applied on dual problem (D) with the additional projection on $\mathcal{C}$ (Section 6.3.1)

14. **FISTA**: FISTA on dual problem (D) (Section 6.2.2)

15. **Newton-PD**: Newton method applied on primal-dual problem whose primal is (P) regularised with a barrier (Section 6.3.3)

16. **Newton-D**: Variant of **Newton-PD** where Newton method is only applied to he dual problem (Section 6.3.3)

17. **RFBPD**: RFBPD on dual problem (D) (Section 6.2.2)

18. **SGD-D**: Steepest gradient descent on the dual problem whose primal is (P) regularised with $\frac{\epsilon}{2}\|\mathbf{X}\|^2$ (Section 6.3.4).

6.4.2 Computational time versus SDP dimensions

We first compare the computational time of the different algorithms depending on the dimension of the SDP problem to investigate their scalability. Notice that we choose a value of m higher than n to foster first-order methods while insuring m is less than $\frac{n(n+1)}{2}$.

We discard **FISTA** from our tests since the nested loop calling another FISTA results into a very slow convergence as shown in Table 6.1. Moreover, we also discard **DougRachD**, Douglas-Rachford algorithm performed on the dual problem (D). Indeed it involves the resolution of a linear system of size $n^2 \times n^2$ which is untractable in high dimension.

Table 6.1: *Computational time of **FISTA** ($\epsilon = 10^4$)*

n	m	Computational time
3	2	0.190s
5	3	0.260s
5	7	1.588s
8	5	1.646s
8	10	2.272s
8	27	5.173s
30	20	>**12**h

Tables 6.2 and 6.3 show the running time, the number of iterations as well as the value of the objective function for the different algorithms. Conversely to interior point methods, the iterates of proximal algorithms are not necessarily feasible points. As a consequence, we observe some the proximal algorithms return a value of the criterion lower than the true minimizer as they have not converged yet. Moreover, we notice that some algorithms are extremely slow to converge and prohibit their use in higher dimensional SDP. Especially, we observe that **CombEck**, **ChamPockP**, **ChamPockP+Π**, **FBHF** and **FBHF+Π** have not converged yet after reaching the maximum number of iterations. Moreover, adding an extra projection in **ChamPock+Π** and **FBHF+Π** does not help to speed up the convergence. On the other hand, **ADAL**, **ADMM** and **DougRachP+I** are extremely fast and beat the interior point method. We therefore focus on those three algorithms in the following.

Table 6.2: *Computational time of the different algorithms ($\epsilon = 10^{-4}$, $n = 50$)*

	$m = 70$			$m = 600$		
	time (s)	iter	obj	time (s)	iter	obj
ADAL	0.382	363	-38.423475	0.498	141	841.807956
ADMM	0.146	135	-38.406085	0.874	248	841.807956
RFBPD	12.349	6461	-38.425691	6.252	1116	841.976605
ChamPockP	99.712	**100000**	-49.821443	206.492	**100000**	1581.347600
ChamPockP+Π	94.104	**100000**	-49.821443	155.350	**100000**	1581.347603
ChamPockD	14.475	15480	-38.430796	17.782	11080	841.828828
DougRachP	0.101	68	-38.374347	3.527	148	841.584556
DougRachP+I	0.077	63	-38.297025	0.724	56	841.773766
DougRachD+AL	1.379	1027	-37.796357	4.590	296	842.028312
DougRachD+Q	4.805	3227	-38.436199	154.092	6446	833.078498
FBHF	134.967	**100000**	-37.127093	353.559	26948	843.035183
FBHF+Π	136.501	**100000**	-37.127093	352.529	26948	843.035183
CombEck	211.599	**100000**	-38.547321	690.991	**100000**	-883.063651
SeDuMi	0.127	9	-38.433519	2.082	7	841.801410

Table 6.4 shows the same information for higher dimensional SDP problems. We first confirm that when n is bigger than m, interior point method is the fastest algorithm as predicted. Then, we observe that the three first-order methods scale well with m as opposed to **SeDuMi**. However m needs to be much larger than n to make first-order faster than interior point methods; if m is only slightly larger, interior point methods still converge faster.

The results also show that in lower dimensions, **DougRachP+I** is the fastest algorithm albeit it gives the looser criterion. It is due to the fact that Douglas-Rachford algorithm iterates on an auxiliary variable and not on the variable of interest. Nevertheless, in high dimensions, **ADMM** yields faster convergence and more accurate solution.

We finally remark that **SeDuMi** has a very low number of iterations compared to first-order methods. It is a well-known property of interior point methods; they converge in a very low number of iterations but each iteration may be expensive depending on the size of m.

Table 6.3: *Computational time of the different algorithms ($\epsilon = 10^{-4}$, $n = 80$)*

	$m = 100$			$m = 800$		
	time (s)	iter	obj	time (s)	iter	obj
ADAL	2	705	49.493843	1	86	718.360621
ADMM	1	305	49.492202	1	86	718.375590
RFBPD	103	16677	49.494452	65	3009	718.518761
ChamPockP	290	**100000**	15.884938	793	**100000**	909.639582
ChamPockP+Π	274	**100000**	15.884938	663	**100000**	909.639615
ChamPockD	84	30553	49.503565	284	46304	718.373913
DougRachP	1	103	49.669413	8	77	718.380938
DougRachP+I	<1	71	49.712329	3	65	718.257126
DougRachD+AL	6	1172	50.684067	21	343	719.396799
DougRachD+Q	23	3660	49.465634	518	4756	718.120190
FBHF	13	2461	117.548962	263	4808	1229.568565
FBHF+Π	13	2461	117.548962	262	4808	1229.568565
CombEck	661	**100000**	49.379735	2926	**100000**	714.006605
SeDuMi	1	9	49.484113	9	7	718.363112

Table 6.4: *Computational time of the fastest algorithms ($\epsilon = 10^{-4}$)*

	ADAL			**ADMM**			**DougRachP+I**			**SeDuMi**		
	time (s)	iter	obj	time (s)	iter	obj	time (s)	iter	obj	time (s)	iter	obj
$\boldsymbol{n = 80}$												
$m = 50$	5	948	168.87	2	421	168.86	1	83	170.56	<1	8	168.84
$m = 100$	3	431	330.11	1	192	330.11	1	68	330.08	1	7	330.04
$m = 450$	**1**	64	564.05	**1**	78	564.03	**2**	67	564.07	**5**	7	564.03
$m = 800$	**2**	84	-887.60	**2**	86	-887.62	**4**	60	-887.48	**15**	7	-887.63
$\boldsymbol{n = 300}$												
$m = 200$	644	3293	-963.08	133	680	-963.09	29	108	-970.29	39	9	-963.27
$m = 500$	394	1333	7453.38	110	350	7452.75	69	115	7357.07	176	9	7451.81
$m = 1500$	**119**	206	2195.97	**126**	220	2189.42	**211**	148	2083.70	**807**	8	2189.06
$m = 4000$	**340**	244	7820.09	**185**	132	7821.39	**758**	146	7786.32	**5125**	8	7821.67

6.4.3 Computational time versus precision

We are now interested by the impact of the desired accuracy ϵ on the stopping criterion. Indeed, the accuracy of the solution returned by the SDP solver is critical for SDP relaxations of rational optimization problems [WNM11, LM18]. Table 6.5 compares the computational time and the value of the objective function for different values of precision. We first notice that even with a low desired accuracy, **SeDuMi** is often highly precise. It is another well-known property of interior point methods. For first-order methods, we need to set at least an accuracy of 10^{-3} or 10^{-4} to be close to a minimum. Furthermore, increasing precision is more and more costly when we approach a minimum. This effect is even amplified with the slowest first-order methods as shown in Table 6.6. Here, we notice that increasing the precision blows up the number of iterations.

Table 6.5: *Impact of precision on running time*

	ADAL			**ADMM**			**DougRachP+I**			**SeDuMi**		
	time (s)	iter	obj	time (s)	iter	obj	time (s)	iter	obj	time (s)	iter	obj
$(n,m) = (300,200)$												
$\epsilon = 10^{-1}$	3	23	-851.40	3	23	-851.40	1	3	-1243.83	20	5	-971.47
$\epsilon = 10^{-2}$	10	76	-910.07	18	139	-942.08	1	4	-1193.42	27	7	-963.29
$\epsilon = 10^{-3}$	262	1693	-961.29	58	385	-961.53	4	15	-1200.92	39	8	-963.28
$\epsilon = 10^{-4}$	644	3293	-963.08	133	680	-963.09	29	108	-970.29	39	9	-963.27
$\epsilon = 10^{-5}$	792	5226	-963.25	150	1044	-963.25	57	215	-961.96	47	10	-963.27
$\epsilon = 10^{-6}$	1121	7258	-963.27	253	1447	-963.27	144	467	-963.30	55	12	-963.27
$(n,m) = (300,500)$												
$\epsilon = 10^{-1}$	5	15	7648.86	5	15	7648.86	3	3	6850.18	83	4	52106.29
$\epsilon = 10^{-2}$	11	44	7512.45	11	44	7512.45	3	4	7034.06	124	6	7448.59
$\epsilon = 10^{-3}$	31	106	7483.64	45	160	7461.44	9	16	6864.28	169	8	7451.78
$\epsilon = 10^{-4}$	394	1333	7453.38	110	350	7452.75	69	115	7357.07	176	9	7451.81
$\epsilon = 10^{-5}$	729	2496	7451.97	180	604	7451.88	140	231	7448.16	190	10	7451.82
$\epsilon = 10^{-6}$	1044	3575	7451.84	254	887	7451.83	248	398	7451.56	167	11	7451.82
$(n,m) = (300,1500)$												
$\epsilon = 10^{-1}$	10	17	2305.30	10	17	2305.30	9	3	620.82	411	4	31525.00
$\epsilon = 10^{-2}$	21	36	2232.00	21	36	2232.00	11	4	1594.80	609	6	2188.62
$\epsilon = 10^{-3}$	44	76	2204.50	51	88	2197.17	30	18	551.62	706	7	2188.94
$\epsilon = 10^{-4}$	119	206	2195.97	126	220	2189.42	211	148	2083.70	807	8	2189.06
$\epsilon = 10^{-5}$	825	1435	2189.11	198	345	2189.09	307	215	2185.00	903	9	2189.06
$\epsilon = 10^{-6}$	1187	2068	2189.07	270	471	2189.07	413	292	2188.71	1020	10	2189.07
$(n,m) = (300,4000)$												
$\epsilon = 10^{-1}$	40	29	6896.46	41	29	6896.46	59	4	7116.89	2628	4	2628.44
$\epsilon = 10^{-2}$	70	51	7771.71	74	53	7772.72	62	5	7076.62	3910	6	7821.25
$\epsilon = 10^{-3}$	122	89	7808.27	120	87	7817.89	63	5	7076.62	4644	7	7821.53
$\epsilon = 10^{-4}$	340	244	7820.09	185	132	7821.39	758	146	7786.32	5125	8	7821.67
$\epsilon = 10^{-5}$	508	376	7821.22	553	186	7821.66	888	176	7819.83	6257	9	7821.68
$\epsilon = 10^{-6}$	800	582	7821.61	336	243	7821.68	1073	210	7821.52	7541	11	7821.68

Table 6.6: *Number of iterations for different values of precision ($(n,m) = (80,800)$)*

	RFBPD	**ChamPockD**	**DougRachD+Q**	**FBHF**
$\epsilon = 10^{-3}$	2213	4458	40	887
$\epsilon = 10^{-4}$	3902	46566	41	4752
$\epsilon = 10^{-5}$	5752	88073	1862	41533

6.4.4 Comparison on SDP relaxations from polynomial optimization

We now compare our fastest methods on SDP problems which are relaxations of the unconstrained minimization of random polynomials on 10 variables. Those problems are generated from the code given in [HM11]. We compare our methods with the augmented Lagrangian method proposed by [YST15] and implemented in the software SDPNAL.

Results are presented in Tables 6.7 and 6.8 where we have fixed the relaxation order to 3 to compare later with the case of constrained minimization. Note that the maximum number of iterations has been set to 5000. We observe that **ADAL**, our fastest solver, shows similar performance with **SDPNAL**. Moreover, both have a shorter computational time than the interior point methods **SeDuMi**.

However, when constraints are added in the polynomial optimization problem, the computational times of **SDPNAL** and our first order methods both explode and become higher than the one of **SeDuMi**. This is illustrated in Table 6.9 where we add some polynomial constraints to the previous considered problems and increase the maximum number of iterations to 20000. We observe that none of the proximal methods have converged before reaching the maximum number of iterations. In this case, both dimensions n and m are high as explained in the complexity study of Appendix A. Indeed, in the unconstrained case, we had $m = 8008$ and $n = 286$ while in this constrained case, we still have $m = 8088$ but now $n = 1606$.

Table 6.7: *Comparison of running time with SDPNAL for SDP problems coming from relaxations of quadratic polynomial optimization*

	Time (s)	Number of iterations	Optimal value
SeDuMi	**0.656**	7	$8.75 \cdot 10^{-3}$
ADAL	**0.140**	123	$8.74 \cdot 10^{-3}$
DougRachP	0.306	94	$8.75 \cdot 10^{-3}$
DougRachD	0.318	93	$8.75 \cdot 10^{-3}$
DougRach+I	0.813	82	$8.75 \cdot 10^{-3}$
ChamPock+AL	0.597	362	$8.75 \cdot 10^{-3}$
SDPNAL	**0.140**	102	$8.75 \cdot 10^{-3}$

Table 6.8: *Comparison of running time with SDPNAL for SDP problems coming from relaxations of cubic polynomial optimization*

	Time (s)	Number of iterations	Optimal value
SeDuMi	**319**	6	$6.27 \cdot 10^{-4}$
ADAL	**3**	367	$6.26 \cdot 10^{-4}$
DougRachP	95	286	$6.11 \cdot 10^{-4}$
DougRachD	92	282	$6.18 \cdot 10^{-4}$
DougRach+I	204	53	$6.27 \cdot 10^{-4}$
ChamPock+AL	20	612	$6.03 \cdot 10^{-4}$
SDPNAL	**1**	101	$6.27 \cdot 10^{-4}$

§ 6.5 Summary

From our study of proximal algorithms to solve large-scale SDP problems, we make the following conclusions: (i) First-order methods are only interesting to solve SDP problems when m is much higher than n and when n is small enough. Indeed, we have seen that when both n and m are high, interior point methods are more efficient. Conversely, the considered methods have a slow convergence that requires a high number of iterations.

Table 6.9: *Comparison of running time with SDPNAL for SDP problems coming from relaxations of cubic polynomial optimization with* 20 *polynomial constraints*

	Time (s)	Number of iterations	Optimal value
SeDuMi	**1851**	25	$\mathbf{-3.29 \cdot 10^4}$
ADAL	**1588**	**20000**	$\mathbf{-4.64 \cdot 10^3}$
DougRachP	13144	**20000**	$-1.20 \cdot 10^1$
DougRachD	12100	**20000**	$-1.20 \cdot 10^1$
DougRach+I	175008	**20000**	$\mathbf{-2.15 \cdot 10^4}$
ChamPock+AL	2304	**20000**	-6.43
SDPNAL	**2272**	**20000**	$\mathbf{-8.85 \cdot 10^6}$

This may be due to the angle between the cone and of positive semidefinite matrix with the hyperplane of the linear constraints which makes them almost parallel. Therefore, we have shown in this section that interior point methods are the most efficient to solve constrained rational optimization problems such as the ones considered in the other chapters. (ii) First-order methods converge slowly when the desired precision increases. This point can be an issue for application to rational optimization as the extraction algorithm is highly sensitive to noise. This point will be discussed in more details in Chapter 7. (iii) Among the considered algorithms, ADMM and ADAL are the most efficient ones.

- Chapter 7 -

Tensor decomposition and moment problem

Symmetric tensors are closely related to homogeneous polynomials: for instance, a decomposition of a symmetric tensor can be seen as the decomposition of a homogeneous polynomial as a sum of powers of linear forms [BCMT10]. In this chapter, we propose to interpret the extraction step of rational optimization as the decomposition of a symmetric tensor. Both are instances of the same moment problem, i.e. searching for an R-atomic measure supported on a compact set, some of its moments being known. The goal of this chapter is to explore this connection in both directions.

§ 7.1 Background

Tensors are useful mathematical tools in a wide range of scientific areas. The diffusion kurtosis tensor is used in medical imaging and the elasticity tensor in physics. Tensors play also important roles in image authenticity validation, in crystal study, or in quantum physics [BAC+15, QCC18]. At the origin of the wide spread of tensors stand tensor decompositions, which have become fundamental operations in today's science and engineering. Indeed, due to their high dimensionality, tensors remain difficult to visualize, to handle, and to interpret. More precisely, tensor factorizations aim at decomposing an intricate or large tensor into a sum of smaller and simpler atoms that are more easily interpreted and lead to faster computations. Several decompositions have been proposed for different applications such as the Canonical Polyadic Decompostion (CPD) or Tucker decomposition [KB09, SLF+17] as well as tensor-train [Ose11] or block-term [SBL13] decompositions.

The focus of this chapter is on CPD. The current methods for performing this decomposition include mainly optimization techniques such as alternating least squares (ALS) [SLF+17], nonlinear least squares (NLS) [SLF+17], and unconstrained non-linear optimization (OPT) [ADK11, SBL12]. A few algebraic methods have also been proposed such as the generalized eigenvalue method (GEVD) [SK90] or methods based on the decomposition of a homogeneous polynomial into a sum of given powers of linear forms [CGLM08].

Moreover, in order to perform the decomposition, all the mentioned methods require to know explicitly the tensor rank, i.e. the number of atoms in the CPD. Yet, finding the true rank is difficult, as it is known to be an NP-hard problem [HL13]. Furthermore, any error in the sought rank can yield dramatic consequences, as the set of tensors of given rank does not form a closed set. Common rank estimation methods are based on optimization problems using the nuclear norm as a surrogate for the rank [RFP10], Bayesian models [ZZZ+16], or on matrix unfoldings such as balanced matricization [SLF+17].

In this chapter, we deal more specifically with the CPD of a symmetric tensor, which has application in blind identification of under-determined mixtures [CGLM08] for instance. Applications of symmetric tensors to machine learning can also be found in [AGH+14] and applications to other areas in [QCC18]. Note that, for symmetric tensors, one can introduce the notion of symmetric rank, which is also NP-hard to determine [HL13] but has the benefit to be computable by some existing algebraic methods [BGI11]. Although, the rank and the symmetric rank may be different [Shi18], they are equal in many cases.

We first propose an alternative moment-based approach for CPD that offers theoretical guarantees to determine the symmetric rank as well as a method to recover the generating vectors of the CPD of a symmetric tensor. Our method has the advantage to provide a necessary and sufficient condition to obtain the true rank whereas for instance, rank-revealing matrix unfoldings [SLF+17] give only a sufficient condition. Note that the authors in [TS15] suggest also to reformulate the minimum rank CPD problem as a generalized problem of moments. Nevertheless, their approach is different as the moment problem is relaxed into a hierarchy of convex semi-definite programming problems using Lasserre's hierarchy, the dual problem of which provides certificates for the correctness of the CPD. Experiments are restricted to small dimensional tensors, which, from our experience, is related to the heavy computational load of Lasserre's hierarchy. In contrast, our proposed method scales well for medium to high dimensional tensors. On the other hand, the extraction of the generating vectors is grounded on methods for solving polynomial systems based on algebraic results, which allow us to solve a moment problem. Theoretical results guarantee that the retrieved CPD is exact. The algebraic methods in [BBCM13, Nie15, Nie17] yield similar result but provide a completely different perspective based on a sum of given powers of linear forms that requires a heavy theoretical background. Although similar in spirit, the simplicity of our method may provide further insight and accessibility.

Secondly, we adopt the reverse viewpoint: we use CPD algorithms to solve a moment problem. This is especially of interest in the context of rational and polynomial optimization while performing the extraction of the global optima. Indeed, we show here that this extraction is equivalent to performing a CPD on a specific tensor. The extraction method [HL05] used so far and recalled in Appendix D, shows good performance when the truncated moment vector is estimated with a high accuracy and has a sufficient size. However, if the recovered moments are subject to a small perturbation, it yields inexact results. We thus use the prolific literature on CPD to propose an extraction method which is more robust to noise as shown experimentally. An additional benefit of using those CPD algorithms is the reduction in dimensions of the relaxation to be solved, which results in a lighter computational load.

This chapter is organized as follows: Section 7.2 recalls basic definitions for tensors and their CPD. Section 7.3 reformulates the CPD of a symmetric tensor as a moment problem and reviews the main tools to solve the latter problem. Based on these tools, Section 7.4 derives a necessary and sufficient condition to determine the rank of a symmetric tensor illustrated by a practical signal processing application, while Section 7.5 proposes a new algebraic CPD algorithm. Section 7.6 expresses the extraction method of Appendix D as the CPD of a symmetric tensor. It then exposes a robust extraction algorithm for rational optimization problems based on tensors decomposition algorithms. The benefits of our proposed approach are exposed numerically on a few examples.

§ 7.2 Tensor decomposition

We consider in this chapter tensors as multidimensional arrays. The dimension d of those arrays is called the order of the tensor. A tensor $\mathcal{T}$ is hence an element of the product of d vector spaces that can be indexed using a d-tuple $\mathbf{i} = (i_1, \dots, i_d)$. We consider here only tensors that are defined on the product of d real spaces $\mathbb{R}^{n+1}$. The entries of the tensor are thereby denoted by $(\mathcal{T}_{i_1,\dots,i_d})_{0\leq i_1,\dots,i_d\leq n}$. Matrices are a special case of tensor of order 2. Moreover, in this chapter, we deal with symmetric tensors only, i.e. tensors whose entries $(\mathcal{T}_{i_1,\dots,i_d})_{0\leq i_1,\dots,i_d\leq n}$ are unchanged by any permutation of the indices.

7.2.1 Canonical polyadic decomposition

Let $\mathcal{T}$ denote a tensor of order d on $\mathbb{R}^{n+1}$ with d an even integer higher than or equal to 4. A tensor is said to be symmetric rank-1 if it can be expressed as

$$\mathbf{v}^{\otimes d} = \underbrace{\mathbf{v} \otimes \cdots \otimes \mathbf{v}}_{d \text{ times}}$$

for a vector $\mathbf{v} = (v_i)_{i\in[\![0,n]\!]}$ of $\mathbb{R}^{n+1}$, that is $\left[\mathbf{v}^{\otimes d}\right]_{i_1,\dots,i_d} = v_{i_1} \dots v_{i_d}$.

The CPD problem that we consider consists in finding a decomposition of $\mathcal{T}$ into a finite sum of rank-1 tensors, $\mathcal{T} = \sum_{r=1}^{R} \mathbf{v}(r)^{\otimes d}$, or equivalently

$$\mathcal{T}_{i_1,\dots,i_d} = \sum_{r=1}^{R} v_{i_1}(r) \dots v_{i_d}(r)\,. \tag{7.1}$$

The symmetric rank[1], denoted by $\operatorname{rank}_S \mathcal{T}$ is the minimum number of terms required in any representation of $\mathcal{T}$ as above. To determine the CPD of $\mathcal{T}$, we first detect its rank R and then look for an approximation of rank R, that is we determine the vectors $(\mathbf{v}(r))_{r\in[\![1,R]\!]}$. Notice that, since d is strictly higher than 2, Decomposition (7.1) is unique up to the order of the generating vectors $(\mathbf{v}(r))_{r\in[\![1,R]\!]}$ as well as their sign. This is in contrast with the matrix case [SLF+17].

7.2.2 Dehomogenization

We make the following assumption, which is not restrictive in most applications,

Assumption 1. $(\exists l \in [\![0,n]\!])(\forall r \in [\![1,R]\!]) \quad v_l(r) \neq 0.$

Thence, by normalizing each vector $\mathbf{v}(r)$ with its l^{th} coordinate, Decomposition (7.1) can be expressed in the equivalent form

$$\mathcal{T} = \sum_{r=1}^{R} \lambda_r \left(\frac{\mathbf{v}(r)}{v_l(r)}\right)^{\otimes d} = \sum_{r=1}^{R} \lambda_r \mathbf{u}(r)^{\otimes d}\,, \tag{7.2}$$

where $\mathbf{u}(r) = \left(\frac{v_1(r)}{v_l(r)}, \dots, \frac{v_{l-1}(r)}{v_l(r)}, \frac{v_{l+1}(r)}{v_l(r)}, \dots, \frac{v_n(r)}{v_l(r)}\right)$ is the dehomogenization of $\mathbf{v}(r)$ and $\lambda_r = v_l(r)^d$ is a coefficient that is positive since d is even. The coordinate index $l \in [\![0,n]\!]$ used for the above normalization is the same for all r. With no loss of generality, we take $l = 0$ in the following but all the results still hold for any other index l after an adequate permutation of coordinates.

[1] In the following, rank will systematically mean symmetric rank.

7.2.3 Re-indexing of the tensor elements

Due to the symmetry assumption, the order of the indices in $\mathbf{i} = (i_1, \dots, i_d)$ has no influence on the value of the tensor elements $\mathcal{T}_{i_1,\dots,i_d}$, which are uniquely defined by specifying the number of times each index value appears in $\mathbf{i}$. More precisely, to any d-tuple $\mathbf{i} = (i_1, \dots, i_d)$, we associate an n-tuple $\boldsymbol{\alpha}(\mathbf{i}) = (\alpha_1(\mathbf{i}), \dots, \alpha_n(\mathbf{i}))$ of $\mathbb{N}_d^n$, where for each j in $[\![1, n]\!]$, $\alpha_j(\mathbf{i})$ is the number of times the index value j appears in $\mathbf{i}$. Note that, since the order of the tensor is d, the number of times the index 0 appears is uniquely defined by $\boldsymbol{\alpha}$ through $d - |\boldsymbol{\alpha}|$. We therefore index our tensor with $\boldsymbol{\alpha}$ in $\mathbb{N}_d^n$ instead of $\mathbf{i}$ and define the tensor values

$$\mathcal{T}_{\mathbf{i}} = \mathfrak{T}_{\boldsymbol{\alpha}(\mathbf{i})} \,.$$

In the following, we simply write $\mathfrak{T}_{\boldsymbol{\alpha}}$ instead of $\mathfrak{T}_{\boldsymbol{\alpha}(\mathbf{i})}$ for readability.

Example Let us take a tensor $\mathcal{T}$ of order 4 in $\mathbb{R}^3$ ($d = 4$ and $n = 2$). This example runs through this chapter to illustrate the different concepts. The natural description of any symmetric tensor is by its coefficients $\mathcal{T}_{i_1 i_2 i_3 i_4}$ with $0 \leq i_1, i_2, i_3, i_4 \leq 2$ and the latter coefficients are unchanged by any permutation of $\{i_1, i_2, i_3, i_4\}$. We can equivalently describe the same tensor with the indices α_1, α_2 counting the number of occurrences of 1 and 2 respectively. For example, with $\alpha_1 = 1, \alpha_2 = 1$ we have

$$\mathfrak{T}_{11} \longleftrightarrow \mathcal{T}_{0012} = \mathcal{T}_{0021} = \mathcal{T}_{0102} = \mathcal{T}_{0120} = \mathcal{T}_{1002} = \mathcal{T}_{1020} = \mathcal{T}_{1200} = \mathcal{T}_{2001} = \mathcal{T}_{2010} = \mathcal{T}_{2001} \,.$$

Note that the number of times the index 0 appears is equal to $d - (\alpha_1 + \alpha_2) = 2$, as already mentioned.

§ 7.3 CPD and moment problem

This sections gives an interpretation of Decomposition (7.1) as an integral with respect to a measure supported on R points.

7.3.1 CPD as a measure integration

Following the re-indexing in 7.2.3, and with the dehomogenization performed in Section 7.2.2, the rank R decomposition in (7.1) also reads

$$\begin{aligned} \mathfrak{T}_{\alpha_1,\dots,\alpha_n} &= \sum_{r=1}^{R} v_0(r)^{d-|\boldsymbol{\alpha}|} v_1(r)^{\alpha_1} \dots v_n(r)^{\alpha_n} \\ &= \sum_{r=1}^{R} \lambda_r u_1(r)^{\alpha_1} \dots u_n(r)^{\alpha_n} = \sum_{r=1}^{R} \lambda_r \mathbf{u}(r)^{\boldsymbol{\alpha}} \,. \end{aligned} \tag{7.3}$$

Now, we write (7.3) in an equivalent integral form

$$\mathfrak{T}_{\boldsymbol{\alpha}} = \mathfrak{T}_{\alpha_1,\dots,\alpha_n} = \int \mathbf{x}^{\boldsymbol{\alpha}} \mu(\mathrm{d}\mathbf{x}) \,, \tag{7.4}$$

where μ is the discrete positive measure

$$\mu = \sum_{r=1}^{R} \lambda_r \delta_{\mathbf{u}(r)} \,, \tag{7.5}$$

defined on n variables and supported on the points $(\mathbf{u}(r))_{r\in[\![1,R]\!]}$. Such a measure, which is concentrated on a finite set of R points, is referred to as an R-atomic measure. Finding the vectors $(\mathbf{u}(r))_{r\in[\![1,R]\!]}$ and the coefficients $(\lambda_r)_{r\in[\![1,R]\!]}$ of the CPD is hence equivalent to determining the R-atomic measure μ in (7.5).

Notice that the right hand side of (7.4) is the moment of order $\boldsymbol{\alpha}$ of the measure μ. Therefore, (7.4) shows that finding a CPD of a symmetric tensor $\mathcal{T}$ as in (7.1) is equivalent to the estimation of a discrete measure μ from its moments of degree up to d. This is a truncated moment problem similar to the one encountered in Appendix D during the extraction step where the minimizers of the rational problem are extracted from the solution to SDP relaxations. We can hence leverage the tools from truncated moment problems [KN77, CF96], and especially the extraction method, to find a CPD of the tensor $\mathcal{T}$.

Example In the example provided in Section 7.2.3, the elements of $\mathcal{T}$ are moments up to degree 4 of a 2-atomic measure μ on $\mathbf{x} = (x_1, x_2)$. Table 7.1 represents some of the monomials corresponding to those moment along with their corresponding tensor elements expressed with both kinds of indexing.

Table 7.1: *Tensors elements and related moments*

Tensor elements		Moment monomials $\mathbf{x}^{\boldsymbol{\alpha}}$
$\mathcal{T}_{0000}$	$\mathfrak{T}_{00}$	1
$\mathcal{T}_{0001}$	$\mathfrak{T}_{10}$	x_1
$\mathcal{T}_{0002}$	$\mathfrak{T}_{01}$	x_2
$\mathcal{T}_{0011}$	$\mathfrak{T}_{20}$	x_1^2
$\mathcal{T}_{0012}$	$\mathfrak{T}_{11}$	x_1x_2
$\mathcal{T}_{0022}$	$\mathfrak{T}_{02}$	x_2^2

7.3.2 Moment matrix

An important tool for solving the truncated moment problem (7.4) is the moment matrix $\mathbf{M}_k$ of order $k = \frac{d}{2}$ defined in Chapter 3. Here, this matrix is built from the tensor $\mathcal{T}$ such that

$$\left(\forall(\boldsymbol{\alpha},\boldsymbol{\beta}) \in (\mathbb{N}^n_k)^2\right) \quad (\mathbf{M}_k)_{(\boldsymbol{\alpha},\boldsymbol{\beta})} = \mathfrak{T}_{\boldsymbol{\alpha}+\boldsymbol{\beta}}\,,$$

where the multi-indices $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are arranged with respect to the graded lexicographic order. Hence, the matrix $\mathbf{M}_k$ contains all the moments up to degree $d = 2k$ of the measure μ. Its number of rows and columns are both $N = \binom{n+k}{k}$, i.e. the number of moments up to degree k.

Furthermore, for any integer l such that $k - l \geq 1$, we define the moment matrix $\mathbf{M}_{k-l}$ of order $k-l$ as the leading principal submatrix of $\mathbf{M}_k$ of size $\binom{n+k-l}{k-l}$, i.e. the matrix composed of the $\binom{n+k-l}{k-l}$ first rows and columns of $\mathbf{M}_k$.

We notice that the moment matrix only contains N^2 terms which is smaller than the n^d terms of the original tensor or of its matricization. This is an important point since moment matrices are the primal tool of our moment framework to solve CPD. Table 7.2 compares the dimensions for some values of n and d. For $n = 1$ and $n = 2$, the moment matrices contain more elements than the full tensors but those cases are not challenging in terms of computation and memory. However, the dimension of moment

matrix becomes much smaller for higher n at any order d, which is a significant gain in terms of memory storage requirements when compared to rank estimation methods that require the full tensor or its matricization [SLF+17].

Table 7.2: *Comparison of the number of element in* $\mathcal{T}$ *and* $\mathbf{M}_k$

		n+1						
		2	3	4	5	6	8	10
$d=4$	n^d	16	81	256	625	1296	4096	10000
	N^2	36	100	225	441	784	2025	4356
$d=6$	n^d	64	729	4096	15625	46656	262144	1000000
	N^2	100	400	1225	3136	7056	27225	81796

Example We proceed with our example from Section 7.2.3. We build the following moment matrices from the tensor $\mathcal{T}$ with respect to the lexicographic ordering:

$$\mathbf{M}_2 = \begin{bmatrix} & \mathbf{M}_1 & & \mathfrak{T}_{20} & \mathfrak{T}_{11} & \mathfrak{T}_{02} \\ & & & \mathfrak{T}_{30} & \mathfrak{T}_{21} & \mathfrak{T}_{12} \\ & & & \mathfrak{T}_{31} & \mathfrak{T}_{22} & \mathfrak{T}_{03} \\ \mathfrak{T}_{20} & \mathfrak{T}_{30} & \mathfrak{T}_{31} & \mathfrak{T}_{40} & \mathfrak{T}_{31} & \mathfrak{T}_{22} \\ \mathfrak{T}_{11} & \mathfrak{T}_{21} & \mathfrak{T}_{22} & \mathfrak{T}_{31} & \mathfrak{T}_{22} & \mathfrak{T}_{13} \\ \mathfrak{T}_{02} & \mathfrak{T}_{12} & \mathfrak{T}_{03} & \mathfrak{T}_{22} & \mathfrak{T}_{13} & \mathfrak{T}_{04} \end{bmatrix}$$

with

$$\mathbf{M}_1 = \begin{bmatrix} \mathfrak{T}_{00} & \mathfrak{T}_{10} & \mathfrak{T}_{01} \\ \mathfrak{T}_{10} & \mathfrak{T}_{20} & \mathfrak{T}_{11} \\ \mathfrak{T}_{01} & \mathfrak{T}_{11} & \mathfrak{T}_{02} \end{bmatrix}.$$

Remark: We assumed that d is an even integer. However, this is not a restriction to our method. Indeed, odd order tensors can be handled identically by setting $k = \frac{d-1}{2}$. The moment matrix hence contents all the moments up to degree $d-1$. Note that some elements of the tensors are then not used.

Kernel of the moment matrix

As we will see in Section 7.5, a key element to estimate μ from (7.4) is the kernel of the moment matrix $\mathbf{M}_k$ which is defined as

$$\operatorname{Ker} \mathbf{M}_k = \{\mathbf{p} \in \mathbb{R}^N \mid \mathbf{M}_k \mathbf{p} = 0\}.$$

The moment matrix $\mathbf{M}_k$ is indexed by the pair of multi-indices $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ in $\mathbb{N}^n_k \times \mathbb{N}^n_k$. Since each multi-index $\boldsymbol{\alpha}$ corresponds to a monic monomial $\mathbf{x}^{\boldsymbol{\alpha}}$, we can associate to each vector $\mathbf{p}$ in the kernel of the moment matrix $\mathbf{M}_k$, a polynomial p such that

$$(\forall \mathbf{x} \in \mathbb{R}^n) \quad p(\mathbf{x}) = \sum_{\boldsymbol{\alpha} \in \mathbb{N}^n_k} p_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}}.$$

7.3.3 Solving the truncated moment problem

To solve the moment problem corresponding to the CPD of $\mathcal{T}$, we rely on the following important result on the moment matrices from [Lau08, Theorem 5.29].

Theorem 6
If $\mathbf{M}_k$ is positive semi-definite and if the rank of both $\mathbf{M}_k$ and $\mathbf{M}_{k-1}$ are equal to R, then the moments they contain represent a unique R-atomic measure whose support are the zeros of the polynomials associated to the kernel of $\mathbf{M}_k$.

Based on Theorem 6, Section 7.4 links the tensor rank to the ranks of its associated moment matrices and offers a necessary and sufficient condition in contrast to other methods such as matrix unfoldings. On the other hand, Section 7.5 details how to use this theorem to extract the generating vectors of the CPD by exploiting the kernel of the moment matrix $\mathbf{M}_k$.

Note that, in our context, the necessary positive semi-definiteness condition for a matrix to be a moment matrix is always verified. Indeed, since $\mathcal{T}$ admits a CPD, there exists an unknown integer R and vectors $(\mathbf{v}(r))_{r\in[\![1,R]\!]}$ satisfying (7.1). A corresponding positive measure μ can thus always be defined as in (7.5). Then, we have

$$\begin{aligned}(\forall \mathbf{a}\in\mathbb{R}^N)\quad \mathbf{a}^\top\mathbf{M}_k\mathbf{a} &= \sum_{|\boldsymbol{\alpha}|\leq k,|\boldsymbol{\beta}|\leq k} a_{\boldsymbol{\alpha}}a_{\boldsymbol{\beta}}\mathfrak{T}_{\boldsymbol{\alpha}+\boldsymbol{\beta}}\\ &= \sum_{|\boldsymbol{\alpha}|\leq k,|\boldsymbol{\beta}|\leq k} a_{\boldsymbol{\alpha}}a_{\boldsymbol{\beta}}\int \mathbf{x}^{\boldsymbol{\alpha}+\boldsymbol{\beta}}\mu(\mathrm{d}\mathbf{x})\\ &= \int p_{\mathbf{a}}(\mathbf{x})^2\mu(\mathrm{d}\mathbf{x})\geq 0\end{aligned}$$

where $p_{\mathbf{a}}$ is the polynomial $p_{\mathbf{a}}=\sum_{|\boldsymbol{\alpha}|\leq k} a_{\boldsymbol{\alpha}}\mathbf{x}^{\boldsymbol{\alpha}}$. It follows that the moment matrix $\mathbf{M}_k$ that we defined for a tensor is always positive semi-definite.

§ 7.4 Tensor rank detection

Using the tools of Section 7.3, we provide here two results concerning the rank detection of a symmetric tensor before illustrating them on numerical examples.

Since the points supporting the measure μ in (7.5) are the generating vectors of the CPD, if the rank condition in Theorem 6 is verified, we obtain that R is also the rank of the tensor $\mathcal{T}$. This gives the following result that provides a sufficient condition on the moment matrices $\mathbf{M}_k$ and $\mathbf{M}_{k-1}$ for certifying the rank of the corresponding symmetric tensor $\mathcal{T}$:

Corollary 1
The tensor $\mathcal{T}$ has rank R if its moment matrices of order k and $k-1$ both have rank equal to R:

$$(\operatorname{rank}\mathbf{M}_k=\operatorname{rank}\mathbf{M}_{k-1}=R)\quad\Longrightarrow\quad(\operatorname{rank}_S\mathcal{T}=R)\ .$$

Corollary 1 offers a conceptually simple tool to get the rank of a tensor. Indeed, we can first build the associated moment matrix $\mathbf{M}_k$ from $\mathcal{T}$, then extract the principal submatrix $\mathbf{M}_{k-1}$ and finally check whether their ranks are equal. The conditions for which this corollary is applicable are discussed in more details below and an extension is given.

7.4.1 Extended detection result

Since the order k of the moment matrix needs to be non-negative to make sense, the matrix than $\mathbf{M}_{k-1}$ is defined for $k\geq 2$ only. This means that our detection method can only be applied to symmetric tensors of order at least $d=4$.

Furthermore, Corollary 1 cannot be used to detect ranks exceeding the size of the moment matrix $\mathbf{M}_{k-1}$. It can thus only be used to detect rank values smaller than $\binom{n+k-1}{k-1}$. The tensor rank can be greater since, to our knowledge, the lowest rank upper-bound is $\binom{n+d}{d}$ (see [CGLM08]). The rank value restriction of our method is hence $R \leq \binom{n+k-1}{k-1}$. Note that for a tensor of order 4, it reduces to $R \leq n+1$. In large tensor data, and in many applications where a low rank representation of the tensor is looked for, our result may however provide interesting guarantees.

When the tensor rank is known in advance to have smaller rank than the size of $\mathbf{M}_{k-1}$, the following reciprocal of Corollary 1 holds

Corollary 2
Under Assumption 1, if $R \leq \binom{n+k-1}{k-1}$, *we have the following equivalence*

$$(\operatorname{rank}_S \mathcal{T} = R) \quad \Longleftrightarrow \quad (\operatorname{rank} \mathbf{M}_k = \operatorname{rank} \mathbf{M}_{k-1} = R) \ .$$

The proof of the equivalence in Corollary 2 results directly from a theorem proved by Curto and Fialkow [CF96, Theorem 7.10]. Nevertheless Assumption 1 must hold. Indeed, dehomogenization is always possible when $\mathbf{M}_k$ and $\mathbf{M}_{k-1}$ have same rank. Conversely, if $\mathcal{T}$ has rank R but Assumption 1 does not hold, then $\operatorname{rank} \mathbf{M}_{k-1}$ may be different from $\operatorname{rank} \mathbf{M}_k$ as shown by the following counterexample.

Counterexample Let $\mathcal{T}$ be a tensor of order 4 on R^2 with rank 2. Hence, $\mathcal{T}$ reads:

$$\mathcal{T} = \mathbf{v}(1)^{\otimes 4} + \mathbf{v}(2)^{\otimes 4} \ .$$

We assume that both $v_0(1)$ and $v_1(2)$ are equal to zero. Therefore, we have:

$$\begin{aligned}
T_{0000} = \mathfrak{T}_0 &= v_0(1)^4 + v_0(2)^4 &&= v_0(2)^4 \\
T_{0001} = \mathfrak{T}_1 &= v_0(1)^3 v_1(1) + v_0(2)^3 v_1(2) &&= 0 \\
T_{0011} = \mathfrak{T}_2 &= v_0(1)^2 v_1(1)^2 + v_0(2)^2 v_1(2)^2 &&= 0 \\
T_{0111} = \mathfrak{T}_3 &= v_0(1) v_1(1)^3 + v_0(2) v_1(2)^3 &&= 0 \\
T_{1111} = \mathfrak{T}_4 &= v_1(1)^4 + v_1(2)^4 &&= v_1(1)^4
\end{aligned}$$

The moment matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ are thus given by:

$$\mathbf{M}_1 = \begin{bmatrix} v_0(2)^4 & 0 \\ 0 & 0 \end{bmatrix} , \quad \mathbf{M}_2 = \begin{bmatrix} v_0(2)^4 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & v_1(1)^4 \end{bmatrix} .$$

We notice that $\operatorname{rank} \mathbf{M}_1$ is equal to 1 while $\operatorname{rank} \mathbf{M}_2$ is equal to 2.

7.4.2 Numerical results

For numerical illustration, we generate randomly rank-R tensors by drawing their CPD vectors $(\mathbf{v}(r))_{r \in [\![1,R]\!]}$ according to a uniform distribution on $[0,1]^{n+1}$.

7.4.2.1 Importance of rank detection

Most of the standard algorithms to perform CPD require as input, the rank of the sought CPD.

To show the importance of rank detection, we compare the relative error between a tensor of known rank and the CPD returned by algorithms ALS, NLS, OPT, and GEVD for various input ranks. For the latter algorithms, we use the implementation

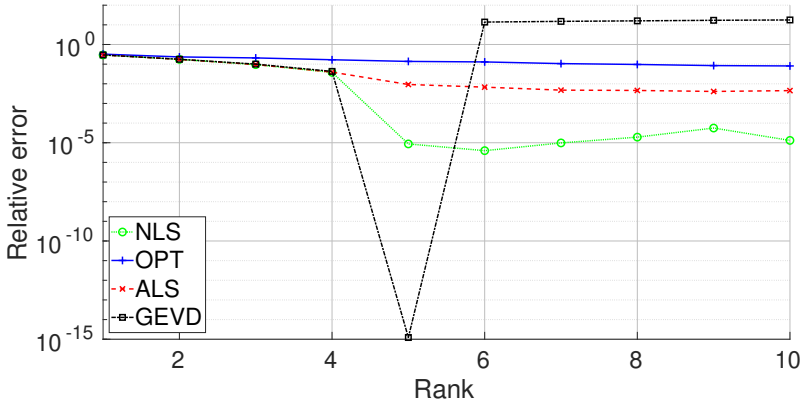


Figure 7.1: *Relative error of the CPD*

from Tensorlab 3.0 [VDS+16]. The relative error between tensors $\mathcal{T}$ and $\hat{\mathcal{T}}$ is used to assess the quality of the CPD and it is defined as

$$\text{relative error} = \frac{\left\|\mathcal{T} - \hat{\mathcal{T}}\right\|_F}{\|\mathcal{T}\|_F},$$

where $\|.\|_F$ is the Frobenius norm.

Figure 7.1 shows the average relative error on 100 random tensors of rank 5, order 4 and dimension 30 for the different input ranks. We can notice the sensitivity of the algorithms to the input rank. More specifically, the algebraic method GEVD shows a high improvement potential when the true rank is known. Similarly NLS shows a particularly good performance for the real rank value. This shows the importance of a correct rank detection and thus of our method, which can be used to find directly the rank of a given tensor before feeding it into a CPD algorithm.

7.4.2.2 Finding the rank of a symmetric tensor

We now look at the numerical rank of moment matrices to verify that Corollary 1 applies well in practice. We compute the rank of the tensor through the ranks of moment matrices $\mathbf{M}_k$ and $\mathbf{M}_{k-1}$ and if both are equal, then this delivers the rank of the tensor.

On four different examples with different tensor ranks, Figure 7.2 plots the first twelve singulars values of the moment matrices, normalized by the largest singular value σ_1. We observe a significant drop after the same singular value for both moment matrices. We can conclude that they have the same numerical rank and infer that this rank is also the rank of the tensor.

We then generate 100 random rank-7 tensors of dimension 20 and of order 6. Figure 7.3 shows the ratio of the successive singular values (sorted in decreasing order) σ_5/σ_6, σ_6/σ_7 and, σ_7/σ_8 of $\mathbf{M}_2$, without noise and in a presence of an additive zero-mean Gaussian noise of variance 10^{-4}. In both case, we observe a gap in the ratios that indicate a rank of 7 for the moment matrix. We observe a similar gap for $\mathbf{M}_3$ which shows that our method detects the rank of the tensor correctly.

In conclusion, for noiseless or moderate level of noise scenarios, we observe that $\mathbf{M}_k$ and $\mathbf{M}_{k-1}$ always have same rank under conditions of Section 7.4.1. We therefore can determine the rank of the corresponding tensor and confirm that Corollary 1 leads to satisfactory numerical results.

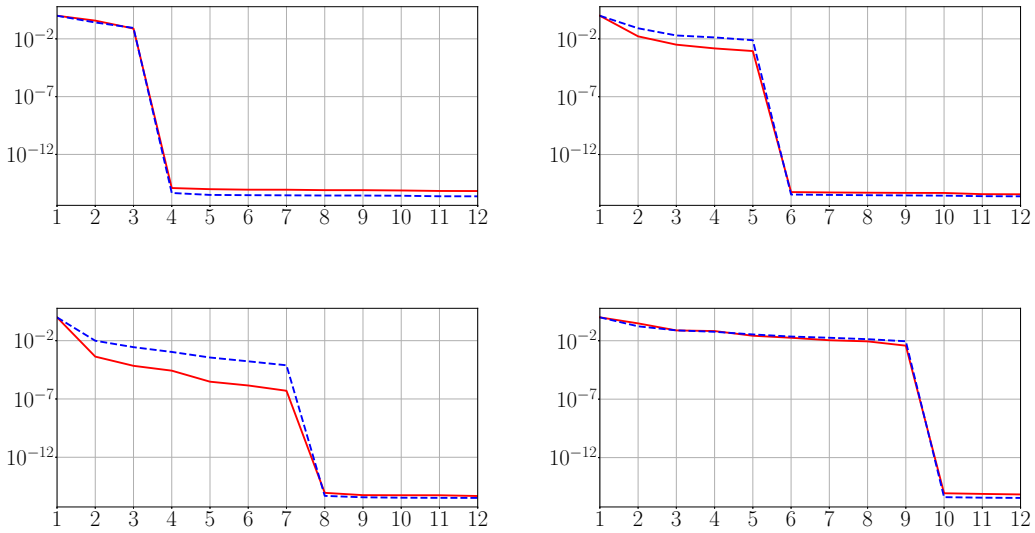
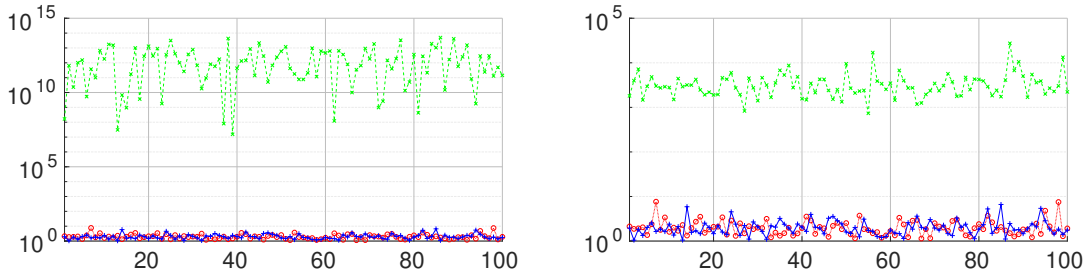


Figure 7.2: *First singular values of* $\mathbf{M}_3$ *(red) and* $\mathbf{M}_2$ *(blue)*
(Top-left: $R = 3$*, Top-right:* $R = 5$*, Bottom-left:* $R = 7$*, Bottom-right:* $R = 9$*)*



(a) *No noise* **(b)** *Noisy*

Figure 7.3: *Singular value ratios gap of* $\mathbf{M}_2$ *for 100 tests*
(In red: σ_5/σ_6*, in blue:* σ_6/σ_7*, and in green:* σ_7/σ_8*)*

7.4.2.3 Application to cumulant-based source separation

We consider the cumulant tensor of a random vector $\mathbf{y}$ such that

$$\mathbf{y} = \mathbf{A}\mathbf{s} + \mathbf{w}$$

where $\mathbf{A} \in \mathbb{R}^{n \times R}$ is an unknown matrix, $\mathbf{w}$ is a white Gaussian noise with zero-mean, and $\mathbf{s}$ is a vector of R independent random variables. In our experiments, we draw the elements of $\mathbf{A}$ according to a uniform distribution on $[0, 1]$ and the elements of $\mathbf{s}$ take value -1 or 1 with equal probability.

Our goal is to retrieve the number of sources R from several samples of the observation vector $\mathbf{y}$. It is known [Com10] that the tensor of cumulant of order 4 follows a CPD

$$\mathrm{Cum}^4(y_i, y_j, y_k, y_l) = \sum_{r=1}^{R} A_{ir} A_{jr} A_{kr} A_{lr} \, \mathrm{Cum}^4(s_r) \, .$$

We use the empirical estimation of moments to compute the estimated cumulant tensor, which is a noise-corrupted version of a low-rank tensor. We then apply our method to detect the low rank model and thereby recover the number of sources. Table 7.3

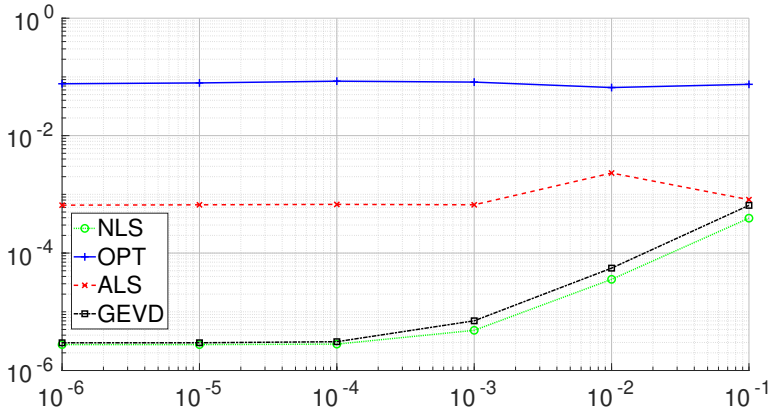


Figure 7.4: *Average relative error depending on noise variance ($R = 3$)*

shows the percentage of cases where the number of sources is correctly detected over 200 runs. For each of them, 100,000 samples of vectors $\mathbf{y}$ of size 20 are generated, for various variance values of the noise $\mathbf{w}$. We detect the rank numerically in the moment matrices by a gap of 10^3 in the ratio of successive singular values. We then feed the detected ranks into NLS, ALS, OPT and GEVD algorithms and look at the CPD they returned. Figure 7.4 plots the average relative errors for the four algorithms in the case of three sources.

Table 7.3: *Percentage of successful detection of the number of sources*

	Number of sources R		
Variance of the noise	3	4	5
0	100%	97%	69%
$1 \cdot 10^{-6}$	100%	97%	69%
$1 \cdot 10^{-4}$	100%	97%	69%
$1 \cdot 10^{-2}$	100%	97%	69%
$1 \cdot 10^{-1}$	100%	95%	57%

We note that even with a reasonable level of noise, the moment matrices have still the same rank that corresponds to the rank of the tensor. The method shows satisfactory results for low rank tensor corrupted with noise. Nevertheless, we observe that the higher the rank, the higher the sensitivity to the estimation noise and the higher the number of samples must be. Figure 7.4 also shows that algebraic methods such as GEVD are more sensitive to the noise despite their good performance in the noiseless case.

§ 7.5 Extracting the CPD vectors

This section deals with the recovery of the support of the measure μ, or equivalently the vectors in the CPD, from $\mathbf{M}_k$. Results from Section 7.3.3 guarantee the existence of the measure μ satisfying (7.4). Moreover, we assume that we know the rank R of the tensor $\mathcal{T}$ and thus the rank of $\mathbf{M}_k$. It can be detected using a rank detection method such as the one from Section 7.4.

7.5.1 Eigenvalue method and CPD generating vectors

According to Theorem 6, the vectors $(\mathbf{u}(r))_{r\in[\![1,R]\!]}$ forming the support of the sought measure μ are the common zeros of the polynomials with coefficients in $\operatorname{Ker}\mathbf{M}_k$, that is

$$(\mathbf{u}(r))_{r\in[\![1,R]\!]} = \{\mathbf{x}\in\mathbb{R}^n \mid (\forall \mathbf{p}\in \operatorname{Ker}\mathbf{M}_k) \quad p(\mathbf{x})=0\}\,.$$

Finding the generating vectors of the CPD is therefore equivalent to solving a multivariate polynomial system.

The eigenvalue method transforms the original polynomial system into a linear algebra problem. Despite being described in a few places [HL05], this method seems to be widely ignored by the signal processing community. We briefly describe here the different steps in our context. A more general and theoretical explanation of the method can be found in [CLO05].

The first step consists in computing through Gaussian elimination the reduced row echelon form of $\mathbf{M}_k$, which is an upper triangular $N\times N$ matrix whose last $N-R$ rows are composed solely of zeros. Dropping the last $N-R$ rows of zeros, we note $\mathbf{U}$ the obtained $R\times N$ matrix and $(\mathbf{u}_{\boldsymbol{\alpha}})_{\boldsymbol{\alpha}\in\mathbb{N}^n_k}$ its N column vectors. We then read from $\mathbf{U}$ the column multi-indices of the pivot elements. We get R pivots whose indices are denoted by $(\boldsymbol{\beta}_r)_{r\in[\![1,R]\!]}$. We have then the following result:

Theorem 7
For every i in $[\![1,n]\!]$, the i-th coordinates of the R points $(\mathbf{u}(r))_{r\in[\![1,R]\!]}$ are the R eigenvalues of the matrices $\mathbf{N}_i$ extracted from $\mathbf{U}$ such that

$$\mathbf{N}_i = [\mathbf{u}_{\boldsymbol{\beta}_1+\mathbf{e}_i}\ldots\mathbf{u}_{\boldsymbol{\beta}_R+\mathbf{e}_i}]\,.$$

where $\mathbf{e}_i$ is a multi-index of $\mathbb{N}^n$ whose all elements are equal to zero except its i-th element which is 1.

This result is a direct application of Stickelberger eigenvalue theorem [CLO05, Theorem 4.5]. The matrices $(\mathbf{N}_i)_{i\in[\![1,n]\!]}$ are called the multiplication matrices. The origin of this name can be found in [CLO05].

Example The moment matrix $\mathbf{M}_2$ corresponding to the example from Section 7.2.3 is a 6×6 matrix that can be seen indexed by the following ordered set

$$\mathbb{N}^2_2 = \{(0,0),(1,0),(0,1),(2,0),(1,1),(0,2)\}\,.$$

We then obtain its reduced row echelon form, e.g.

$$\mathbf{U} = \begin{bmatrix} 1 & 0 & 0 & u_1 & u_4 & u_7 \\ 0 & 1 & 0 & u_2 & u_5 & u_8 \\ 0 & 0 & 1 & u_3 & u_6 & u_9 \end{bmatrix}\,.$$

We read out the indices of the pivots

$$\boldsymbol{\beta}_1 = (0,0)\,,\quad \boldsymbol{\beta}_2 = (1,0)\,,\quad \boldsymbol{\beta}_3 = (0,1)\,,$$

whence the corresponding multiplication matrices

$$\mathbf{N}_1 = \begin{bmatrix} 0 & u_1 & u_4 \\ 1 & u_2 & u_5 \\ 0 & u_3 & u_6 \end{bmatrix}\,,\quad \mathbf{N}_2 = \begin{bmatrix} 0 & u_4 & u_7 \\ 0 & u_5 & u_8 \\ 1 & u_6 & u_9 \end{bmatrix}\,.$$

Note, for instance, that the second column of $\mathbf{N}_1$ is $\mathbf{u}_{(2,0)}$.

7.5.2 Computation of the eigenvalues of $(\mathbf{N}_i)_{i\in[\![1,n]\!]}$

Since multiplication matrices all commute pairwise, they preserve the eigenspaces of each others. A numerically stable way to compute their eigenvalues based on Schur factorization [CGT97] is then available and summarized below.

First, build a random linear combination $\mathbf{N}_h$ of the matrices $(\mathbf{N}_i)_{i\in[\![1,n]\!]}$

$$\mathbf{N}_h = \sum_{i=1}^{n} a_i \mathbf{N}_i \,, \tag{7.6}$$

where $(a_i)_{i\in[\![1,n]\!]}$ are randomly chosen real numbers summing up to one. Now, a key point is that the left eigenspaces of $\mathbf{N}_h$ need all to be one-dimensional in order to avoid to miss any points in the support of μ. Since the rank of the matrix $\mathbf{M}_k$ is R, this holds almost surely [CLO05] for any random choice of the $(a_i)_{i\in[\![1,n]\!]}$ with absolutely continuous probability density function. Following [CGT97], the left eigenspaces of $\mathbf{N}_h$ are then found by computing an ordered Schur decomposition $\mathbf{QTQ}^\top$ of $\mathbf{N}_h$ where $\mathbf{Q}$ is an orthogonal matrix and $\mathbf{T}$ is upper triangular. The coordinates of the points $(\mathbf{u}(r))_{r\in[\![1,R]\!]}$ are thus given by

$$(\forall i \in [\![1,n]\!])(\forall r \in [\![1,R]\!]) \quad u_i(r) = \mathbf{q}_r^\top \mathbf{N}_i \mathbf{q}_r \,,$$

where $\mathbf{q}_r$ is the r-th column of matrix $\mathbf{Q}$. The sketch of the extraction method is provided in Algorithm 19. Finally, the weighting coefficients $(\lambda_r)_{r\in[\![1,R]\!]}$ of (7.2) can be retrieved as explained below.

Algorithm 19: Extraction of CPD vectors

Inputs : Moment matrix $\mathbf{M}_k$ and rank R of $\mathcal{T}$
Output: Vectors $(\mathbf{u}(r))_{r\in[\![1,R]\!]}$ generating $\mathcal{T}$

1 Compute the reduced row echelon form of $\mathbf{M}_k$ and extract $\mathbf{U}$;
2 Find the column indices of the pivots in $\mathbf{U}$;
3 Read multiplication matrices $(\mathbf{N}_i)_{i\in[\![1,n]\!]}$ from $\mathbf{U}$;
4 Find the common eigenvalues of the multiplication matrices:
5 Compute a random combination $\mathbf{N}_h$ of the multiplication matrices as in (7.6) ;
6 Compute the ordered Schur decomposition $\mathbf{QTQ}^\top$ of $\mathbf{N}_h$;
7 Read the i-th coordinate of points $(\mathbf{u}(r))_{r\in[\![1,R]\!]}$ by computing $\mathbf{Q}^\top\mathbf{N}_i\mathbf{Q}$;

7.5.3 Retrieving the coefficients $(\lambda_r)_{r\in[\![1,R]\!]}$

In order to retrieve the coefficients $(\lambda_r)_{r\in[\![1,R]\!]}$, we solve a linear system. We select randomly S samples from both the original tensor $\mathcal{T}$ and the R rank-1 tensors generated from each vector $\mathbf{u}(r)$. We stack the samples of $\mathcal{T}$ into a vector $\mathbf{b}$ of $\mathbb{R}^S$ and the samples from the R rank-1 tensors into row vectors of a matrix $\mathbf{A}$ of $\mathbb{R}^{S\times R}$. We then find the least squares solution to the overdetermined linear system $\mathbf{A}\boldsymbol{\lambda} = \mathbf{b}$ using a least squares, to retrieve the vector $\boldsymbol{\lambda}$ that contains the values of $(\lambda_r)_{r\in[\![1,R]\!]}$. Sampling the tensors is a cheaper alternative than flattening the whole tensors, especially when the tensors have high orders or dimensions. We typically choose a number of samples S equal to ten times the order of the tensors $10d$.

7.5.4 Decomposition of high order tensors

For high order tensor, it may not be necessary to build the full moment matrix $\mathbf{M}_k$ to retrieve a CPD. In fact, the rank condition in Corollary 1 can be verified for moment matrices $\mathbf{M}_l$ and $\mathbf{M}_{l-1}$ with l much smaller than $k = \frac{d}{2}$. The additional moments contained in $\mathbf{M}_k$ but not in $\mathbf{M}_l$ thus do not bring more information on the measure μ and hence on the CPD. Since the dimension N of the moment matrix is increasing exponentially with the order k, it is computationally interesting to look whether the rank condition in Corollary 1 is full-filled by the principal submatrices This is equivalent to building the moment matrix associated to a subtensor of $\mathcal{T}$. Indeed fixing some indices in $\mathbf{i}$, we can take a slice of $\mathcal{T}$ and get a smaller order symmetric tensor. We then apply our moment framework to this slice.

For instance, decomposing a tensor $\mathcal{T}$ of order $n = 6$ on $\mathbb{R}^{30}$ of rank $R = 4$ using our method takes around 560 seconds. On the other hand, if we take a slice of $\mathcal{T}$ of order 4 by fixing the indices i_5 and i_6 and build the corresponding moment matrix, Algorithm 19 takes only 3 seconds. In both cases, we retrieve the correct generating vectors with an accuracy higher than 10^{-8}.

7.5.5 Numerical experiments

7.5.5.1 Performance of the proposed method

We generate each symmetric rank-R tensor $\mathcal{T}$ randomly by drawing the coefficients of its vectors $(\mathbf{v}(r))_{r\in[\![1,R]\!]}$ from a uniform distribution on $[-1, 1]$. We then apply our method to retrieve the CPD of $\mathcal{T}$ and denote by $\hat{\mathcal{T}}$ the tensor reconstructed from the computed CPD. We assume that the rank R is known (using any existing rank detection method) and we focus only on the retrieval of the generating vectors in the CPD.

For each test case, we run 100 simulations and show only the average results. To assess the quality of the reconstruction, we use the relative error between tensors $\mathcal{T}$ and $\hat{\mathcal{T}}$. Moreover, we also compute a score inspired by [BK+17] to evaluate the reconstruction quality. The score measures the similarity between the original generating vectors $(\mathbf{v}(r))_{r\in[\![1,R]\!]}$ of a symmetric rank-R tensor and the vectors $(\hat{\mathbf{v}}(r))_{r\in[\![1,R]\!]}$ obtained after computing its CPD. It is computed as the product of the correlation between the vectors $(\mathbf{v}(r))_{r\in[\![1,R]\!]}$ and $(\hat{\mathbf{v}}(r))_{r\in[\![1,R]\!]}$, namely

$$\text{score} = \prod_{r=1}^{R} \frac{\langle \mathbf{v}(r) \mid \hat{\mathbf{v}}(r) \rangle}{\|\mathbf{v}(r)\| \, . \, \|\hat{\mathbf{v}}(r)\|} \, .$$

Thereby, when the score is close to 1, the vectors $(\hat{\mathbf{v}}(r))_{r\in[\![1,R]\!]}$ are strongly correlated to $(\mathbf{v}(r))_{r\in[\![1,R]\!]}$ and the CPD is accurate. However, if the score is close to 0, the CPD yields a poor quality decomposition.

Table 7.4 shows that our method can accurately reconstruct the CPD of a symmetric tensor for various combinations of dimension, order and rank. The running time is still fair and scales well with the order or the rank of the tensor.

Figure 7.5 shows several cases where the data in the tensor are corrupted with an additive i.i.d. zero-mean Gaussian noise. We first perform a truncated SVD of $\mathbf{M}_k$ at rank R before applying Algorithm 19. For a low noise level, the score is very high, close to 1; the CPD is exactly retrieved. Indeed, in the noiseless case, the decomposition returned by our method is guaranteed to be exact in contrast to some methods such as ALS. However, as the variance increases, the score reduces, the CPD is not accurate anymore and we lose any guarantee. Furthermore, for given dimension and order, we observe that the lower the rank, the lower the sensitivity to the noise.

Table 7.4: *Quality and reconstruction time of our method*

$n+1$	d	R	Relative error	Time (s)
10	4	10	4.10e-12	0.02
30	4	10	7.62e-12	7.11
50	4	10	4.51e-12	178.3
100	4	10	9.95e-12	13818
30	4	5	7.83e-13	3.97
30	4	30	1.20e-11	20.27
30	5	10	7.16e-13	7.27
30	6	10	9.06e-13	7.30

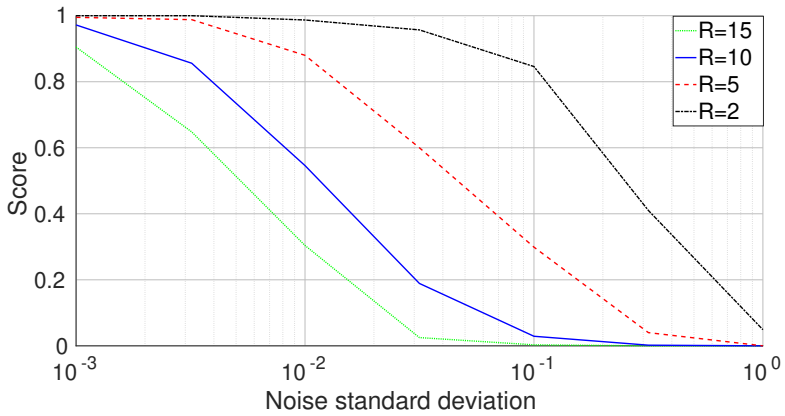


Figure 7.5: *Reconstruction score for noisy tensor* ($n = 29, d = 4$)

7.5.5.2 Comparison with other methods

We now compare our method to state-of-the-art CPD algorithms, especially the implementation of ALS, NLS, OPT and GEVD from Tensorlab 3.0 [VDS+16]. Table 7.5 shows a comparison of the relative error for the different algorithms and several different types of symmetric tensors. Results for GEVD have not been reported as they are similar to results for our method. Generally speaking, algebraic methods retrieve faithful CPD but, as shown in Figure 7.5 for our method, are sensitive to noise. On the other hand, Table 7.5 shows that methods based on optimization strategies are much less accurate than our method for exact decomposition.

§ 7.6 Robust extractions of global solutions in polynomial optimization with tensor CPD

In the previous sections of this chapter, we have interpreted the CPD of a symmetric tensor as a moment problem and we have used the tools from the moment problem to perform the CPD. However, we adopt in this section the reverse point of view and we propose to use the CPD literature to solve a specific moment problem: the extraction of the global minima in rational and polynomial optimization presented in Chapter 3. The standard extraction method, detailed in Appendix D, is similar to Algorithm 19 and thus,

Table 7.5: *Comparison with standard CPD methods*

Tensor features			Relative error			
$n+1$	d	R	ALS	OPT	NLS	Our method
10	4	10	$9 \cdot 10^{-3}$	$2 \cdot 10^{-2}$	$1 \cdot 10^{-3}$	$4 \cdot 10^{-10}$
30	4	10	$1 \cdot 10^{-2}$	$2 \cdot 10^{-2}$	$5 \cdot 10^{-4}$	$8 \cdot 10^{-12}$
50	4	10	$1 \cdot 10^{-2}$	$2 \cdot 10^{-2}$	$1 \cdot 10^{-4}$	$5 \cdot 10^{-12}$
100	4	10	$1 \cdot 10^{-2}$	$7 \cdot 10^{-3}$	$8 \cdot 10^{-5}$	$1 \cdot 10^{-13}$
30	4	5	$7 \cdot 10^{-3}$	$2 \cdot 10^{-1}$	$1 \cdot 10^{-3}$	$8 \cdot 10^{-13}$
30	4	20	$4 \cdot 10^{-3}$	$5 \cdot 10^{-2}$	$3 \cdot 10^{-4}$	$1 \cdot 10^{-11}$
30	4	30	$4 \cdot 10^{-3}$	$5 \cdot 10^{-2}$	$4 \cdot 10^{-4}$	$1 \cdot 10^{-11}$

shows the same advantages and weaknesses. Namely, although providing very accurate results in a noiseless context, the standard extraction method requires the knowledge of enough moments in order to recover the measure μ_* solution of the infinite-dimensional moment problem, i.e. a sufficiently high relaxation order, and it is highly sensitive to noise on the moments. The two latter drawbacks limit its practical use. Indeed, since the dimensions of SDP problems increase exponentially with the relaxation order, solving them with a sufficient order is often computationally too intensive for state-of-the-art SDP solvers and one has to settle for the solutions at the first orders of relaxations. As a consequence, we have access only to a limited number of moments which are moreover approximations to the true moments of μ_*. In this context, the extraction method in [HL05] either fails due to the lack of some moment values, or extract minimizers far from the global optima of the original rational problem due to the perturbation on the moments. Therefore, a robust extraction method is required for many practical applications of rational optimization in order to retrieve the exact global minimizers.

A first robust extraction has been proposed by Kleb et al. [KPV18] and is based on functional analysis and the Gelfand-Naimark-Segal construction. We propose here a different approach based on the connection from Section 7.3 between moment problem and tensor CPD. Namely, we suggest to use robust CPD methods such as NLS to solve the moment problem corresponding to the extraction of the solutions.

In this section we consider the generic rational problem (2.1):

$$\begin{aligned} \mathcal{J}^* = \underset{\mathbf{x}\in\mathbb{R}^T}{\text{minimize}} \quad & \frac{p(\mathbf{x})}{q(\mathbf{x})} \\ \text{subject to} \quad & \mathbf{x} \in \mathcal{K}\,, \end{aligned} \tag{2.1}$$

where p and q are polynomial functions on $\mathbb{R}^T$ and $\mathcal{K}$ is the basic semi-algebraic set defined in (3.2). Following the methodology of Chapter 3, we lift Problem (2.1) into the moment problem (3.4) and relax it into a hierarchy of SDP problems that are solved with a SDP solver. The optimal solution to the order k SDP problem is denoted $\mathbf{y}^*(k) = (y^*_{\boldsymbol{\alpha}}(k))_{\boldsymbol{\alpha}\in\mathbb{N}^T_{2k}}$ and it corresponds to all the moments up to degree $2k$.

7.6.1 Benefit of using interior point methods

Many state-of-the-art SDP solvers rely on interior point methods. Those methods have the advantage to return as a solution an interior point, i.e. a point $\mathbf{y}^*(k)$ in the relative interior of the optimum face of the feasible set. It can be proved [Lau08, Lemma 1.4] that the corresponding moment matrix returned by the SDP solver then has maximum rank

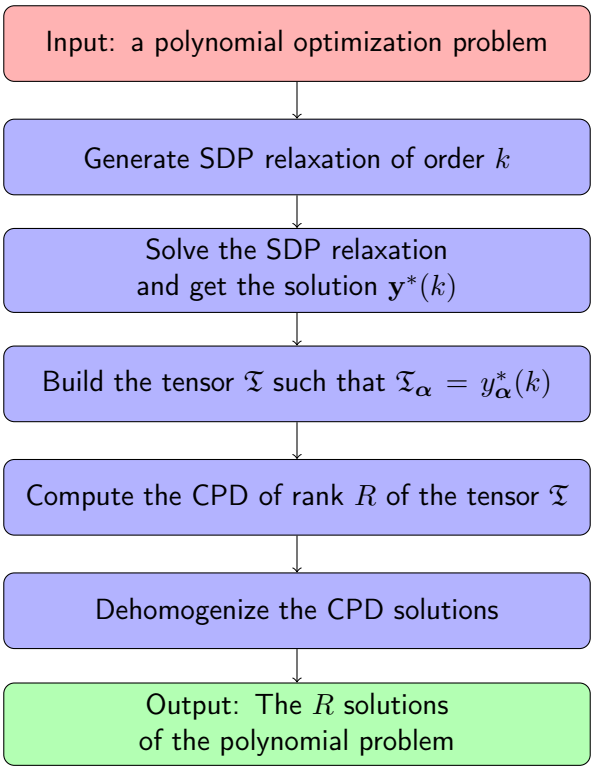


Figure 7.6: *Solving polynomial optimization problem with a robust extraction method*

among all the matrices which are solutions to the SDP problem. This is an important feature as it guarantees [Lau08] that all global minimizers of (2.1) can be recovered from $\mathbf{y}^*(k)$. Indeed, the rank of the moment matrix, and thus of the tensor $\mathcal{T}$ built from $\mathbf{y}^*(k)$, is equal to the number of global optima of (2.1). Having a maximal rank solution implies that no minimizers will be missed after performing the extraction. The choice of the SDP solver is therefore important for the extraction.

7.6.2 Robust extraction methods

We showed in Section 7.3 that the extraction problem is equivalent to finding the CPD of a symmetric tensor. Thereby robust tensor CPD methods can be used to perform the extraction instead of the algorithm in [HL05]. Especially, many algorithms relying on the minimization of a fit function instead of algebraic tools such as ALS or NLS have been developed to recover faithfully tensors corrupted by noise, even when some elements are missing [VDSL14].

Hence, after solving the SDP relaxations of order k, we build the symmetric tensor $\mathcal{T}$ out of the solutions $\mathbf{y}^*(k)$ of the SDP problem by setting $\mathfrak{T}_{\boldsymbol{\alpha}} = y^*_{\boldsymbol{\alpha}}(k)$ as shown in Figure 7.6. This tensor can be seen as a noisy subtensor of the infinite tensor of rank R containing all the moments of the sought measure μ_*. Therefore, we apply robust CPD algorithms to get a better approximation to the rank R infinite tensor and enhance the quality of the global optimum of the polynomial problem. Hence, using a moment vector from a low order of the hierarchy, we can avoid huge computational burden while obtaining accurate global minimizers of Problem (2.1).

An important remark is that many CPD algorithms require the prior knowledge of the rank R of the tensor decomposition which implies, in our context, to know the number of solutions to Problem (2.1).

7.6.3 Cases Study

7.6.3.1 A simple polynomial optimization problem

Let us take the following simple polynomial optimization problem from [HL05]

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^3}{\text{minimize}} \quad & -(x_1-1)^2-(x_2-1)^2-(x_3-1)^2 \\ \text{s.t.} \quad & 1-(x_1-1)^2 \geq 0 \\ & 1-(x_2-1)^2 \geq 0 \\ & 1-(x_3-1)^2 \geq 0\,. \end{aligned} \tag{7.7}$$

Problem (7.7) has eight global minima

$$\begin{aligned} x_1^* = (0,0,0) \quad x_2^* = (2,0,0) \quad x_3^* = (0,2,0) \quad x_4^* = (0,0,2) \\ x_5^* = (2,2,0) \quad x_6^* = (2,0,2) \quad x_7^* = (0,2,2) \quad x_8^* = (2,2,2)\,, \end{aligned}$$

and the optimal value is -3. Notice that it is not a convex problem. We use GloptiPoly [HLL09] to perform the relaxation into SDP problems and to extract solutions using the algebraic method [HL05]. Those SDP relaxations are solved by the SDP solver SDPT3 [TTT99]. We compare the extraction method from GloptiPoly with the implementation of NLS from Tensorlab [VDS+16]. We choose NLS as it gives better results than ALS and OPT.

Extraction method from GloptiPoly. We first apply directly Lasserre's framework in GloptiPoly to solve Problem (7.7). At the relaxation orders $k = 1$, $k = 2$, and $k = 3$, we do not have convergence in Lasserre's hierarchy and the extraction method fails. Indeed, at those orders, there are not enough moments in $\mathbf{y}^*(k)$ and thus the extraction procedure from [HL05] fails while extracting the multiplication matrices from the moment matrix. More precisely, in Theorem 7, the multi-index $\boldsymbol{\beta}_r+\mathbf{e}_i$ may be greater than the dimension of the moment matrix $\mathbf{M}_k$ and thus the multiplication matrix $\mathbf{N}_i$ cannot be extracted from it. The certificate of convergence is obtained for the relaxation order $k = 4$ and the algebraic extraction procedure hence retrieves the eight solutions and the optimal value with a precision higher than 10^{-4}.

Robust Extraction using NLS. On the other hand, instead of the standard algebraic method, we extract the solutions by applying the NLS algorithm on the tensor generated by the moment vector $\mathbf{y}^*(k)$ for $k = 1$, $k = 2$, and $k = 3$. At each order, we retrieve the eight approximate solutions listed in Table 7.6. Furthermore, Table 7.7 shows the value of the criterion at optimality for the solutions extracted with NLS and GloptiPoly, respectively. We observe that, by applying NLS on the tensor corresponding to $\mathbf{y}^*(3)$, we retrieve the eight global solutions and the correct optimal value at a precision higher than 10^{-4}. Therefore, there is no need to solve the SDP problem of order $k = 4$. Moreover, at the order $k = 2$, NLS gives already a good approximation within an accuracy of 10^{-1} that can be enough in several applications. The same conclusion holds for the optimal value: it is clear from Table 7.7 that the optimality is not reached in the hierarchy at $k = 3$ since the obtained optimal value is -2.94 instead of -3. However, using NLS as an extraction method yields the correct optimal value with an accuracy higher than 10^{-4}. We yet remark that the lower bounds $\mathcal{J}_1^*$, $\mathcal{J}_2^*$, and $\mathcal{J}_3^*$ all equal the optimal value -3.

Table 7.6: *Extraction of solution of Problem (7.7) using NLS (accuracy of* 10^{-4}*)*

	$k=1$	$k=2$	$k=3$
x_1^*	$\begin{pmatrix}-0.0116\\-0.0110\\-0.0459\end{pmatrix}$	$\begin{pmatrix}0\\0\\0\end{pmatrix}$	$\begin{pmatrix}0\\0\\0\end{pmatrix}$
x_2^*	$\begin{pmatrix}1.0835\\-0.1286\\-0.2224\end{pmatrix}$	$\begin{pmatrix}1.9998\\0.0006\\0.0015\end{pmatrix}$	$\begin{pmatrix}2\\0\\0\end{pmatrix}$
x_3^*	$\begin{pmatrix}-0.2187\\1.1305\\-0.2055\end{pmatrix}$	$\begin{pmatrix}0.0002\\1.9994\\0.0015\end{pmatrix}$	$\begin{pmatrix}0\\2\\0\end{pmatrix}$
x_4^*	$\begin{pmatrix}-0.1728\\-0.1758\\0.9500\end{pmatrix}$	$\begin{pmatrix}0.0002\\0.0006\\1.9985\end{pmatrix}$	$\begin{pmatrix}0\\0\\2\end{pmatrix}$
x_5^*	$\begin{pmatrix}1.3728\\1.2453\\-0.3417\end{pmatrix}$	$\begin{pmatrix}2.0006\\2.0017\\-0.0044\end{pmatrix}$	$\begin{pmatrix}2\\2\\0\end{pmatrix}$
x_6^*	$\begin{pmatrix}1.4839\\-0.3589\\1.5066\end{pmatrix}$	$\begin{pmatrix}2.0006\\-0.0017\\2.0044\end{pmatrix}$	$\begin{pmatrix}2\\0\\2\end{pmatrix}$
x_7^*	$\begin{pmatrix}-0.3968\\1.5726\\1.5603\end{pmatrix}$	$\begin{pmatrix}-0.0006\\2.0017\\2.0044\end{pmatrix}$	$\begin{pmatrix}0\\2\\2\end{pmatrix}$
x_8^*	$\begin{pmatrix}4.7932\\5.2755\\4.7767\end{pmatrix}$	$\begin{pmatrix}1.9984\\1.9957\\1.9862\end{pmatrix}$	$\begin{pmatrix}2\\2\\2\end{pmatrix}$

Table 7.7: *Value of the criterion at the extracted global minima (accuracy of* 10^{-4}*)*

	$k=1$	$k=2$	$k=3$
Using NLS	-3.1395	-3.0001	-3.0000
	-2.7751	-2.9954	-3.0000
	-2.9556	-2.9954	-3.0000
	-2.7604	-2.9954	-3.0000
	-1.9994	-3.0135	-3.0000
	-2.3375	-3.0135	-3.0000
	-2.5927	-3.0135	-3.0000
	-4.6932	-2.9666	-3.0000
Using GloptiPoly	-1.2762	-2.5297	-2.9401
Lower bound from GloptiPoly	-3	-3	-3

7.6.3.2 Blind source separation application

We consider a blind source separation problem similar to the one of Section 7.4.2.3. Namely, we observe a vector $\mathbf{y}$ according to the model

$$\mathbf{y} = \mathbf{A}\mathbf{s} + \mathbf{w}$$

where $\mathbf{A} \in \mathbb{R}^{n \times R}$ is a mixing matrix, $\mathbf{w}$ is a white Gaussian noise with zero-mean, and $\mathbf{s}$ is a vector of R independent random variables. The vectors $\mathbf{s}$ and $\mathbf{w}$ as well as the matrix $\mathbf{A}$ are all unknown. However, we assume in the following that $\mathbf{w}$ and $\mathbf{s}$ are independent.

We aim at recovering the source vector $\mathbf{s}$ from a given number of samples of $\mathbf{y}$. To do so, we look for a left inverse of the mixing matrix $\mathbf{A}$. We hence want to solve the following polynomial optimization problem [Com10]:

$$\begin{array}{ll} \underset{\mathbf{u} \in \mathbb{R}^n}{\text{maximize}} & \left|\text{Cum}^4(\mathbf{u}^\top \mathbf{y})\right| \\ \text{subject to} & \mathbf{u}^\top \mathbf{R}_\mathbf{y} \mathbf{u} = 1\,, \end{array} \tag{7.8}$$

where $\mathbf{R}_\mathbf{y}$ is the covariance matrix of $\mathbf{y}$ and $\left|\text{Cum}^4(\mathbf{u}^\top \mathbf{y})\right|$ is given by:

$$\left|\text{Cum}^4(\mathbf{u}^\top \mathbf{y})\right| = \sum_{(i,j,k,l) \in [\![1,Q]\!]^4} \left| u_i u_j u_k u_l \text{Cum}^4(\mathbf{y})_{i,j,k,l} \right| .$$

If the distribution $\mathbf{s}$ has independent components and is not Gaussian, then its 4^{th} order cumulants are nonzero and all global solutions $\mathbf{u}_*$ of Problem (7.8) are such that

$$(\exists r \in [\![1, R]\!]) \quad \left|\mathbf{u}_*^\top \mathbf{A}\right| = \mathbf{e}_r\,, \tag{7.9}$$

where $\mathbf{e}_r$ is the vector of $\mathbb{R}^R$ whose elements are all equal to zero except its r-th element which is 1. Note that Problem (7.8) is a polynomial problem which has R global solutions ($2R$ if the sign indeterminacy is taken into account). In practice, we use the empirical estimates of the cumulant tensor $\text{Cum}^4(\mathbf{y})$ and of the covariance matrix $\mathbf{R}_\mathbf{y}$.

Let us illustrate our methodology with $R = 5$ and $n = 5$. We draw the elements of $\mathbf{A}$ according to a uniform distribution on $[0, 1]$ and the elements of $\mathbf{s}$ take value -1 or 1 with equal probability. The variance of the noise $\mathbf{w}$ is set to 0.1. We generate 1,000,000 samples of the vector $\mathbf{y}$ and estimate the covariance matrix $\mathbf{R}_\mathbf{y}$ and the cumulant $\text{Cum}^4(\mathbf{y})$. We then solve Problem (7.8) using GloptiPoly [HLL09] and SDPT3 [TTT99]. From relaxation order 3 to relaxation order 7, GloptiPoly is not able to recover the global solutions as the extraction method from [HL05] fails. We only obtain a noisy truncated moment vector and a lower bound of the optimal value. Nevertheless, since Problem (7.8) has many solutions, we cannot directly read the first order moments. Moreover, computation limits prohibit the use of higher relaxation orders. We therefore build the tensor corresponding to the truncated moment vector for the relaxation order 3 and perform a CPD on the latter tensor using the implementation of NLS from Tensorlab [VDS+16]. The solutions of (7.8) are obtained by reading the generating vector of the CPD.

NLS requires as input, the tensor and the rank of the CPD, which is the number of solutions of Problem (7.8). In our numerical application, the theoretical number of solutions is $2R = 10$. Table 7.8 shows the ten solutions obtained after performing the CPD together with their corresponding vector $\mathbf{e}_r$, the value of the criterion and the value of the constraint. Note that the lower bound at the order 3 has the value -2 which confirm we have obtained global solutions.

Remark: If the dimension n is greater than R, than there is a continuum of solutions to Problem (7.8). Indeed, for any solution $\mathbf{w}_*$, the affine space $\mathbf{w}_* + \text{Ker}(\mathbf{A}^\top)$ is also a solution. However, if additional constraints are considered, it is possible to use our proposed method.

Table 7.8: *Solutions to Problem (7.8) extracted from the SDP relaxation of order 3 using a CPD ($n = 5$, $R = 5$)*

Solution $\mathbf{u}_*$	Vector $\mathbf{e}_r$	Criterion value	Constraint value
$\begin{pmatrix} -1.1225 \\ 0.4673 \\ 1.0796 \\ 1.1697 \\ -0.3323 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ -1 \cdot 10^{-6} \end{pmatrix}$	-2.0000	1.0000
$\begin{pmatrix} 1.1225 \\ -0.4673 \\ -1.0796 \\ -1.1697 \\ 0.03323 \end{pmatrix}$	$\begin{pmatrix} -1 \\ 0 \\ 0 \\ 0 \\ 1 \cdot 10^{-6} \end{pmatrix}$	-2.0000	1.0000
$\begin{pmatrix} -0.1056 \\ 1.1335 \\ -0.1775 \\ -1.8194 \\ 0.8228 \end{pmatrix}$	$\begin{pmatrix} -1 \cdot 10^{-6} \\ 1 \\ -3 \cdot 10^{-6} \\ 0 \\ -3 \cdot 10^{-6} \end{pmatrix}$	-2.0000	1.0000
$\begin{pmatrix} 0.1056 \\ -1.1335 \\ 0.1775 \\ 1.8194 \\ -0.8228 \end{pmatrix}$	$\begin{pmatrix} 1 \cdot 10^{-6} \\ -1 \\ 3 \cdot 10^{-6} \\ 0 \\ 3 \cdot 10^{-6} \end{pmatrix}$	-2.0000	1.0000
$\begin{pmatrix} 0.2016 \\ 2.1834 \\ -1.4173 \\ 0.6765 \\ -0.7927 \end{pmatrix}$	$\begin{pmatrix} 3 \cdot 10^{-6} \\ -7 \cdot 10^{-6} \\ 1 \\ 0 \\ -1 \cdot 10^{-6} \end{pmatrix}$	-1.9999	1.0000
$\begin{pmatrix} -0.2016 \\ -2.1834 \\ 1.4173 \\ -0.6765 \\ 0.7927 \end{pmatrix}$	$\begin{pmatrix} -3 \cdot 10^{-6} \\ 7 \cdot 10^{-6} \\ -1 \\ 0 \\ 1 \cdot 10^{-6} \end{pmatrix}$	-1.9999	1.0000
$\begin{pmatrix} 1.8057 \\ 2.1418 \\ -1.2604 \\ -3.2253 \\ -0.8676 \end{pmatrix}$	$\begin{pmatrix} -2 \cdot 10^{-5} \\ 1 \cdot 10^{-6} \\ 2 \cdot 10^{-6} \\ 1 \\ 7 \cdot 10^{-6} \end{pmatrix}$	-1.9999	1.0000
$\begin{pmatrix} -1.8057 \\ -2.1418 \\ 1.2604 \\ 3.2253 \\ 0.8676 \end{pmatrix}$	$\begin{pmatrix} -2 \cdot 10^{-5} \\ -1 \cdot 10^{-6} \\ -2 \cdot 10^{-6} \\ -1 \\ -7 \cdot 10^{-6} \end{pmatrix}$	-1.9999	1.0000
$\begin{pmatrix} 0.0071 \\ -4.9901 \\ 2.0384 \\ 2.4953 \\ 1.5310 \end{pmatrix}$	$\begin{pmatrix} -6 \cdot 10^{-6} \\ 5 \cdot 10^{-6} \\ 2 \cdot 10^{-5} \\ 4 \cdot 10^{-5} \\ 1 \end{pmatrix}$	-1.9998	0.9999
$\begin{pmatrix} -0.0071 \\ 4.9901 \\ -2.0384 \\ -2.4953 \\ -1.5310 \end{pmatrix}$	$\begin{pmatrix} 6 \cdot 10^{-6} \\ -5 \cdot 10^{-6} \\ -2 \cdot 10^{-5} \\ -4 \cdot 10^{-5} \\ -1 \end{pmatrix}$	-1.9998	0.9999

This application illustrates that our proposed extraction method allows to retrieve all the global solutions at a low order of relaxation in contrast to the standard one.

§ 7.7 Summary

In this chapter, we have studied the link between tensor decompositions, the moment problem and polynomial optimization.

We have first reformulated the CPD of a symmetric tensor into a moment problem. By using the moment matrix, we have provided a tensor rank detection method as well as an algebraic algorithm to extract the CPD generating vectors. Our rank detection method relies on the estimation of the rank of the moment matrix, which has a smaller dimension compared to the matrix obtained by unfolding the tensor. Our proposed methods both provide theoretical guarantee and are highly accurate in a noise free context.

Then, we have proposed to use optimization-based tensor decomposition methods in order to perform the extraction step in Lasserre's hierarchy. While the standard extraction method works well with a sufficiently large and accurate moment vector, it often returns wrong results if the latter is noisy. Interpreting the extraction as a tensor decomposition, we have used optimization-based CPD methods that are more resilient to noise. As a result, we have solved polynomial optimization problem at a lower relaxation order, thus decreasing the computational complexity of the relaxed SDP problem.

- Chapter 8 -

Conclusion

§ 8.1 Summary

Within this thesis, our main focus has been on the application of rational optimization to different signal processing applications. This has led us to investigate three main problems.

- First, we have used the recent advances in polynomial optimization to solve intricate signal processing problems. While such problems are usually solved approximately, our goal has been to tackle them directly. Our objective has been to find the exact global solutions to the problem.

  In Chapter 3, we have started by recalling the basis of Lasserre's hierarchy that relaxes a rational problem into a sequence of convex problems whose optimal values converge to the one of the initial problem. This method is especially interesting as it provides theoretical guarantees to recover all the global minimizers. Then, we showed through a complexity study, how the structure of the problem can decrease the overall computation and memory complexity. More specifically, if the objective function can be expressed as a sum of rational functions sharing only a few variables, the complexity can be decreased drastically.

  In Chapter 4, we have applied the method developed in Chapter 3 to an inverse problem that arises in chromatography. Namely, we have considered the reconstruction of a sparse signal deteriorated by linear and nonlinear degradations, a Gaussian noise, and a subsampling. We have proposed criteria involving nonconvex but exact relaxation of the ℓ_0 function in order to promote sparsity. We have shown that those criteria are all piecewise rational and thus, the corresponding variational problems can be reformulated as rational optimization problems. Moreover, we have proved that the latter problems have a structure that can be exploited using our method from Chapter 3. Then, we have derived the computational complexity of our optimization method and we have discussed several ways to decrease it. Numerical simulations illustrate the domain of applicability of the method and its high potential for finding a good approximation to a global minimum.

  In Chapter 5, we have extended the method developed in Chapter 4 in several directions. We have first adapted our method to handle Poisson-Gaussian noise instead of Gaussian noise. We have also proposed a more accurate approximation to the log-likelihood of Poisson-Gaussian distribution based on an accurate rational approximation of the log term. Moreover, instead of considering only sparse

signals, we have extended our method to signals that are sparse in a transformed domain. We have shown that our approach provides better reconstruction results than standard ones for two different types of application, namely the reconstruction of sparse signals and of visible light communication signals.

On the other hand, we have extended the model of Chapter 4 in order to make it robust to outliers. Especially, we have demonstrated how to handle various common convex and nonconvex robust functions in our criterion. Moreover, we have considered a different structure on the sought signal: we have assumed here that the signal is in a union of subsets. The latter assumption results in nonconvex constraints that are hard to handle in the state-of-the-art methods whereas it has several applications. We have expressed this assumption in a polynomial form and solved the overall problem using the method of Chapter 3. We finally have provided numerical simulations that illustrate the good quality of the signal reconstructed through our method.

- In a second part, we have studied several proximal algorithms to solve large-scale SDP problems. Although interior point methods are the current state-of-the-art solvers for SDP problems, they are quickly limited when they have to deal with the high-dimensional SDP relaxations of rational problems. They are currently the bottleneck of the method developed in Chapter 3 and our goal was therefore to find a more adapted algorithm to treat such SDP problems.

  In Chapter 6, we have applied several common proximal algorithms as well as some more advanced ones on the canonical formulations of SDP problems. We have applied these methods on the primal and the dual problems. We also applied primal-dual methods. Then, we have proposed several reformulations of SDP problems in order to speed up the convergence. Two second-order methods inspired by interior point algorithms have also been developed. Finally, numerical comparisons between the proposed methods and an interior point method implementation have been conducted. They illustrate that proximal methods outperform interior point methods when the number of linear constraints m is high while the dimension of the semi-definite constraint n is kept low. Among them, ADMM and ADAL are especially efficient. However, they also show that when both m and n are high, interior point methods are still faster and more accurate than proximal methods. Unfortunately, the latter case corresponds to the SDP relaxations of the rational problems we have considered in Chapters 4 and 5.

- Finally, we have explored the connection between polynomial optimization and tensor decomposition through the moment problem. More specifically, we have shown that the extraction step in polynomial optimization and the canonical polyadic decomposition of a symmetric tensor are both instances of the moment problem.

  In Chapter 7, we have explored this fruitful link in two different ways. By interpreting the CPD of a symmetric tensor as a moment problem, we have proposed a method to find the symmetric rank of a symmetric tensor by using tools related to truncated moment problems. We have hence obtained a necessary condition to deduce the tensor rank. Numerical simulations have been performed on a blind source separation example. We have also proposed an algebraic method that guarantees to recover the generating vectors of the CPD. Our method has shown to be quite competitive with other common decomposition algorithms in a noiseless context.

  On the other hand, we have proposed an alternative approach to the standard method for performing the extraction of the solutions in polynomial optimization.

By interpreting the extraction as a tensor decomposition problem, we have used robust methods to perform CPD in order to extract the global solutions at lower relaxation orders. Using robust extraction allows to use lower relaxation orders in the method of Chapter 3 and thus, alleviate the computational burden. This point has been illustrated along a case study.

In the next section, we suggest several directions to extend the aforementioned results that could be investigated in future works.

§ 8.2 Perspectives

We propose here some possible extensions of our signal reconstruction framework of Chapter 4.

Extension to longer signals: Using the structure of signal processing problems, we showed that we could handle medium size signals. The next step would be to handle large signals. The signal could be split into overlapping chunks of small size before we apply our framework to reconstruct each chunk. The whole signal could then be reconstructed with a method to piece together the different chunks. For instance, the overlapping part of each chunk could be averaged. This point has been sketched in Section 4.6.4.3 but a thorough study should be conducted.

Extension to 2D signals: Our signal reconstruction method could be extended for images. The latter could be split into overlapping small patches, say of size of 15×15, before applying our method. The patches would be then glued together by averaging the values of overlapping pixels.

Extension to complex signals: By interpreting complex numbers as a vector of two real ones, our reconstruction framework could be used to reconstruct complex signals such as the ones obtained after a wavelet of a Fourier transform. A successful application of Lasserre's hierarchy to handle complex variables and data can be found in the work of Josz [Jos16].

Extension to trigonometric polynomials: In this work, we have considered only polynomial expressed in the canonical basis. However, the use of trigonometric polynomials is particularly fruitful in many applications such as filter design [Dum07]. Trigonometric polynomials can also a suitable way to handle complex signals.

Comparison of the different criteria to find the global minima of $f_{\mathbf{y}} + \lambda\ell_0$: In order to quantify the improvement of our method, it would be interesting to compare the minimization of $f_{\mathbf{y}} + \mathcal{R}_\lambda$ with the four nonconvex regularizers given in Section 4.2.2 and the minimization of $f_{\mathbf{y}} + \lambda\ell_1$. All the minimizations have to be done with our proposed method in order to show the benefit of the nonconvex regularizations to promote sparsity.

Moreover, a comparison of the minimization of the criterion $f_{\mathbf{y}} + \mathcal{R}_\lambda$ with the minimization of $f_{\mathbf{y}}^{\text{lin}} + \lambda\ell_1$ and $f_{\mathbf{y}}^{\text{lin}} + \lambda\ell_0$ could be performed, where $f_{\mathbf{y}}^{\text{lin}}$ is a linearization of the function $f_{\mathbf{y}}$. This comparison could show the interest of considering a nonlinear function Φ in our model instead of a linearization of it.

In Chapter 5, we considered Poisson-Gaussian noise as well as outliers. We suggest here some research direction concerning other noise models that can be handled by our framework. Moreover, we provide possible extensions of the union of subsets model we considered.

Reconstruction under impulse noise: Recent approaches to handle impulse noise following Levy alpha-stable distributions rely on rational data fit and rational regularization terms of low degrees [CF18, CHN04] This particularly fits the framework we developed in the first part of this thesis.

Mixture of models: The union of subsets model could also be used to handle mixture of models. For instance, we could consider a collection of models that fit different part of a signal. However, we would not know which model corresponds to a given part of the signal. This problem can be formulated as a union of subspaces problem to which our framework could apply.

In Chapter 6, we explored several algorithms to solve the high-dimensional SDP we obtain from our framework. However, they were not able to outperform interior point methods. We suggest some directions which could improve our proposed proximal algorithms.

Burer-Monteiro factorization: The Burer-Monteiro factorization consists in replacing the symmetric positive semidefinite variable $\mathbf{X}$ in a SDP problem with the rectangular matrix $\mathbf{U}$ such that $\mathbf{X} = \mathbf{U}\mathbf{U}^\top$. It is particularly well adapted when a low rank solution is sought as the matrix $\mathbf{U}$ then has only a few columns. Nevertheless, it leads to a nonconvex problem that is then solved with standard proximal methods that guarantees only to find a local extremum. Some recent works have been searching for conditions for which retrieving global optimum is guaranteed [WW18, BVB19]. Together with the robust extraction proposed in Chapter 7, it could be an alternative to solve SDP relaxations for Lasserre's hierarchy.

Acceleration scheme: Some acceleration schemes for ADMM and ADAL have been proposed recently [PL19]. Although they have not been designed to solve SDP problems specifically, these accelerations could speed up ADMM and ADAL enough to beat interior point methods at solving our SDP relaxations.

Loop unrolling: Unrolling several iterations of iterative algorithms before replacing them by layers of a convolutional neural network has recently shown great success [BCC+20, MJU17]. The main benefit of this method is to adapt parameters for each iteration. It could be used to replace our proximal algorithms in order to solve high-dimensional SDP problems faster than interior point methods. However, the number of iterations to unroll is hard to determine at first glance and a correct data set of SDP problems needs to prepared in order to train the network.

Distributed SDP solvers: A recent trend to solve high-dimensional SDP problems are the distributed solvers [MKL18, KL15]. Those approaches could be used to handle the high-dimensional SDP relaxations emerging form Lasserre's framework. However, they require a dedicated computation infrastructure and cannot be used on a standard laptop or desktop.

In Chapter 7, we limit our studies to tensors defined on product of real spaces. However, the CPD and its associated algorithms like NLS are well-defined for tensors with complex values. An extension of our work to complex-valued tensors could be performed.

Appendices

- Appendix A -

Complexity of the relaxed SDP problems

We detail here the computation of the complexity of the SDP relaxation of Chapter 4, i.e. the quadruple $(n, m, m_{\mathrm{s}}, \ell)$, depending on the initial data like U, L and T as well as the relaxation order k.

§ A.1 Number of blocks

The number of blocks in the matrices $\mathbf{C}$ and $(\mathbf{A}_i)_{i\in[\![1,m]\!]}$ in (3.8) is given in Equation (3.13). In order to solve (4.13), we introduce $U+T$ measures and thus $\tilde{I} = U+T$ in (3.13). We now determine the number of localizing matrices η_i in each measure which is equal to the number of polynomial constraints defining the set $\mathcal{K}_i$. Equation (4.15) gives T_i constraints for the definition of each set $\mathcal{K}_i$ associated to the measures $(\mu_u)_{u\in[\![1,U]\!]}$ while (4.16) gives $1+3I$ constraints for each set $\mathcal{K}_i$ associated to the measures $(\nu_t)_{t\in[\![1,T]\!]}$. Indeed the polynomial equality constraint in (4.16) is translated into two polynomial inequality constraints. The first $L-1$ measures $(\mu_u)_{u\in[\![1,U]\!]}$ are defined on a number T_i of variables smaller than L due to the convolution filter. In the following, we neglect it for the sake of clarity and assume that T_i is equal to L for all the measure $(\mu_u)_{u\in[\![1,U]\!]}$. Considering the above elements, Equation (3.13) finally becomes

$$m_{\mathrm{s}} = U(1+L) + T(2+3I)\,.$$

It is interesting to notice that the relaxation order k does not have any effect on the number of blocks; it only increases the size of the blocks.

§ A.2 Number of linear equality constraints

We then count the number of linear equality constraints in (3.12), without considering the redundant ones. For u belonging to $[\![1, U-1]\!]$, θ_u denotes the overlap parameter defined as the number of variables shared between g_u and g_{u+1}. Note that θ_u depends on u but also on the length of the filter L and on the parameter of the decimation δ. Furthermore, we remark that all the rational functions $(g_u)_{u\in[\![1,U]\!]}$ have same degree at their numerator and denominator. We denote their denominator by q_u, and we define $\mathrm{d}_q = \mathrm{d}_{q_u}$.

Following Section 3.2.4, we need to consider equality of moments of monomials in θ_u variables up to degree $2(k-d_q)$, which gives $\binom{\theta_u+2(k-d_q)}{2(k-d_q)}$ equality constraints for every u

in $[\![1, U-1]\!]$ on consecutive measures $(\mu_u)_{u\in[\![1,U]\!]}$. Adding the linear constraints linking moment related to $(\mu_u)_{u\in[\![1,U]\!]}$ and $(\nu_t)_{t\in[\![1,T]\!]}$, we finally obtain

$$\ell = \sum_{u=1}^{U-1} \binom{\theta_u + 2(k-d_q)}{2(k-d_q)} + 2(k-d_\zeta)T\,,$$

where d_ζ corresponds to (1.1) for the maximal degree of the denominator of rational function $(\zeta_i)_{i\in[\![1,I]\!]}$. The impact of linear equality constraints on the computational time of SDP solver is minor compared to n, m and m_s.

§ A.3 Dimension of the global moment vector

Considering $(\mu_u)_{u\in[\![1,U]\!]}$ as U measures on L variables and $(\nu_t)_{t\in[\![1,T]\!]}$ as T measures on $1+I$ variables, it follows from Equation (3.14) that

$$m = U\binom{L+2k}{2k} + T\binom{1+I+2k}{2k}\,.$$

§ A.4 Dimension of the semi-definite constraint

The measures $(\mu_u)_{u\in[\![1,U]\!]}$ are on L variables while the measures $(\nu_t)_{t\in[\![1,T]\!]}$ are on $1+I$ variables. Since all the polynomial constraints defining the sets $(\mathcal{K}_i)_{i\in[\![1,U+T]\!]}$ are linear or quadratic, the value of d_{s_j} in (3.15) is equal to 1 in this case. Finally, Equation (3.15) yields

$$n = U\left(\binom{L+k}{k} + L\binom{L+k-1}{k-1}\right) + T\left(\binom{1+I+k}{k} + (1+3I)\binom{I+k}{k-1}\right)\,.$$

- Appendix B -

Optimization background

Let $(\mathcal{H}, \langle . \mid . \rangle_{\mathcal{H}})$ be a Hilbert space.

§ B.1 Monotone Operators

We recall here some definitions for operators.

Definition 1 (Power set)
Let $\mathcal{X}$ be a set.
The power set $2^{\mathcal{X}}$ of $\mathcal{X}$ is the family of all subsets of $\mathcal{X}$.

Definition 2 (Graph of an operator)
Let $\mathcal{M}$ be an operator from the set $\mathcal{X}$ into the power set $2^{\mathcal{Y}}$.
The graph of $\mathcal{M}$ is defined as

$$\operatorname{gra} \mathcal{M} = \{(x, y) \in \mathcal{X} \times \mathcal{Y} \mid y \in \mathcal{M}(x)\}$$

Definition 3 (Monotone operator)
Let $\mathcal{M}$ be an operator from $\mathcal{H}$ into its power set $2^{\mathcal{H}}$.
$\mathcal{M}$ is monotone if

$$(\forall (x, u) \in \operatorname{gra} \mathcal{M})(\forall (y, v) \in \operatorname{gra} \mathcal{M}) \quad \langle x - y \mid u - v \rangle_{\mathcal{H}} \geq 0$$

Definition 4 (Maximally monotone operator)
Let $\mathcal{M}$ be an operator from $\mathcal{H}$ into its power set $2^{\mathcal{H}}$.
$\mathcal{M}$ is maximally monotone if there is no monotone operator $\mathcal{T}$ from $\mathcal{H}$ into $2^{\mathcal{H}}$ such that $\operatorname{gra} \mathcal{T}$ *contains* $\operatorname{gra} \mathcal{M}$.

Definition 5 (Cocoercive operator)
Let β a strictly positive real, $\mathcal{D}$ be a non-empty subset of $\mathcal{H}$ and let $\mathcal{M}$ be an operator from $\mathcal{D}$ to $\mathcal{H}$.
$\mathcal{M}$ is β-cocoercive if $\beta\mathcal{M}$ is firmly non-expansive,

$$(\forall (x, y) \in \mathcal{D} \times \mathcal{D}) \quad \langle x - y \mid \mathcal{M}(x) - \mathcal{M}(y) \rangle_{\mathcal{H}} \geq \beta \left\| \mathcal{M}(x) - \mathcal{M}(y) \right\|^2 .$$

Definition 6 (Resolvent of an operator)
Let $\mathcal{M}$ be an operator on $\mathcal{H}$.
We define the monotone operator $J_{\mathcal{M}}$ associated with the operator $\mathcal{M}$ such that

$$J_{\mathcal{M}} = (\mathcal{M} + \mathrm{Id})^{-1} \tag{B.1}$$

This operator is the resolvent of $\mathcal{M}$.

§ B.2 Convexity and subdifferential

Some definitions and properties related to convex optimization are remind in this section.

Definition 7 (Convex conjugate)
Let $f : \mathcal{H} \to \mathbb{R}$ be a function.
Its convex conjugate f^* is defined as:

$$(\forall y \in \mathcal{H}) \quad f^*(y) = \sup_{x \in \mathcal{H}} \langle y \mid x \rangle_{\mathcal{H}} - f(x)$$

Definition 8 (Moreau subdifferential)
Let $f : \mathcal{H} \to]-\infty; +\infty]$ be a proper function.
The Moreau subdifferential of f, denoted ∂f, is defined as:

$$(\forall x \in \mathcal{H}) \quad \partial f(x) = \{u \in \mathcal{H} \mid (\forall y \in \mathcal{H}) \langle y - x \mid u \rangle_{\mathcal{H}} + f(x) \leq f(y)\}$$

Property 1 (Monotonicity of the subdifferential)
Let $f : \mathcal{H} \to]-\infty; +\infty]$ be a proper function.
Then ∂f is a monotone operator.

Property 2

$$u \in \partial f(x) \quad \Longrightarrow \quad x \in \partial f^*(u)$$

Corollary 3
Let f be a proper convex lower semi-continuous function.
Then,

$$u \in \partial f(x) \quad \Longleftrightarrow \quad x \in \partial f^*(u)\,.$$

It implies that $(\partial f)^{-1} = \partial f^*$.

Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be two Hilbert spaces., f and g be two proper functions respectively from $\mathcal{H}_1$ to $\mathbb{R} \cup \{\infty\}$ and from $\mathcal{H}_1$ to $\mathbb{R} \cup \{\infty\}$, and L be a bounded linear operator from $\mathcal{H}_1$ to $\mathcal{H}_2$.

Property 3 (Addition and pre-composition with linear mapping rule)
If f and g are convex semi-continuous and $\mathrm{int}(\mathrm{dom}(g)) \cap L(\mathrm{dom}(f))$ *is not empty*
Then,

$$(\forall x \in \mathcal{H}_1) \quad \partial f(x) + L^*(\partial g(L(x))) = \partial(f + g \circ L)(x)$$

Definition 9 (Proximal operator)
Let $f : \mathcal{H} \to]-\infty; +\infty]$ be a proper convex function.
The proximal operator of f is defined as:

$$(\forall x \in \mathcal{H}) \quad \mathrm{prox}_f(x) = \operatorname*{arg\,min}_{y \in \mathcal{H}} f(y) + \frac{1}{2} \|y - x\|_2^2$$

Property 4 (Subgradient characterisation of the proximal operator)
The proximal operator can be characterized using the subdifferential:

- $v = \mathrm{prox}_f(x) \quad \Longleftrightarrow \quad 0 \in \partial f(v) + v - x$
- $v = \mathrm{prox}_{\lambda f}(x) \quad \Longleftrightarrow \quad 0 \in \partial f(v) + \frac{1}{\lambda}(v - x)$
- $v = \mathrm{prox}_f^B(x) \quad \Longleftrightarrow \quad 0 \in \partial f(v) + B(v - x)$

Remark: (Link between proximal operator and resolvent)

$$\begin{aligned} v = \operatorname{prox}_f(x) &\iff 0 \in \partial f(v) + v - x \\ &\iff x \in v + \partial f(v) \\ &\iff x \in (\partial f + \mathrm{Id})(v) \\ &\iff v = (\partial f + \mathrm{Id})^{-1}(x) \end{aligned}$$

Hence prox_f is equal to $(\partial f + \mathrm{Id})^{-1}$ which is $J_{\partial f}$, the resolvent of ∂f.

Property 5 (Moreau's decomposition formula)
This formula generalises the decomposition by orthogonal projection on subspaces.

- $x = \operatorname{prox}_f(x) + \operatorname{prox}_{f^*}(x)$
- $x = \operatorname{prox}_{\lambda f}(x) + \lambda \operatorname{prox}_{\lambda^{-1} f^*}(\lambda^{-1} x)$
- $x = \operatorname{prox}_f^B(x) + B^{-1} \operatorname{prox}_{f^*}^{B^{-1}}(Bx)$

§ B.3 Some formulae

We recall in this section some common formulas that are used in Chapter 6.

- Projection on an affine set $\mathcal{X} = \{x \in \mathcal{H} \mid Ax = b\}$:

$$\Pi_{\mathcal{X}}(x) = x - A^*(AA^*)^{-1}(Ax - b)$$

- Subdifferential of an indicator function of a set $\mathcal{X}$:

$$\partial \iota_{\mathcal{X}}(x) = \mathcal{N}_{\mathcal{X}}(x)$$

- Convex conjugate of an indicator function of a set $\mathcal{X}$:

$$\iota_{\mathcal{X}}^*(y) = \sup_{x \in \mathcal{X}} \langle y \mid x \rangle_{\mathcal{H}}$$

- Resolvent at x of the subdifferential of an indicator function of a set: $\mathcal{X}$

$$J_{\gamma \partial \iota_{\mathcal{X}}}(x) = \Pi_{\mathcal{X}}(x)$$

- If $\mathcal{X}$ is a non-empty convex cone, then

$$(x - z) \in \mathcal{N}_{\mathcal{X}}(z) \iff z = \Pi_{\mathcal{X}}(x)$$

§ B.4 Proximal algorithms

We present in this section, the proximal algorithms used in Chapter 6.

- Douglas-Rachford Splitting [LM79]:
  Let $\mathcal{H}$ be an Hilbert space and $\mathcal{A}$ and $\mathcal{B}$ two monotone operators on $\mathcal{H}$.
  We want to solve the following monotone inclusion problem

  Find x in $\mathcal{H}$ such that

$$0 \in \mathcal{A}(x) + \mathcal{B}(x)\,. \tag{B.2}$$

Algorithm 20: Douglas-Rachford algorithm

Input: Set $z^{(0)} \in \mathcal{H}$
Output: x, solution of (B.2)
for $k = 0,1,\dots$ **do**
  $z^{(k+1)} \leftarrow J_{\gamma\mathcal{A}}\left(2J_{\gamma\mathcal{B}}z^{(k)} - z^{(k)}\right) + z^{(k)} - J_{\gamma\mathcal{B}}z^{(k)}$;
$x \leftarrow J_{\gamma\mathcal{B}}z^{(k+1)}$;

- Chambolle-Pock Algorithm [CP10]:
Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be two Hilbert spaces, $\mathcal{A}$ and $\mathcal{B}$ two monotone operators on respectively $\mathcal{H}_1$ and $\mathcal{H}_2$, and L a linear operator from $\mathcal{H}_1$ to $\mathcal{H}_2$.
We want to solve the following monotone inclusion problem

Find x in $\mathcal{H}_1$ such that
$$0 \in \mathcal{A}(x) + \mathcal{B} \circ L(x)\,. \tag{B.3}$$

Algorithm 21: Chambolle-Pock algorithm

Input: Set $(x^{(0)}, y^{(0)}) \in \mathcal{H}_1 \times \mathcal{H}_2$
Input: Set $\theta \in [0,1]$
Output: x, solution of (B.3)
$\bar{x}^{(0)} = x^{(0)}$;
for $k = 0,1,\dots$ **do**
  $y^{(k+1)} \leftarrow J_{\sigma\mathcal{B}^*}\left(y^{(k)} + \sigma L\bar{x}^{(k)}\right)$;
  $x^{(k+1)} \leftarrow J_{\tau\mathcal{A}}\left(x^{(k)} - \tau L^* y^{(k+1)}\right)$;
  $\bar{x}^{(k+1)} \leftarrow x^{(k+1)} + \theta\left(x^{k+1} - x^k\right)$;
$x \leftarrow \bar{x}^{(k+1)}$;

- Forward-backward algorithm [BC11]:
Let $\mathcal{H}$ be an Hilbert space, $\mathcal{A}$ is a maximally monotone operator on $\mathcal{H}$ and $\mathcal{B}$ is a β-cocoercive operator on $\mathcal{H}$.
We want to solve the following monotone inclusion problem

Find x in $\mathcal{H}$ such that
$$0 \in \mathcal{A}(x) + \mathcal{B}(x)\,. \tag{B.4}$$

Algorithm 22: Forward-Backward algorithm

Input: Set $x^{(0)} \in \mathcal{H}$
Input: Set $\gamma \in]0, 2\beta[$
Output: x, solution of (B.4)
for $k = 0,1,\dots$ **do**
  $x^{(k+1)} \leftarrow J_{\gamma\mathcal{A}}\left(x^{(k)} - \gamma Bx^{(k)}\right)$;

- Forward-Backward Half-Forward Algorithm [BAD18]:
Let $\mathcal{H}$ be an Hilbert space and $\mathcal{A}$, $\mathcal{B}_1$ and $\mathcal{B}_2$ are operators on $\mathcal{H}$.
We assume that $\mathcal{A}$ is maximally monotone, $\mathcal{B}_1$ is β-cocoercive and $\mathcal{B}_2$ is K_L-lipschitz and monotone.
We want to solve the following monotone inclusion problem

Find x in $\mathcal{H}$ such that
$$0 \in \mathcal{A}(x) + \mathcal{B}_1(x) + \mathcal{B}_2(x)\,. \tag{B.5}$$

Let $\mathcal{C}$ be a closed convex and non empty subset of an Hilbert space $\mathcal{H}$.

Algorithm 23: FBHF

Input: Set $(x^{(0)}, z^{(0)}) \in \mathcal{H} \times \mathcal{H}$
Input: Set $\gamma \in [\epsilon, \chi - \epsilon]$
Output: x, solution of (B.5)

for $k = 0,1,\ldots$ **do**
  $x^{(k+1)} \leftarrow J_{\gamma\mathcal{A}}\left(z^{(k)} - \gamma(\mathcal{B}_1 + \mathcal{B}_2)z^{(k)}\right)$;
  $z^{(k+1)} \leftarrow \Pi_{\mathcal{C}}\left(x^{(k+1)} + \gamma(\mathcal{B}_2 z^{(k)} - \mathcal{B}_2 x^{(k+1)})\right)$;
  $\bar{x}^{(k+1)} \leftarrow x^{(k+1)} + \theta\left(x^{k+1} - x^{k}\right)$;
$x \leftarrow \bar{x}^{(k+1)}$;

The step-size γ has to be chosen in $[\epsilon; \chi - \epsilon]$ where χ is given by

$$\chi = \frac{4\beta}{1 + \sqrt{1 + 16\beta^2 K_L^2}}.$$

- Appendix C -

Detailed computations for SDP

§ C.1 SDP weird cases

For $n = 2$, we set

$$\mathbf{X} = \begin{bmatrix} x_1 & x_2 \\ x_2 & x_3 \end{bmatrix}$$

For $n = 3$, we set

$$\mathbf{X} = \begin{bmatrix} x_1 & x_2 & x_3 \\ x_2 & x_4 & x_5 \\ x_3 & x_5 & x_6 \end{bmatrix}$$

C.1.1 Case 1: Feasible and bounded primal without solution

We set the following SDP problem of dimensions $n = 2$, $m = 1$

$$\mathbf{A}_1 = \begin{bmatrix} 0 & -1 \\ -1 & 2 \end{bmatrix}, \ \mathbf{C} = \begin{bmatrix} 2 & -1 \\ -1 & 0 \end{bmatrix}, \ b = 2 \, .$$

The semi-definite constraint in (D) gives

$$\begin{aligned} \mathbf{C} - y\mathbf{A}_1 \in \mathbb{S}_+^2 \quad &\Longrightarrow \quad \mathbf{X} = \begin{bmatrix} 2 & y-1 \\ y-1 & -2y \end{bmatrix} \in \mathbb{S}_+^2 \\ &\Longrightarrow \quad -4y - (y-1)^2 \geq 0 \\ &\Longrightarrow \quad (y+1)^2 \leq 0 \\ &\Longrightarrow \quad y_* = -1 \end{aligned}$$

Hence the optimal value of the dual objective function is -2.

On the other hand, the affine condition in (P) gives

$$\langle \mathbf{A}_1 \mid \mathbf{X} \rangle_{\mathbb{S}^n} = b \quad \Longrightarrow \quad x_3 = x_2 + 1$$

The value -2 is obtained in the primal objective function if and only if $x_2 = x_1 + 1$. The semi-definiteness of X implies that its determinant is positive and thus gives

$$(x_2 - 1)(x_2 + 1) - x_2^2 \geq 0 \quad \Longrightarrow \quad -1 \geq 0 \, .$$

This is impossible. We conclude that the optimal value -2 is never reached.

C.1.2 Case 2: Feasible dual without solution

We set the following SDP problem of dimensions $n = 2$, $m = 2$

$$\mathbf{A}_1 = \begin{bmatrix} -1 & 0 \\ 0 & 0 \end{bmatrix}, \ \mathbf{A}_2 = \begin{bmatrix} 0 & 0 \\ 0 & -1 \end{bmatrix}, \ \mathbf{C} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \ \mathbf{b} = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$$

The semi-definite constraint in (D) gives

$$\begin{aligned} \mathbf{C} - y_1\mathbf{A}_1 - y_2\mathbf{A}_2 \in \mathbb{S}_+^2 \quad &\implies \quad \mathbf{X} = \begin{bmatrix} y_1 & 1 \\ 1 & y_2 \end{bmatrix} \in \mathbb{S}_+^2 \\ &\implies \quad y_1 + y_2 \geq 0 \quad \text{and} \quad y_1 y_2 \geq 1 \\ &\implies \quad y_1 \geq 0 \end{aligned}$$

The dual problem maximises $-y_1$, thus the optimal value of the dual objective function is 0. However, there is no feasible point satisfying $y_1 = 0$.

Looking at the affine constraints of (P), we obtain

$$\langle \mathbf{A}_1 \mid \mathbf{X} \rangle_{\mathbb{S}^n} = -1 \quad \implies \quad x_1 = 1 \tag{C.1}$$
$$\langle \mathbf{A}_2 \mid \mathbf{X} \rangle_{\mathbb{S}^n} = 0 \quad \implies \quad x_3 = 0 \tag{C.2}$$

The semi-definiteness of $\mathbf{X}$ thus gives $x_2 = 0$ and we derive that the primal objective function optimal value is equal to 0. There is no duality gap.

C.1.3 Case 3: Feasible dual and infeasible primal

We set the following SDP problem of dimensions $n = 2$, $m = 2$

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \ \mathbf{A}_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \ \mathbf{C} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \ \mathbf{b} = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$$

The semi-definite constraint in (D) gives

$$\begin{aligned} \mathbf{C} - y_1\mathbf{A}_1 - y_2\mathbf{A}_2 \in \mathbb{S}_+^2 \quad &\implies \quad \mathbf{X} = \begin{bmatrix} -y_1 & -y_2 \\ -y_2 & 0 \end{bmatrix} \in \mathbb{S}_+^2 \\ &\implies \quad y_1 \leq 0 \quad \text{and} \quad y_2^2 \leq 0 \\ &\implies \quad y_1 \leq 0 \quad \text{and} \quad y_2 = 0 \end{aligned}$$

Therefore the optimal value of the dual objective function is 0.

The affine constraints of (P) gives two conditions on X

$$\langle \mathbf{A}_1 \mid \mathbf{X} \rangle_{\mathbb{S}^n} = 0 \quad \implies \quad x_1 = 0 \tag{C.3}$$
$$\langle \mathbf{A}_2 \mid \mathbf{X} \rangle_{\mathbb{S}^n} = 2 \quad \implies \quad x_2 = 1 \tag{C.4}$$

The semi-definiteness of $\mathbf{X}$ thus gives $-1 \geq 0$, which is impossible. The primal problem is infeasible.

C.1.4 Case 4: Non-zero duality gap at optimality

We set the following SDP problem of dimensions $n = 3$, $m = 1$

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \ \mathbf{A}_2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 2 \end{bmatrix}, \ \mathbf{C} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \ \mathbf{b} = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$$

The semi-definite constraint in (D) gives

$$\begin{aligned}\mathbf{C} - y_1\mathbf{A}_1 - y_2\mathbf{A}_2 \in \mathbb{S}_+^2 \quad &\Longrightarrow \quad \mathbf{X} = \begin{bmatrix} -y_1 & -y_2 & 0 \\ -y_2 & 0 & 0 \\ 0 & 0 & 1-2y_2 \end{bmatrix} \in \mathbb{S}_+^2 \\ &\Longrightarrow \quad y_1 \leq 0 \quad \text{and} \quad y_2^2 \leq 0 \\ &\Longrightarrow \quad y_1 \leq 0 \quad \text{and} \quad y_2 = 0\end{aligned}$$

Therefore the optimal value of the dual objective function is 0.

Looking at the affine constraints of (P), we obtain

$$\langle \mathbf{A}_1 \mid \mathbf{X} \rangle_{\mathbb{S}^n} = 0 \quad \Longrightarrow \quad x_1 = 0 \tag{C.5}$$

$$\langle \mathbf{A}_2 \mid \mathbf{X} \rangle_{\mathbb{S}^n} = 2 \quad \Longrightarrow \quad x_2 + x_6 = 1 \tag{C.6}$$

The semi-definiteness of $\mathbf{X}$ thus gives $x_6 = 1$, which is also the optimal value of the primal objective function.

There is a duality gap of 1.

§ C.2 Computations of subproblems in proximal methods

C.2.1 Resolvents for Douglas-Rachford algorithm on the augmented Lagrangian formulation

We compute here the two resolvents $J_{\gamma\mathcal{B}}$ and $J_{\gamma\mathcal{A}}$ for the algorithm **DougRachD+AL** in Section 6.3.1.

- Computation of $J_{\gamma\mathcal{B}}$:

$$\begin{pmatrix}\mathbf{y}\\ \mathbf{S}\\ \mathbf{X}\end{pmatrix} = J_{\gamma\mathcal{B}}\begin{pmatrix}\mathbf{w}\\ \mathbf{Z}\\ \mathbf{V}\end{pmatrix} \Longleftrightarrow \gamma^{-1}\begin{pmatrix}\mathbf{w}-\mathbf{y}\\ \mathbf{Z}-\mathbf{S}\\ \mathbf{V}-\mathbf{X}\end{pmatrix} = \begin{pmatrix}\rho AA^*(\mathbf{y}) + \rho A(\mathbf{S}) + A(\mathbf{X})\\ \rho A^*(\mathbf{y}) + \rho\mathbf{S} + \mathbf{X}\\ -A^*(\mathbf{y}) - \mathbf{S}\end{pmatrix}$$

$$\Longleftrightarrow \begin{cases}\mathbf{S} = -A^*(\mathbf{y}) + \gamma^{-1}(\mathbf{X}-\mathbf{V})\\ \mathbf{Z} = (1+\gamma\rho)\mathbf{S} + \gamma\mathbf{X} + \gamma\rho A^*(\mathbf{y})\\ \mathbf{w} = (\mathrm{Id} + \gamma\rho AA^*)\mathbf{y} + \gamma\rho A(\mathbf{S}) + \gamma A(\mathbf{X})\end{cases}$$

Let us express $A(\mathbf{X})$

$$\begin{aligned}\mathbf{Z} &= (1+\gamma\rho)\mathbf{S} + \gamma\mathbf{X} + \gamma\rho A^*\mathbf{y}\\ \mathbf{Z} &= (1+\gamma\rho)(-A^*(\mathbf{y}) + \gamma^{-1}(\mathbf{X}-\mathbf{V})) + \gamma\mathbf{X} + \gamma\rho A^*(\mathbf{y})\\ \mathbf{Z} &= -A^*(\mathbf{y}) + (\gamma + \gamma^{-1} + \rho)\mathbf{X} - (\gamma^{-1}+\rho)\mathbf{V}\\ \mathbf{X} &= \frac{1}{\gamma+\gamma^{-1}+\rho}\left(\mathbf{Z} + A^*(\mathbf{y}) + (\gamma^{-1}+\rho)\mathbf{V}\right)\\ A(\mathbf{X}) &= \frac{1}{\gamma+\gamma^{-1}+\rho}\left(A(\mathbf{Z}) + AA^*(\mathbf{y}) + (\gamma^{-1}+\rho)A(\mathbf{V})\right)\end{aligned}$$

We can now express $\mathbf{w}$

$$\begin{aligned}\mathbf{w} &= (\mathrm{Id} + \gamma\rho AA^*)\mathbf{y} + \gamma\rho A(\mathbf{S}) + \gamma A(\mathbf{X})\\ &= (\mathrm{Id} + \gamma\rho AA^*)\mathbf{y} + \gamma\rho A\left(-A^*(\mathbf{y}) + \gamma^{-1}(\mathbf{X}-\mathbf{V})\right) + \gamma A(\mathbf{X})\\ &= \mathbf{y} + (\gamma+\rho)A(\mathbf{X}) - \rho^{-1}A(\mathbf{V})\\ &= \mathbf{y} + \frac{\gamma+\rho}{\gamma+\gamma^{-1}+\rho}\left(A(\mathbf{Z}) + AA^*(\mathbf{y}) + (\gamma^{-1}+\rho)A(\mathbf{V})\right) - \rho A(\mathbf{V})\end{aligned}$$

We set κ such that

$$\kappa = \frac{\gamma + \rho}{\gamma + \gamma^{-1} + \rho}$$

Hence,

$$\mathbf{w} = (\mathrm{Id} + \kappa AA^*)\mathbf{y} + \kappa A(\mathbf{Z}) + (\kappa(\gamma + \rho) - \rho)\, A(\mathbf{V})$$

But,

$$\begin{aligned}\kappa(\gamma + \mu^{-1}) - \rho &= \frac{(\gamma + \rho)(\gamma + \rho)}{\gamma + \gamma^{-1} + \rho} - \rho \\ &= \frac{(\gamma + \rho)(\gamma + \rho) - \rho(\gamma + \gamma^{-1} + \rho)}{\gamma + \gamma^{-1} + \rho} \\ &= \frac{1}{\gamma + \gamma^{-1} + \rho}\end{aligned}$$

Thus,

$$\mathbf{y} = (\mathrm{Id} + \kappa AA^*)^{-1}\left(\mathbf{w} - \kappa A(\mathbf{Z}) - \frac{1}{\gamma + \gamma^{-1} + \rho} A(\mathbf{V})\right)$$

Finally, we get

$$\begin{cases} \mathbf{y} = (\mathrm{Id} + \kappa AA^*)^{-1}\left(\mathbf{w} - \kappa A(\mathbf{Z}) - \dfrac{1}{\gamma + \gamma^{-1} + \rho} A(\mathbf{V})\right) \\ \mathbf{X} = \dfrac{1}{\gamma + \gamma^{-1} + \rho}\left(\mathbf{Z} + A^*(\mathbf{y}) + (\gamma^{-1} + \rho)\mathbf{V}\right) \\ \mathbf{S} = -A^*\mathbf{y} + \gamma^{-1}(\mathbf{X} - \mathbf{V}) \end{cases}$$

- Computation of $J_{\gamma\mathcal{A}}$:

$$\begin{pmatrix} \mathbf{w} \\ \mathbf{Z} \\ \mathbf{V} \end{pmatrix} = J_{\gamma\mathcal{A}} \begin{pmatrix} \mathbf{y} \\ \mathbf{S} \\ \mathbf{X} \end{pmatrix} \Leftrightarrow \gamma^{-1} \begin{pmatrix} \mathbf{y} - \mathbf{w} \\ \mathbf{S} - \mathbf{Z} \\ \mathbf{X} - \mathbf{V} \end{pmatrix} \in \begin{pmatrix} -\mathbf{b} - \rho A(\mathbf{C}) \\ \mathcal{N}_{\mathbb{S}^n_+}(\mathbf{Z}) - \rho\mathbf{C} \\ \mathbf{C} \end{pmatrix}$$

$$\Leftrightarrow \begin{cases} \mathbf{w} = \mathbf{y} + \gamma(\mathbf{b} + \rho A(\mathbf{C})) \\ \mathbf{Z} = \Pi_{\mathbb{S}^n_+}(\mathbf{S} + \gamma\rho\mathbf{C}) \\ \mathbf{V} = \mathbf{X} - \gamma\mathbf{C} \end{cases}$$

C.2.2 Subproblems in ADMM

W solve here the two subproblems to get the expression of $\mathbf{y}^{(k+1)}$ and $\mathbf{Z}^{(k+1)}$ for ADMM in Section 6.3.1.3. We start with $\mathbf{y}^{(k+1)}$

$$\begin{aligned}\mathbf{y}^{(k+1)} &= \underset{\mathbf{y}\in\mathbb{R}^m}{\arg\min}\, \mathcal{L}_\rho(\mathbf{y}, \mathbf{Z}^{(k)}, \mathbf{X}^{(k)}) \\ &= \underset{\mathbf{y}\in\mathbb{R}^m}{\arg\min} \langle -\mathbf{b} \mid \mathbf{y}\rangle_{\mathbb{R}^m} + \frac{\rho}{2}\left\| A^*(\mathbf{y}) - \mathbf{Z}^{(k)} + \rho^{-1}\mathbf{X}^{(k)}\right\|^2 \\ \Longrightarrow \mathbf{0} &= -\mathbf{b} + \rho A(A^*(\mathbf{y}^{(k+1)}) - \mathbf{Z}^{(k)} + \rho^{-1}\mathbf{X}^{(k)}) \\ \Longrightarrow \mathbf{y}^{(k+1)} &= (AA^*)^{-1}(\rho^{-1}\mathbf{b} + A(\mathbf{Z}^{(k)} - \rho^{-1}\mathbf{X}^{(k)}))\,,\end{aligned}$$

and then $\mathbf{Z}^{(k+1)}$

$$\begin{aligned}
\mathbf{Z}^{(k+1)} &= \operatorname*{arg\,min}_{\mathbf{Z}\in\mathbb{S}^n} \mathcal{L}_\rho(\mathbf{y}^{(k+1)}, \mathbf{Z}, \mathbf{X}^{(k)}) \\
&= \operatorname*{arg\,min}_{\mathbf{Z}\in\mathbb{S}^n} \iota_{\mathbb{S}^n_+} \circ \tau_c(\mathbf{Z}) + \frac{\rho}{2}\left\| A^*(\mathbf{y}^{(k+1)}) - \mathbf{Z} + \rho^{-1}\mathbf{X}^{(k)} \right\|^2 \\
\implies 0 &\in -\mathcal{N}_{\mathbb{S}^n_+}(\mathbf{C} - \mathbf{Z}^{(k+1)}) - \rho(A^*(\mathbf{y}^{(k+1)}) - \mathbf{Z}^{(k+1)} + \rho^{-1}\mathbf{X}^{(k)}) \\
\implies -A^*(\mathbf{y}^{(k+1)}) + \mathbf{Z}^{(k+1)} - \rho^{-1}\mathbf{X}^{(k)} &\in \mathcal{N}_{\mathbb{S}^n_+}(\mathbf{C} - \mathbf{Z}^{(k+1)}) \\
\implies \mathbf{C} - A^*(\mathbf{y}^{(k+1)}) - \rho^{-1}\mathbf{X}^{(k)} - (\mathbf{C} - \mathbf{Z}^{(k+1)}) &\in \mathcal{N}_{\mathbb{S}^n_+}(\mathbf{C} - \mathbf{Z}^{(k+1)}) \\
\implies \mathbf{C} - \mathbf{Z}^{(k+1)} &= \Pi_{\mathbb{S}^n_+}(\mathbf{C} - A^*(\mathbf{y}^{(k+1)}) - \rho^{-1}\mathbf{X}^{(k)}) \\
\implies \mathbf{Z}^{(k+1)} &= \mathbf{C} - \Pi_{\mathbb{S}^n_+}(\mathbf{C} - A^*(\mathbf{y}^{(k+1)}) - \rho^{-1}\mathbf{X}^{(k)})\,.
\end{aligned}$$

C.2.3 Proximal operator for Douglas-Rachford algorithm on the augmented quadratic objective

We compute here the proximal operator used in the Douglas-Rachford algorithm in Section 6.3.2

$$\begin{aligned}
\begin{pmatrix}\mathbf{w}\\ \mathbf{Z}\end{pmatrix} = \operatorname{prox}_{\gamma f}\begin{pmatrix}\mathbf{y}\\ \mathbf{S}\end{pmatrix} &\iff \gamma^{-1}\begin{pmatrix}\mathbf{y}-\mathbf{w}\\ \mathbf{S}-\mathbf{Z}\end{pmatrix} \in \begin{pmatrix}-(\langle -\mathbf{b} \mid \mathbf{w}\rangle_{\mathbb{R}^m} + \|\mathbf{b}\| K)\mathbf{b}\\ N_{\mathbb{S}^n_+}(\mathbf{Z})\end{pmatrix} \\
&\iff \begin{cases}\mathbf{w} = \mathbf{y} - \gamma\langle \mathbf{b} \mid \mathbf{w}\rangle_{\mathbb{R}^m}\mathbf{b} + \gamma\|\mathbf{b}\| K\mathbf{b} \\ \mathbf{S} - \mathbf{Z} \in N_{\mathbb{S}^n_+}(\mathbf{Z})\end{cases} \\
&\iff \begin{cases}\mathbf{w} = \mathbf{y} - \gamma\langle \mathbf{b} \mid \mathbf{w}\rangle_{\mathbb{R}^m}\mathbf{b} + \gamma\|\mathbf{b}\| K\mathbf{b} \\ \mathbf{Z} = \Pi_{\mathbb{S}^n_+}(\mathbf{S})\end{cases}
\end{aligned}$$

We first compute $\langle \mathbf{w} \mid \mathbf{b}\rangle_{\mathbb{R}^m}$:

$$\begin{aligned}
\mathbf{w} &= \mathbf{y} - \gamma\langle \mathbf{b} \mid \mathbf{w}\rangle_{\mathbb{R}^m}\mathbf{b} + \gamma\|\mathbf{b}\| K\mathbf{b} \\
\langle \mathbf{w} \mid \mathbf{b}\rangle_{\mathbb{R}^m} &= \langle \mathbf{y} \mid \mathbf{b}\rangle_{\mathbb{R}^m} - \gamma\langle \mathbf{b} \mid \mathbf{w}\rangle_{\mathbb{R}^m}\|\mathbf{b}\|^2 + \gamma\|\mathbf{b}\|^3 K \\
\langle \mathbf{w} \mid \mathbf{b}\rangle_{\mathbb{R}^m} &= \frac{\langle \mathbf{y} \mid \mathbf{b}\rangle_{\mathbb{R}^m} + \gamma\|\mathbf{b}\|^3 K}{(1+\gamma\|\mathbf{b}\|^2)}\,.
\end{aligned}$$

We then express $\mathbf{w}$ as a function of $\mathbf{y}$:

$$\begin{aligned}
\mathbf{w} &= \mathbf{y} - \gamma\langle \mathbf{b} \mid \mathbf{w}\rangle_{\mathbb{R}^m}\mathbf{b} + \gamma\|\mathbf{b}\| K\mathbf{b} \\
&= \mathbf{y} - \gamma\frac{\langle \mathbf{y} \mid \mathbf{b}\rangle_{\mathbb{R}^m} + \gamma\|\mathbf{b}\|^3 K}{(1+\gamma\|\mathbf{b}\|^2)} + \gamma\|b\| K\mathbf{b} \\
&= \mathbf{y} - \frac{\gamma(\langle \mathbf{y} \mid \mathbf{b}\rangle_{\mathbb{R}^m} - \|\mathbf{b}\| K)\mathbf{b}}{1+\gamma\|\mathbf{b}\|^2}\,.
\end{aligned}$$

Finally, we obtain

$$\begin{pmatrix}\mathbf{w}\\ \mathbf{Z}\end{pmatrix} = \operatorname{prox}_{\gamma f}\begin{pmatrix}\mathbf{y}\\ \mathbf{S}\end{pmatrix} \iff \begin{cases}\mathbf{w} = \mathbf{y} - \dfrac{\gamma(\langle \mathbf{y} \mid \mathbf{b}\rangle_{\mathbb{R}^m} - \|\mathbf{b}\| K)\mathbf{b}}{1+\gamma\|\mathbf{b}\|^2} \\ \mathbf{Z} = \Pi_{\mathbb{S}^n_+}(\mathbf{S})\end{cases}\,.$$

- Appendix D -

Extraction method for rational optimization

This appendix shows how to recover the support of an R-atomic measure μ from its truncated moment matrix $\mathbf{M}_k$ containing all its moments up to degree $2k$. This method is used to extract the minimizers $(\mathbf{x}^*(r))_{r\in[\![1,R]\!]}$ of polynomial and rational problems from the solutions of the SDP relaxations in Lasserre's hierarchy, i.e. the truncated vector of moments.

§ D.1 Link between minimizers and moment matrix

The moment matrix $\mathbf{M}_k$ is indexed by the pair of multi-indices $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ in $\mathbb{N}^T_k \times \mathbb{N}^T_k$. To each multi-index $\boldsymbol{\alpha}$ corresponds a monic monomial $\mathbf{x}^{\boldsymbol{\alpha}}$ in $\mathbb{R}[\mathbf{x}]_k$ and the moment matrix can thus be seen as indexed by monomials. We define $\mathcal{M}^n_k = \{\mathbf{x}^{\boldsymbol{\alpha}} \mid \boldsymbol{\alpha} \in \mathbb{N}^n_k\}$ the set of all monomials with degree less than k. It is a basis of $\mathbb{R}[\mathbf{x}]_k$, the set of polynomials in $\mathbb{R}[\mathbf{x}]$ whose degree is lower or equal to k. $\mathbf{M}_k$ can thereby be interpreted as the matrix expressed in $\mathcal{M}^n_k$ of a linear application $\phi : \mathbb{R}[\mathbf{x}]_k \longrightarrow \mathbb{R}[\mathbf{x}]_k$. We write $\operatorname{Ker}\phi$ its kernel,

$$\operatorname{Ker}\phi = \{p \in \mathbb{R}[\mathbf{x}]_k \mid \phi(p) = 0\}\,.$$

According to [Lau08, Theorem 5.29], if the sufficient rank condition on $\mathbf{M}_k$ and $\mathbf{M}_{k-1}$ is satisfied, the minimizers $(\mathbf{x}^*(r))_{r\in[\![1,R]\!]}$ are the common zeros of the polynomials in $\operatorname{Ker}\phi$, that is

$$(\mathbf{x}^*(r))_{r\in[\![1,R]\!]} = \{\mathbf{w} \in \mathbb{R}^n \mid (\forall p \in \operatorname{Ker}\phi) \quad p(\mathbf{w}) = 0\}\,.$$

The extraction of the minimizers is therefore equivalent to solving a polynomial system. Note that, if $\mathbf{x}^*$ is a solution of the latter polynomial system, then for any polynomials p_1, p_2 in $\operatorname{Ker}\phi$ and q_1, q_2 in $\mathbb{R}[\mathbf{x}]$, $\mathbf{x}^*$ is a zero of $p_1q_1 + p_2q_2$. Hence, we define $\mathcal{I}$, the ideal generated by $\operatorname{Ker}\phi$:

$$\mathcal{I} = \left\{\sum_{i=1}^{s} p_i q_i \mid s \in \mathbb{N}^*, q_i \in \operatorname{Ker}\phi, p_i \in \mathbb{R}[\mathbf{x}]\right\}$$

The set of common zeros (or roots) of polynomials in $\mathcal{I}$ is the same as the set of common zeros of polynomials in $\operatorname{Ker}\phi$ [CLO05].

§ D.2 Eigenvalue method to solve polynomial systems

The eigenvalue method transforms the original polynomial system into a linear algebra problem and has been described in several places (see [CLO05]). It works in the quotient

space $\mathbb{R}[\mathbf{x}]/\mathcal{I}$ of all the possible remainders in the division of polynomials in $\mathbb{R}[\mathbf{x}]$ by elements of $\mathcal{I}$. In other words, two polynomials whose difference belongs to $\mathcal{I}$ are identical in $\mathbb{R}[\mathbf{x}]/\mathcal{I}$.

The method requires two key elements: a monomial basis $\mathcal{B}$ of $\mathbb{R}[\mathbf{x}]/\mathcal{I}$ and the corresponding multiplication matrices $(\mathbf{N}_i)_{i\in[\![1,n]\!]}$. The latter is the representation in the basis $\mathcal{B}$ of the linear application m_{x_i} defined for each coordinate x_i as

$$\begin{array}{llccc} (\forall i \in [\![1,n]\!]) & m_{x_i}: & \mathbb{R}[\mathbf{x}]/\mathcal{I} & \longrightarrow & \mathbb{R}[\mathbf{x}]/\mathcal{I} \\ & & p & \longmapsto & x_i p \end{array} .$$

By [Lau08, Theorem 5.29], the dimension of $\mathbb{R}[\mathbf{x}]/\mathcal{I}$ is given by the rank of ϕ. Moreover, any set indexing a maximum linearly independent set of columns of $\mathbf{M}_k$ is a basis of $\mathbb{R}[\mathbf{x}]/\mathcal{I}$. In our context, a basis $\mathcal{B}$ and its associated multiplication matrices can be easily computed simultaneously using the reduced row echelon form $\mathbf{U}$ of $\mathbf{M}_k$ [HL05].

Indeed, the columns of $\mathbf{M}_k$ corresponding to the pivot elements in $\mathbf{U}$ are linearly independent and span the range of $\mathbf{M}_k$. Therefore, the set of monomials $\{\mathbf{x}^{\boldsymbol{\beta}_1}, \ldots, \mathbf{x}^{\boldsymbol{\beta}_R}\}$ corresponding to those columns is the sought monomial basis $\mathcal{B}$ of $\mathbb{R}[\mathbf{x}]/\mathcal{I}$. Any other monomial $\mathbf{x}^{\boldsymbol{\alpha}}$ in $\mathcal{M}_k^n \setminus \mathcal{B}$ can be expressed as

$$\mathbf{x}^{\boldsymbol{\alpha}} = \underbrace{\sum_{r=1}^{R} b_r(\mathbf{x}^{\boldsymbol{\alpha}})\mathbf{x}^{\boldsymbol{\beta}_r}}_{\in \mathrm{Span}(\mathcal{B})} + \underbrace{\xi}_{\in \mathrm{Ker}\,\phi} \tag{D.1}$$

where the coefficient $b_r(\mathbf{x}^{\boldsymbol{\alpha}})$ is the element of $\mathbf{U}$ in the column indexed by $\boldsymbol{\alpha}$ and in the row indexed by $\boldsymbol{\beta}_r$

$$(\forall r \in [\![1,R]\!])(\forall \boldsymbol{\alpha} \in \mathbb{N}_k^n) \quad b_r(\mathbf{x}^{\boldsymbol{\alpha}}) = U_{\boldsymbol{\beta}_r,\boldsymbol{\alpha}} .$$

Equation (D.1) shows that all monomials of $\mathcal{M}_k^n$ can be expressed in $\mathbb{R}[\mathbf{x}]/\mathcal{I}$ as a linear combination of the monomials in the basis $\mathcal{B}$. In particular, reading in $\mathbf{U}$ the coefficients $\left(b_s(x_i\mathbf{x}^{\boldsymbol{\beta}_r})\right)_{s\in[\![1,R]\!]}$ of the monomial $x_i\mathbf{x}^{\boldsymbol{\beta}_r}$ for $r \in [\![1,R]\!]$ gives the expression of the multiplication matrix $\mathbf{N}_i$

$$\mathbf{N}_i = \begin{bmatrix} b_1(x_i\mathbf{x}^{\boldsymbol{\beta}_1}) & b_1(x_i\mathbf{x}^{\boldsymbol{\beta}_2}) & \ldots & b_1(x_i\mathbf{x}^{\boldsymbol{\beta}_R}) \\ b_2(x_i\mathbf{x}^{\boldsymbol{\beta}_1}) & b_2(x_i\mathbf{x}^{\boldsymbol{\beta}_2}) & \ldots & b_2(x_i\mathbf{x}^{\boldsymbol{\beta}_R}) \\ \vdots & \vdots & \ddots & \vdots \\ b_R(x_i\mathbf{x}^{\boldsymbol{\beta}_1}) & b_R(x_i\mathbf{x}^{\boldsymbol{\beta}_2}) & \ldots & b_R(x_i\mathbf{x}^{\boldsymbol{\beta}_R}) \end{bmatrix} .$$

§ D.3 Multiplication matrices and minimizers

Stickelberger eigenvalue theorem [CLO05, Theorem 4.5] states that, for any polynomial x_i with $i \in [\![1,n]\!]$, an eigenvalue of the operator m_{x_i} corresponds to a value of x_i at a point $\mathbf{w}$ of $(\mathbf{x}^*(r))_{r\in[\![1,R]\!]}$. That is to say, the eigenvalues of $\mathbf{N}_i$ correspond to the i-th coordinates of $(\mathbf{x}^*(r))_{r\in[\![1,R]\!]}$. Note that since multiplication matrices all commute pairwise, they preserve each other's eigenspaces.

A numerically stable way to compute those eigenvalues based on Schur factorization has been proposed in [CGT97] and is summarised below. First, take a random linear convex combinations $\mathbf{N}_h$ of $(\mathbf{N}_i)_{i\in[\![1,n]\!]}$

$$\mathbf{N}_h = \sum_{i=1}^{n} a_i \mathbf{N}_i ,$$

where $(a_i)_{i\in[\![1,n]\!]}$ are real numbers chosen randomly and summing up to one. Now, a key point is yet to use the radicality of $\mathcal{I}$, i.e. any common roots of polynomials in $\mathcal{I}$ has single multiplicity. Indeed, according to [CLO05, Proposition 4.7], the left eigenspaces of $\mathbf{N}_h$ are then all one-dimensional. In our case, the radicality of $\mathcal{I}$ is implicitly given by [Lau08, Theorem 5.29]. Indeed, it gives that $\dim \mathbb{R}[\mathbf{x}]/\mathcal{I} = R$. This condition is yet equivalent to the radicality of $\mathcal{I}$ [CLO05, Theorem 2.10].

Following [CGT97], the left eigenspaces of $\mathbf{N}_h$ can be found by computing an ordered Schur decomposition of $\mathbf{N}_h$: $\mathbf{N}_h = \mathbf{Q}\mathbf{T}\mathbf{Q}^\top$ where $\mathbf{Q}$ is an orthogonal matrix and $\mathbf{T}$ is upper triangular. The coordinates of the points $(\mathbf{x}^*(r))_{r\in[\![1,R]\!]}$ are finally given by

$$(\forall i \in [\![1,n]\!])(\forall r \in [\![1,R]\!]) \quad x_i^*(r) = \mathbf{q}_r^\top \mathbf{N}_i \mathbf{q}_r \,,$$

where $\mathbf{q}_r$ is the r-th column of the matrix $\mathbf{Q}$.

Remark that, without the radicality of $\mathcal{I}$, the dimension of the left eigenspaces could be different from the multiplicities of the roots and consequently the dimension of $\mathbb{R}[\mathbf{x}]/\mathcal{I}$ could be greater than R.

List of Figures

List of Tables

List of Algorithms

Bibliography

[AB08] Miguel F. Anjos and Samuel Burer. On handling free variables in interior-point methods for conic linear optimization. *SIAM J. Optim.*, 18(4):1310–1325, January 2008. 8, 63

[ADK11] Evrim Acar, Daniel M. Dunlavy, and Tamara G. Kolda. A scalable optimization approach for fitting canonical tensor decompositions. *J. Chemometrics*, 25(2):67–86, January 2011. 87

[AFS13] Marco Artina, Massimo Fornasier, and Francesco Solombrino. Linearly constrained nonsmooth and nonconvex minimization. *SIAM J. Optim.*, 23(3):1904–1937, January 2013. 26, 28

[AGH+14] Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M. Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models. *J. Mach. Learn. Res.*, 15(1):2773–2832, January 2014. 88

[AH17] Amir Ahmadi and Georgina Hall. Sum of squares basis pursuit with linear and second order cone programming. In *Algebraic and Geometric Methods in Discrete Mathematics*, pages 27–53. American Mathematical Society, 2017. 14

[AH18] M. Salman Asif and Chinmay Hegde. Phase retrieval for signals in union of subspaces. In *Proc. IEEE Global Conf. Signal Information Process.*, pages 356–359. IEEE, November 2018. 51

[AH19] Amir Ali Ahmadi and Georgina Hall. On the construction of converging hierarchies for polynomial optimization based on certificates of global positivity. *Math. Oper. Res.*, August 2019. 14

[AL11] Miguel F. Anjos and Frauke Liers. Global approaches for facility layout and VLSI floorplanning. In *Handbook on Semidefinite, Conic and Polynomial Optimization*, pages 849–877. Springer US, September 2011. 59

[AM19] Amir Ali Ahmadi and Anirudha Majumdar. DSOS and SDSOS optimization: More tractable alternatives to sum of squares and semidefinite optimization. *SIAM J. Appl. Algebr. Geom.*, 3(2):193–230, January 2019. 14

[BAC+15] Hanna Becker, Laurent Albera, Pierre Comon, Remi Gribonval, Fabrice Wendling, and Isabelle Merlet. Brain-source imaging: From sparse to tensor models. *IEEE Signal Process. Mag.*, 32(6):100–112, November 2015. 87

[BAD18] Luis M. Briceño-Arias and Damek Davis. Forward-backward-half forward algorithm for solving monotone inclusions. *SIAM J. Optim.*, 28(4):2839–2871, January 2018. 73, 120

[BAR18] Luis Briceño-Arias and Sergio López Rivera. A projected primal-dual method for solving constrained monotone inclusions. *J. Optim. Theory Appl.*, 180(3):907–924, November 2018. 65

[BBCM13] Alessandra Bernardi, Jérôme Brachat, Pierre Comon, and Bernard Mourrain. General tensor decomposition, moment matrices and applications. *J. Symbolic Computat.*, 52:51–71, May 2013. 88

[BC11] Heinz H. Bauschke and Patrick L. Combettes. Convex analysis and monotone operator theory in hilbert spaces. In *CMS Books in Mathematics*, pages 207–222. Springer New York, 2011. 62, 69, 120

[BCC+20] Carla Bertocchi, Emilie Chouzenoux, Marie-Caroline Corbineau, Jean-Christophe Pesquet, and Marco Prato. Deep unfolding of a proximal interior point method for image restoration. *Inverse Problems*, 36(3):034005, February 2020. 112

[BCMT10] Jérôme Brachat, Pierre Comon, Bernard Mourrain, and Elias Tsigaridas. Symmetric tensor decomposition. *Lin. Algebra Appl.*, 433(11-12):1851–1872, December 2010. 87

[BD08] Thomas Blumensath and Mike E. Davies. Iterative thresholding for sparse approximations. *J. Fourier Anal. Appl.*, 14(5-6):629–654, September 2008. 25

[BD14] Christoph Buchheim and Claudia D'Ambrosio. Box-constrained mixed-integer polynomial optimization using separable underestimators. In *Integer Programming and Combinatorial Optimization*, pages 198–209. Springer International Publishing, 2014. 9

[BGI11] Alessandra Bernardi, Alessandro Gimigliano, and Monica Idà. Computing symmetric rank for symmetric tensors. *J. Symbolic Computat.*, 46(1):34–53, January 2011. 88

[BH11] Patrick Breheny and Jian Huang. Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. *Ann. Appl. Stat.*, 5(1):232–253, March 2011. 25

[BHL15] Florian Bugarin, Didier Henrion, and Jean-Bernard Lasserre. Minimizing the sum of many rational functions. *Math. Program. Comput.*, 8(1):83–111, August 2015. 3, 20, 21

[BK+17] Brett W. Bader, Tamara G. Kolda, et al. Matlab tensor toolbox version 3.0-dev. Available online, October 2017. 100

[Ble06] Grigoriy Blekherman. There are significantly more nonegative polynomials than sums of squares. *Israel J. Math.*, 153(1):355–380, December 2006. 14

[Blu11] Thomas Blumensath. Sampling and reconstructing signals from a union of linear subspaces. *IEEE Trans. Inform. Theory*, 57(7):4660–4671, July 2011. 51

[BM03] Samuel Burer and Renato D.C. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Math. Programm.*, 95(2):329–357, February 2003. 59

[BNCM16] Sébastien Bourguignon, Jordan Ninin, Hervé Carfantan, and Marcel Mongeau. Exact sparse approximation problems via mixed-integer programming: Formulations and computational performance. *IEEE Trans. Signal Process.*, 64(6):1405–1419, March 2016. 25

[BNFR19] Renan Del Buono Brotto, Kenji Nose-Filho, and João Marcos Travassos Romano. Antisparse blind source separation. In *Proc. Signal Processing with Adaptive Sparse Structured Representations (SPARS) workshop*, July 2019. 52

[BT09] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):183–202, January 2009. 68

[BTT96] Aharon Ben-Tal and Marc Teboulle. Hidden convexity in some nonconvex quadratically constrained quadratic programming. *Math. Programm.*, 72(1):51–63, January 1996. 9

[BVB19] Nicolas Boumal, Vladislav Voroninski, and Afonso S. Bandeira. Deterministic guarantees for Burer-Monteiro factorizations of smooth semidefinite programs. *Commun. Pure Appl. Math.*, May 2019. 59, 112

[BY07] Brian Borchers and Joseph G. Young. Implementation of a primal–dual method for SDP on a shared memory parallel architecture. *Comput. Optim. Appl.*, 37(3):355–369, March 2007. 59

[BYZ00] Steven J. Benson, Yinyu Ye, and Xiong Zhang. Solving large-scale sparse semidefinite programs for combinatorial optimization. *SIAM J. Optim.*, 10(2):443–461, January 2000. 59

[CBFAB94] Pierre Charbonnier, Laure Blanc-Feraud, Gilles Aubert, and Michel Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *Proc. Int. Conf. Image Process.*, volume 2, pages 168–172 vol.2. IEEE Comput. Soc. Press, November 1994. 51, 53

[CCPW07] Caroline Chaux, Patrick L Combettes, Jean-Christophe Pesquet, and Valérie R Wajs. A variational formulation for frame-based inverse problems. *Inverse Problems*, 23(4):1495–1518, June 2007. 25

[CD15] Antonin Chambolle and Charles Dossal. On the convergence of the iterates of "FISTA". *J. Optim. Theory Appl.*, 166(3):968–982, May 2015. 68

[CE16] Patrick L. Combettes and Jonathan Eckstein. Asynchronous block-iterative primal-dual decomposition methods for monotone inclusions. *Math. Progr. Ser. B*, 168(1-2):645–672, July 2016. 69

[CF96] Raúl E. Curto and Lawrence A. Fialkow. Solution of the truncated complex moment problem for flat data. *Mem. Amer. Math. Soc.*, 119(568):0–0, 1996. 91, 94

[CF18] Zhuo-Xu Cui and Qibin Fan. A "nonconvex+nonconvex" approach for image restoration with impulse noise removal. *Appl. Math. Model.*, 62:254–271, October 2018. 112

[CGLM08] Pierre Comon, Gene Golub, Lek-Heng Lim, and Bernard Mourrain. Symmetric tensors and symmetric tensor rank. *SIAM J. Matrix Anal. Appl.*, 30(3):1254–1279, January 2008. 87, 88, 94

[CGT97] Robert M. Corless, Patrizia M. Gianni, and Barry M. Trager. A reordered Schur factorization method for zero-dimensional polynomial systems with multiple roots. In *Proceedings of the 1997 International Symposium on Symbolic and Algebraic Computation*, pages 133–140, New York, NY, USA, July 1997. ACM Press. 99, 130, 131

[Chi09] Mung Chiang. Nonconvex optimization for communication networks. In *Advances in Mechanics and Mathematics*, pages 137–196. Springer US, 2009. 9

[CHN04] Raymond Honfu Chan, Chen Hu, and Mila Nikolova. An iterative procedure for removing random-valued impulse noise. *IEEE Signal Process. Lett.*, 11(12):921–924, December 2004. 112

[CHYT17] Chao Wang, Hong-Yi Yu, Yi-Jun Zhu, and Tao Wang. Blind detection for SPAD-based underwater VLC system under P–G mixed noise model. *IEEE Commun. Lett.*, 21(12):2602–2605, December 2017. 43, 49

[CJPT15] Emilie Chouzenoux, Anna Jezierska, Jean-Christophe Pesquet, and Hugues Talbot. A convex approach for image restoration with exact Poisson–Gaussian likelihood. *SIAM J. Imaging Sci.*, 8(4):2662–2682, jan 2015. 43, 44, 46

[CLO05] David A. Cox, John Little, and Donal O'Shea. *Using Algebraic Geometry*. Springer-Verlag, 2005. 98, 99, 129, 130, 131

[Com01] Patrick L. Combettes. Fejér monotonicity in convex optimization. In *Encyclopedia of Optimization*, pages 1016–1024. Springer US, 2001. 70

[Com10] Pierre Comon. *Handbook of Blind Source Separation: Independent Component Analysis and Applications*. Academic Press Inc, 2010. 96, 106

[CP08] Patrick L. Combettes and Jean-Christophe Pesquet. Proximal thresholding algorithm for minimization over orthonormal bases. *SIAM J. Optim.*, 18(4):1351–1376, January 2008. 25

[CP10] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vision*, 40(1):120–145, December 2010. 120

[CP11] Patrick L. Combettes and Jean-Christophe Pesquet. Proximal Splitting Methods in Signal Processing. In Heinz H. Bauschke, Regina S. Burachik, Patrick L. Combettes, Veit Elser, D. Russell Luke, and Henry Wolkowicz, editors, *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pages 185–212. Springer, 2011. 25

[CP15] Marc Castella and Jean-Christophe Pesquet. Optimization of a Geman-McClure like criterion for sparse signal deconvolution. In *Proc. IEEE Int. Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 309–312, Cancun, Mexico, December 2015. IEEE. 25, 28

[CP17] Marc Castella and Jean-Christophe Pesquet. A global optimization approach for rational sparsity promoting criteria. In *Proc. European Signal Processing Conference*, pages 156–160, Kos, Greece, August 2017. IEEE. 28

[CP20] Patrick L. Combettes and Jean-Christophe Pesquet. Fixed point strategies in data science. ArXiv Preprint, arXiv:2008.02260, August 2020. 59

[CPM19] Marc Castella, Jean-Christophe Pesquet, and Arthur Marmin. Rational optimization for nonlinear reconstruction with approximate ℓ_0 penalization. *IEEE Trans. Signal Process.*, 67(6):1407–1417, March 2019. 20, 28, 34, 55

[CWB08] Emmanuel J. Candès, Michael B. Wakin, and Stephen P. Boyd. Enhancing sparsity by reweighted ℓ_1 minimization. *J. Fourier Anal. Appl.*, 14(5-6):877–905, October 2008. 25, 38

[DB08] Germund Dahlquist and Åke Björck. *Numerical Methods in Scientific Computing, Volume I*. Society for Industrial and Applied Mathematics, January 2008. 27, 45

[DD15] Yannick Deville and Leonardo Tomazeli Duarte. An overview of blind source separation methods for linear-quadratic and post-nonlinear mixtures. In *Latent Variable Analysis and Signal Separation*, pages 155–167. Springer International Publishing, 2015. 25

[DG13] Peter J. C. Dickinson and Luuk Gijben. On the computational complexity of membership problems for the completely positive cone and its dual. *Comput. Optim. Appl.*, 57(2):403–415, September 2013. 10

[DIdW17] Mareike Dressler, Sadik Iliman, and Timo de Wolff. A positivstellensatz for sums of nonnegative circuit polynomials. *SIAM J. Appl. Algebr. Geom.*, 1(1):536–555, January 2017. 15

[dK02] Etienne de Klerk. *Aspects of Semidefinite Programming*. Springer US, 2002. 59

[dKRT97] Etienne de Klerk, Cornelis Roos, and Tam'as Terlaky. Initialization in semidefinite programming via a self-dual skew-symmetric embedding. *Oper. Res. Lett.*, 20(5):213–221, June 1997. 64

[DLZF18] Qiaoqiao Ding, Yong Long, Xiaoqun Zhang, and Jeffrey A. Fessler. Statistical image reconstruction using mixed Poisson-Gaussian noise model for X-ray CT. ArXiv Preprint, arXiv:1801.09533, January 2018. 43, 44

[DP18] Gordana Drašković and Frédéric Pascal. New insights into the statistical properties of M-estimators. *IEEE Trans. Signal Process.*, 66(16):4253–4263, August 2018. 51

[DTR+14] Nicolas Dobigeon, Jean-Yves Tourneret, Cédric Richard, José Carlos M. Bermudez, Stephen McLaughlin, and Alfred O. Hero. Nonlinear unmixing of hyperspectral images: Models and algorithms. *IEEE Signal Process. Mag.*, 31(1):82–94, January 2014. 25

[Dum07] Bogdan Alexandru Dumitrescu. *Positive Trigonometric Polynomials and Signal Processing Applications*. Springer Netherlands, 2007. 9, 111

[EB92] Jonathan Eckstein and Dimitri P. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. 55(1-3):293–318, April 1992. 74

[EM09] Yonina C. Eldar and Moshe Mishali. Robust recovery of signals from a structured union of subspaces. *IEEE Trans. Inform. Theory*, 55(11):5302–5316, November 2009. 51

[Fel98] Attila Felinger. *Data analysis and signal processing in chromatography.* Elsevier, 1998. 26

[FL01] Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Am. Stat. Assoc.*, 96(456):1348–1360, December 2001. 25, 26, 28

[FNYF07] Katsuki Fujisawa, Kazuhide Nakata, Makoto Yamashita, and Mituhiro Fukuda. Sdpa project: Solving large-scale semidefinite programs. *J. Oper. Res. Soc. Japan*, 50(4):278–298, 2007. 59

[GDP09] Jérôme Gauthier, Laurent Duval, and Jean-Christophe Pesquet. Optimization of synthesis oversampled complex filter banks. *IEEE Trans. Signal Process.*, 57(10):3827–3843, October 2009. 25

[GNS07] David Grimm, Tim Netzer, and Markus Schweighofer. A note on the representation of positive polynomials with structured sparsity. *Arch. Math.*, 89(5):399–403, September 2007. 14

[GOL98] Laurent El Ghaoui, Francois Oustry, and Hervé Lebret. Robust solutions to uncertain semidefinite programs. *SIAM J. Optim.*, 9(1):33–52, January 1998. 61

[Gre84] Peter J. Green. Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 46(2):149–192, 1984. 43

[GW95] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, November 1995. 59

[Hen13] Didier Henrion. Optimization on linear matrix inequalities for polynomial systems control, September 2013. 18

[HIS16] Chinmay Hedge, Piotr Indyk, and Ludwig Schmidt. Fast recovery from a union of subspaces. In *Advances in Neural Information Processing Systems 29*, pages 4394–4402. Curran Associates, Inc., 2016. 51

[HL05] Didier Henrion and Jean-Bernard Lasserre. Detecting global optimality and extracting solutions in GloptiPoly. In *Positive Polynomials in Control*, volume 312, pages 293–310. Springer Berlin Heidelberg, September 2005. 19, 88, 98, 102, 103, 104, 106, 130

[HL13] Christopher J. Hillar and Lek-Heng Lim. Most tensor problems are NP-hard. *J. ACM*, 60(6):1–39, November 2013. 87, 88

[HLL09] Didier Henrion, Jean-Bernard Lasserre, and Johan Löfberg. GloptiPoly 3: moments, optimization and semidefinite programming. *Optim. Methods Softw.*, 24(4-5):761–779, October 2009. 33, 35, 46, 55, 104, 106

[HM11] Didier Henrion and Jérôme Malick. Projection methods in conic optimization. In *Handbook on Semidefinite, Conic and Polynomial Optimization*, pages 565–600. Springer US, September 2011. 59, 84

[HR00] Christoph Helmberg and Franz Rendl. A spectral bundle method for semidefinite programming. *SIAM J. Optim.*, 10(3):673–696, January 2000. 59

[Hub64] Peter J. Huber. Robust estimation of a location parameter. *Ann. Math. Statist.*, 35(1):73–101, March 1964. 51, 53

[HW92] Uwe Helmke and Robert C. Williamson. Rational parametrizations of neural networks. In *Proc. Ann. Conf. Neur. Inform. Proc. Syst.*, NIPS'92, pages 623–630, San Francisco, CA, USA, 1992. Morgan Kaufmann Publishers Inc. 9

[IdW16] Sadik Iliman and Timo de Wolff. Amoebas, nonnegative polynomials and sums of squares supported on circuits. *Research in the Mathematical Sciences*, 3(1), March 2016. 15

[JdK05] Dorina Jibetean and Etienne de Klerk. Global optimization of rational functions: a semidefinite programming approach. *Math. Programm.*, 106(1):93–109, April 2005. 13

[JGX19] Zhimeng Jiang, Chen Gong, and Zhengyuan Xu. Clipping noise and power allocation for OFDM-based optical wireless communication using photon detection. *IEEE Wireless Communications Letters*, 8(1):237–240, February 2019. 44

[JH15] Cédric Josz and Didier Henrion. Strong duality in Lasserre's hierarchy for polynomial optimization. *Optim. Lett.*, 10(1):3–10, February 2015. 64

[Jos16] Cédric Josz. *Application of polynomial optimization to electricity transmission networks*. PhD thesis, Université Pierre et Marie Curie, July 2016. 111

[JTVW11] Anna Jezierska, Hugues Talbot, Olga Veksler, and Daniel Wesierski. A fast solver for truncated-convex priors: Quantized-convex split moves. In *Lecture Notes in Computer Science*, pages 45–58. Springer Berlin Heidelberg, 2011. 26, 28

[KB09] Tamara G. Kolda and Brett W. Bader. Tensor decompositions and applications. *SIAM Rev.*, 51(3):455–500, August 2009. 87

[KdW19] Adam Kurpisz and Timo de Wolff. New dependencies of hierarchies in polynomial optimization. In *Proc. Int. Symp. Symbolic and Algebraic Computation*. ACM, July 2019. 12

[KKS18] Yuri Kalambet, Yuri Kozmin, and Andrey Samokhin. Comparison of integration rules in the case of very narrow chromatographic peaks. *Chemometr. Intell. Lab. Syst.*, 179:22–30, 2018. 26

[KL15] Abdulrahman Kalbat and Javad Lavaei. A fast distributed algorithm for decomposable semidefinite programs. In *IEEE Conf. Decision Control*. IEEE, December 2015. 112

[KN77] Mark Grigorievich Krein and Adol'f Abramovich Nudel'man. *The Markov Moment Problem And Extremal Problems*. American Mathematical Society, December 1977. 91

[KNK07] Kazuhiro Kobayashi, Kazuhide Nakata, and Masakazu Kojima. A conversion of an SDP having free variables into the standard form SDP. *Comput. Optim. Appl.*, 36(2-3):289–307, February 2007. 63

[KP15] Nikos Komodakis and Jean-Christophe Pesquet. Playing with duality: An overview of recent primal-dual approaches for solving large-scale optimization problems. *IEEE Signal Process. Mag.*, 32(6):31–54, November 2015. 49, 69

[KPV18] Igor Klep, Janez Povh, and Jurij Volčič. Minimizer extraction in polynomial optimization is robust. *SIAM J. Optim.*, 28(4):3177–3207, January 2018. 102

[KS15] Aritra Konar and Nicholas D. Sidiropoulos. Hidden convexity in QCQP with toeplitz-hermitian quadratics. *IEEE Signal Process. Lett.*, 22(10):1623–1627, October 2015. 9

[Las01] Jean-Bernard Lasserre. Global optimization with polynomials and the problem of moments. *SIAM J. Optim.*, 11(3):796–817, January 2001. 13, 19

[Las09] Jean-Bernard Lasserre. *Moments, Positive Polynomials and Their Applications*. Imperial College Press, London, U.K., 2009. 8, 12, 13, 18, 20, 21

[Lau08] Monique Laurent. Sums of squares, moment matrices and optimization over polynomials. In *Emerging Applications of Algebraic Geometry*, pages 157–270. Springer New York, September 2008. 12, 92, 102, 103, 129, 130, 131

[LD08] Yue M. Lu and Minh N. Do. A theory for sampling signals from a union of subspaces. *IEEE Trans. Signal Process.*, 56(6):2334–2345, June 2008. 51, 54

[LHY16] Gongmin Lan, Chenping Hou, and Dongyun Yi. Robust feature selection via simultaneous capped ℓ_2-norm and $\ell_{2,1}$-norm minimization. In *Proc. IEEE Int. Conf. on Big Data Analysis (ICBDA)*. IEEE, March 2016. 51

[LLK18] Sangyoon Lee, Min Seok Lee, and Moon Gi Kang. Poisson-Gaussian noise analysis and estimation for low-dose X-ray images in the NSCT domain. *Sensors*, 18(4):1019, April 2018. 43, 44, 48

[LM79] Pierre-Louis Lions and Bertrand Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM J. Numer. Anal.*, 16(6):964–979, December 1979. 119

[LM18] Jean-Bernard Lasserre and Victor Magron. In SDP relaxations, inaccurate solvers do robust optimization. ArXiv Preprint, arXiv:1811.02879, November 2018. 84

[LMS+10] Zhi-Quan Luo, Wing-Kin Ma, Anthony So, Yinyu Ye, and Shuzhong Zhang. Semidefinite relaxation of quadratic optimization problems. *IEEE Signal Process. Mag.*, 27(3):20–34, May 2010. 59

[LNM06] Zhaosong Lu, Arkadi Nemirovski, and Renato D. C. Monteiro. Large-scale semidefinite programming via a saddle point mirror-prox algorithm. *Math. Progr. Ser. B*, 109(2-3):211–237, November 2006. 59

[LP04] Johan Löfberg and Pablo A. Parrilo. From coefficients to samples: a new approach to SOS optimization. In *IEEE Conf. Decision Control*. IEEE, May 2004. 13

[LSYZ15] Jia Li, Zuowei Shen, Rujie Yin, and Xiaoqun Zhang. A reweighted ℓ_2 method for image restoration with Poisson and mixed Poisson-Gaussian noise. *Inverse Problems and Imaging*, 9(3):875–894, July 2015. 43, 44

[LSZ00] Zhendong Luo, Jos F. Sturm, and Shuzhong Zhang. Conic convex programming and self-dual embedding. *Optim. Methods Softw.*, 14(3):169–218, January 2000. 64

[MA89] Renato D. C. Monteiro and Ilan Adler. Interior path following primal-dual algorithms. part II: Convex quadratic programming. *Math. Programm.*, 44(1-3):43–66, May 1989. 64

[MCP19a] Arthur Marmin, Marc Castella, and Jean-Christophe Pesquet. Detecting the rank of a symmetric tensor. In *Proc. European Signal Processing Conference*, pages 1–5. IEEE, September 2019.

[MCP19b] Arthur Marmin, Marc Castella, and Jean-Christophe Pesquet. How to globally solve non-convex optimization problems involving an approximate ℓ_0 penalization. In *Proc. Int. Conf. Acoust. Speech Signal Process.*, pages 5601–5605. IEEE, May 2019.

[MCP19c] Arthur Marmin, Marc Castella, and Jean-Christophe Pesquet. Sparse signal reconstruction with a sign oracle. In *Proc. Signal Processing with Adaptive Sparse Structured Representations (SPARS) workshop*, July 2019.

[MCP20a] Arthur Marmin, Marc Castella, and Jean-Christophe Pesquet. Globally optimizing owing to tensor decomposition. In *Proc. European Signal Processing Conference*. IEEE, September 2020. to appear.

[MCP20b] Arthur Marmin, Marc Castella, and Jean-Christophe Pesquet. A moment-based approach for guaranteed tensor decomposition. In *Proc. Int. Conf. Acoust. Speech Signal Process.*, pages 3927–3931. IEEE, May 2020.

[MCP20c] Arthur Marmin, Marc Castella, and Jean-Christophe Pesquet. Robust reconstruction with nonconvex subset constraints: A polynomial optimization approach. In *IEEE Int. Workshop Mach. Learn. Signal Process.* IEEE, September 2020.

[MCPD18] Arthur Marmin, Marc Castella, Jean-Christophe Pesquet, and Laurent Duval. Signal reconstruction from sub-sampled and nonlinearly distorted observations. In *Proc. European Signal Processing Conference*, pages 1970–1974. IEEE, September 2018.

[MCPD21] Arthur Marmin, Marc Castella, Jean-Christophe Pesquet, and Laurent Duval. Sparse signal reconstruction for nonlinear models via piecewise rational optimization. *Signal Process.*, 179:107835, February 2021.

[Més98] Csaba Mészáros. On free variables in interior point methods. *Optim. Methods Softw.*, 9(1-3):121–139, 1998. 63

[MJCP20] Arthur Marmin, Anna Jezierska, Marc Castella, and Jean-Christophe Pesquet. Global optimization for recovery of clipped signals corrupted with poisson-gaussian noise. *IEEE Signal Process. Lett.*, 27:970–974, May 2020.

[MJU17] Michael T. McCann, Kyong Hwan Jin, and Michael Unser. Convolutional neural networks for inverse problems in imaging: A review. *IEEE Signal Process. Mag.*, 34(6):85–95, November 2017. 112

[MKL18] Ramtin Madani, Abdulrahman Kalbat, and Javad Lavaei. A low-complexity parallelizable numerical algorithm for sparse semidefinite programming. *IEEE Trans. Control Netw. Syst.*, 5(4):1898–1909, December 2018. 112

[MMYSB19] Ricardo A. Maronna, R. Douglas Martin, Victor J. Yohai, and Matías Salibián-Barrera. *Robust statistics: theory and methods (with R)*. Wiley, Hoboken, NJ, 2019. 51

[MPRW09] Jérôme Malick, Janez Povh, Franz Rendl, and Angelika Wiegele. Regularization methods for semidefinite programming. *SIAM J. Optim.*, 20(1):336–356, January 2009. 59, 60

[MS06] Jérôme Malick and Hristo S. Sendov. Clarke generalized Jacobian of the projection onto the cone of positive semidefinite matrices. *Set-Valued Analysis*, 14(3):273–293, June 2006. 79

[MZCP17] Yosra Marnissi, Yuling Zheng, Emilie Chouzenoux, and Jean-Christophe Pesquet. A variational bayesian approach for image restoration-application to image deblurring with Poisson-Gaussian noise. *IEEE Trans. Comput. Imag.*, 3(4):722–737, dec 2017. 44, 46, 49

[Nie13] Jiawang Nie. Optimality conditions and finite convergence of Lasserre's hierarchy. *Math. Programm.*, 146(1-2):97–121, May 2013. 14, 19

[Nie15] Jiawang Nie. Generating polynomials and symmetric tensor decompositions. 17(2):423–465, November 2015. 88

[Nie17] Jiawang Nie. Low rank symmetric tensor approximations. *SIAM J. Matrix Anal. Appl.*, 38(4):1517–1540, January 2017. 88

[Nik02] Mila Nikolova. Minimizers of cost-functions involving nonsmooth data-fidelity terms. application to the processing of outliers. *SIAM J. Numer. Anal.*, 40(3):965–994, January 2002. 51

[Nik13] Mila Nikolova. Description of the minimizers of least squares regularized with ℓ_0 norm. Uniqueness of the global minimizer. *SIAM J. Imaging Sci.*, 6(2):904–937, January 2013. 25

[NW12] Jiawang Nie and Li Wang. Regularization methods for SDP relaxations in large-scale polynomial optimization. *SIAM J. Optim.*, 22(2):408–428, February 2012. 60

[OCPB16] Brendan O'Donoghue, Eric Chu, Neal Parikh, and Stephen Boyd. Conic optimization via operator splitting and homogeneous self-dual embedding. *J. Optim. Theory Appl.*, 169(3):1042–1068, February 2016. 59, 74, 79

[ODBP15] Peter Ochs, Alexey Dosovitskiy, Thomas Brox, and Thomas Pock. On iteratively reweighted algorithms for nonsmooth nonconvex optimization in computer vision. *SIAM J. Imaging Sci.*, 8(1):331–372, January 2015. 25

[Ose11] Ivan V. Oseledets. Tensor-train decomposition. *SIAM J. Sci. Comput.*, 33(5):2295–2317, January 2011. 87

[OWNB17] Greg Ongie, Rebecca Willett, Robert D. Nowak, and Laura Balzano. Algebraic variety models for high-rank matrix completion. In *Proc. Int. Conf. Mach. Learn.*, volume 70, pages 2691–2700. PMLR, August 2017. 51

[Pap12] Dávid Papp. Optimal designs for rational function regression. *J. Am. Stat. Assoc.*, 107(497):400–411, March 2012. 9

[Par03] Pablo A. Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Math. Programm.*, 96(2):293–320, May 2003. 13

[PDCP14] Mai Quyen Pham, Laurent Duval, Caroline Chaux, and Jean-Christophe Pesquet. A primal-dual proximal algorithm for sparse template-based adaptive filtering: Application to seismic multiple removal. *IEEE Trans. Signal Process.*, 62(16):4256–4269, August 2014. 25

[PL19] Clarice Poon and Jingwei Liang. Trajectory of alternating direction method of multipliers and adaptive acceleration. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Proc. Ann. Conf. Neur. Inform. Proc. Syst.*, pages 7357–7365. Curran Associates, Inc., 2019. 112

[PN15] Andrei Patrascu and Ion Necoara. Random coordinate descent methods for ℓ_0 regularized convex optimization. *IEEE Trans. Automat. Contr.*, 60(7):1811–1824, July 2015. 25

[Pól28] George Pólya. Über positive Darstellung von Polynomen. *Vierteljahresschrift der Naturforschenden Gesellschaft in Zürich*, pages 141–145, 1928. 12

[Pow81] Michael J. D. Powell. *Approximation Theory and Methods*. Cambridge University Press, March 1981. 8

[Put93] Mihai Putinar. Positive polynomials on compact semi-algebraic sets. *Indiana Univ. Math. J.*, 42(3):969–984, 1993. 11

[PW98] Victoria Powers and Thorsten Wörmann. An algorithm for sums of squares of real polynomials. *J. Pure Appl. Algebra*, 127(1):99–104, May 1998. 13

[QCC18] Liqun Qi, Haibin Chen, and Yannan Chen. *Tensor Eigenvalues and Their Applications*. Springer Singapore, 2018. 87, 88

[Ren01] James Renegar. *A Mathematical View of Interior-Point Methods in Convex Optimization*. Society for Industrial and Applied Mathematics, January 2001. 59

[Rez95] Bruce Reznick. Uniform denominators in Hilbert's seventeenth problem. *Mathematische Zeitschrift*, 220(1):75–97, December 1995. 12

[RFP10] Benjamin Recht, Maryam Fazel, and Pablo A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Rev.*, 52(3):471–501, January 2010. 87

[RS96] Hong S. Ryoo and Nikolaos V. Sahinidis. A branch-and-reduce approach to global optimization. *J. Global Optim.*, 8(2):107–138, March 1996. 9

[SA99] Hanif D. Sherali and Warren P. Adams. *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems*. Springer US, 1999. 12

[Sal17] Saverio Salzo. The variable metric forward-backward splitting algorithm under mild differentiability assumptions. *SIAM J. Optim.*, 27(4):2153–2181, January 2017. 76

[SBFA15] Emmanuel Soubies, Laure Blanc-Féraud, and Gilles Aubert. A continuous exact ℓ_0 penalty (CEL0) for least squares regularized problem. *SIAM J. Imaging Sci.*, 8(3):1607–1639, January 2015. 25, 26, 28

[SBFA17] Emmanuel Soubies, Laure Blanc-Féraud, and Gilles Aubert. A unified view of exact continuous penalties for $\ell_2 - \ell_0$ minimization. *SIAM J. Optim.*, 27(3):2034–2060, January 2017. 29

[SBL12] Laurent Sorber, Marc Van Barel, and Lieven De Lathauwer. Unconstrained optimization of real functions in complex variables. *SIAM J. Optim.*, 22(3):879–898, January 2012. 87

[SBL13] Laurent Sorber, Marc Van Barel, and Lieven De Lathauwer. Optimization-based algorithms for tensor decompositions: Canonical polyadic decomposition, decomposition in rank-$(l_r, l_r, 1)$ terms, and a new generalization. *SIAM J. Optim.*, 23(2):695–720, January 2013. 87

[Sch91] Konrad Schmüdgen. The K-moment problem for compact semi-algebraic sets. *Math. Annalen*, 289(1):203–206, March 1991. 11

[Sch08] Claus Scheiderer. Positivity and sums of squares: A guide to recent results. In *Emerging Applications of Algebraic Geometry*, pages 271–324. Springer New York, September 2008. 11

[Sch10] Martin Schetzen. Nonlinear system modelling and analysis from the Volterra and Wiener perspective. In *Lecture Notes in Control and Information Sciences*, pages 13–24. Springer London, 2010. 25

[Sel17] Ivan Selesnick. Sparse regularization via convex analysis. *IEEE Trans. Signal Process.*, 65(17):4481–4494, September 2017. 25, 28

[Shi18] Yaroslav Shitov. A counterexample to Comon's conjecture. *SIAM J. Appl. Algebr. Geom.*, 2(3):428–443, January 2018. 88

[SK90] Eugenio Sanchez and Bruce R. Kowalski. Tensorial resolution: A direct trilinear decomposition. *J. Chemometrics*, 4(1):29–45, January 1990. 87

[SLF+17] Nicholas D. Sidiropoulos, Lieven De Lathauwer, Xiao Fu, Kejun Huang, Evangelos E. Papalexakis, and Christos Faloutsos. Tensor decomposition for signal processing and machine learning. *IEEE Trans. Signal Process.*, 65(13):3551–3582, July 2017. 87, 88, 89, 92

[Sot11] Renata Sotirov. SDP relaxations for some combinatorial optimization problems. In *Handbook on Semidefinite, Conic and Polynomial Optimization*, pages 795–819. Springer US, September 2011. 59

[Ste74] Gilbert Stengle. A Nullstellensatz and a Positivstellensatz in semialgebraic geometry. *Math. Annalen*, 207(2):87–97, June 1974. 10

[Stu99] Jos F. Sturm. Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones. *Optim. Methods Softw.*, 11(1-4):625–653, January 1999. 59, 79

[Tel17] Matus Telgarsky. Neural networks and rational functions. In *Proc. Int. Conf. Mach. Learn.*, volume 70 of *ICML'17*, pages 3387–3393. JMLR.org, 2017. 9

[TH01] C. Vincent Tao and Yong Hu. A comprehensive study of the rational function model for photogrammetric processing. *Photogrammetric Engineering and Remote Sensing*, 67(12):1347–1358, 2001. 8

[Tib96] Robert Tibshirani. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 58(1):267–288, 1996. 25, 34

[TK19] Nima Taherkhani and Kamran Kiasaleh. Statistical modelling of the clipping noise in OFDM-based visible light communication system, 2019. 44

[Tod01] Michael J. Todd. Semidefinite optimization. *Acta Numerica*, 10, May 2001. 64

[TPAB09] Hugues Talbot, Harold Phelippeau, Mohamed Akil, and Stefan Bara. Efficient poisson denoising for photography. In *Proc. Int. Conf. Image Process.* IEEE, November 2009. 43

[TS15] Gongguo Tang and Parikshit Shah. Guaranteed tensor decomposition: A moment approach. In Francis Bach and David Blei, editors, *Proc. Int. Conf. Mach. Learn.*, volume 37 of *Proceedings of Machine Learning Research*, pages 1491–1500, Lille, France, 07–09 Jul 2015. PMLR. 88

[TTT99] Kim-Chuan Toh, Michael J. Todd, and Reha H. Tütüncü. SDPT3 — a Matlab software package for semidefinite programming, version 1.3. *Optim. Methods Softw.*, 11(1-4):545–581, January 1999. 35, 46, 55, 59, 104, 106

[Tuy16] Hoang Tuy. *Convex Analysis and Global Optimization*. Springer International Publishing, 2016. 9

[VDS+16] Nico Vervliet, Otto Debals, Laurent Sorber, Marc Van Barel, and Lieven De Lathauwer. Tensorlab 3.0, March 2016. Available online. 95, 101, 104, 106

[VDSL14] Nico Vervliet, Otto Debals, Laurent Sorber, and Lieven De Lathauwer. Breaking the curse of dimensionality using decompositions of incomplete tensors: Tensor-based scientific computing in big data analysis. *IEEE Signal Process. Mag.*, 31(5):71–79, September 2014. 103

[VRGB+05] Colombe Vendeuvre, Rosario Ruiz-Guerrero, Fabrice Bertoncini, Laurent Duval, Didier Thiébaut, and Marie-Claire Hennion. Characterisation of middle-distillates by comprehensive two-dimensional gas chromatography (GC × GC): A powerful alternative for performing various standard analysis of middle-distillates. *J. Chromatogr. A*, 1086(1-2):21–28, 2005. 26

[VRGB+07] Colombe Vendeuvre, Rosario Ruiz-Guerrero, Fabrice Bertoncini, Laurent Duval, and Didier Thiébaut. Comprehensive two-dimensional gas chromatography for detailed characterisation of petroleum products. *Oil Gas Sci. Tech.*, 62(1):43–55, 2007. 26

[WGY10] Zaiwen Wen, Donald Goldfarb, and Wotao Yin. Alternating direction augmented Lagrangian methods for semidefinite programming. *Math. Program. Comput.*, 2(3-4):203–230, September 2010. 74, 80

[WKKM06] Hayato Waki, Sunyoung Kim, Masakazu Kojima, and Masakazu Muramatsu. Sums of squares and semidefinite program relaxations for polynomial optimization problems with structured sparsity. *SIAM J. Optim.*, 17(1):218–242, July 2006. 20

[WNM11] Hayato Waki, Maho Nakata, and Masakazu Muramatsu. Strange behaviors of interior-point methods for solving semidefinite programming problems in polynomial optimization. *Comput. Optim. Appl.*, 53(3):823–844, September 2011. 84

[WW18] Irène Waldspurger and Alden Waters. Rank optimality for the Burer-Monteiro factorization. ArXiv Preprint, arXiv:1812.03046, December 2018. 59, 112

[YL09] Jie Yan and Wu-Sheng Lu. Towards global design of orthogonal filter banks and wavelets. *Can. J. Elect. Comput. E.*, 34(4):145–151, 2009. 9

[YST15] Liuqin Yang, Defeng Sun, and Kim-Chuan Toh. SDPNAL+ : a majorized semismooth newton-CG augmented lagrangian method for semidefinite programming with nonnegative constraints. *Math. Program. Comput.*, 7(3):331–366, May 2015. 74, 84

[ZdG11] Youwei Zhang, Alexandre d'Aspremont, and Laurent El Ghaoui. Sparse PCA: Convex relaxations, algorithms and applications. In *Handbook on Semidefinite, Conic and Polynomial Optimization*, pages 915–940. Springer US, September 2011. 59

[ZFP+17] Yang Zheng, Giovanni Fantuzzi, Antonis Papachristodoulou, Paul Goulart, and Andrew Wynn. Fast ADMM for semidefinite programs with chordal sparsity. In *Proc. American Control Conf.* IEEE, May 2017. 74, 79

[Zha10a] Cun-Hui Zhang. Nearly unbiased variable selection under minimax concave penalty. *Ann. Appl. Stat.*, 38(2):894–942, April 2010. 26, 28

[Zha10b] Tong Zhang. Analysis of multi-stage convex relaxation for sparse regularization. *J. Mach. Learn. Res.*, 11:1081–1107, March 2010. 26, 28

[ZKOM18] Abdelhak M. Zoubir, Visa Koivunen, Esa Ollila, and Michael Muma. *Robust Statistics for Signal Processing*. Cambridge University Press, October 2018. 51

[ZST10] Xin-Yuan Zhao, Defeng Sun, and Kim-Chuan Toh. A Newton-CG augmented Lagrangian method for semidefinite programming. *SIAM J. Optim.*, 20(4):1737–1765, January 2010. 59

[ZXZ19] Shu Zhang, Youshen Xia, and Changzhong Zou. An adaptive regularization method for low-dose CT reconstruction from CT transmission data in Poisson-Gaussian noise. *Optik*, 188:172–186, July 2019. 43

[ZZZ+16] Qibin Zhao, Guoxu Zhou, Liqing Zhang, Andrzej Cichocki, and Shun-Ichi Amari. Bayesian robust tensor factorization for incomplete multiway data. *IEEE Trans. Neural Netw. Learn. Syst.*, 27(4):736–748, April 2016. 87