# The quest of modeling, certification and efficiency in polynomial optimization

Victor Magron

**Université de Toulouse Paul Sabatier**

**École Doctorale EDMITT**

# Mémoire d'habilitation à diriger les recherches

Victor MAGRON

# The quest of modeling, certification and efficiency in polynomial optimization

defended on May 25th 2021 at LAAS CNRS

**Jury:**

*Rapporteurs :*

| | | |
|---|---|---|
| Bernard MOURRAIN | Directeur de recherche | INRIA, Sophia-Antipolis |
| Mihai PUTINAR | Professor | University of California |
| Anders RANTZER | Professor | LTH, Lund |

*Examinateurs :*

| | | |
|---|---|---|
| Antonio ACÍN | Professor | ICFO, Barcelona |
| Salma KUHLMANN | Professor | University of Konstanz |
| Jean-Bernard LASSERRE | Directeur de recherche | LAAS CNRS, IMT Toulouse |
| Patrick PANCIATICI | Scientific Advisor | RTE Paris |
| Bruno SALVY | Directeur de recherche | INRIA, ENS Lyon |

## Remerciements

Je tiens à commencer ce manuscrit en remerciant chaleureusement mon cher collègue émérite et parrain de HDR. Merci Jean-Bernard pour ton dynamisme, ta créativité, ta vivacité d'esprit et ton humour. Tes recherches ont inspiré et inspireront encore longtemps de nombreux chercheurs.

Merci à toutes les équipes et leurs chercheurs qui m'ont accueilli après ma thèse : l'équipe MAC du LAAS, l'équipe Circuits and Systems de l'Imperial College, l'équipe Tempo de Verimag, l'équipe Polsys du LIP6 et l'équipe MAC à nouveau. Merci à Dimitri, Lucie, Didier, Liviu et Jamal, pour m'avoir soutenu à de multiples reprises, dans les mauvais comme les bons moments.

Special thanks to you Anders for welcoming me at LTH, accepting to be a referee and for your wonderful support. Merci Eva pour ton aide et ton support, qui ont contribué à rendre si agréable mon séjour à LTH. Je remercie également mes deux autres rapporteurs, Bernard et Mihai, mes examinateurs Toni, Patrick et Bruno, ainsi que Salma pour avoir accepté de présider ce fantastique jury.

Merci à tous mes collaborateurs, vous vous reconnaîtrez certainement sur la dernière planche de mon exposé.

Thank you Yoshimura Sensei for guiding me during my early research quests.

Merci à Benjamin, Stéphane et Xavier pour avoir été des directeurs de thèse exemplaires. Je tente chaque jour de m'inspirer de votre confiance, rigueur et ouverture d'esprit.

Special thanks to my PhD students: Hieu, Tong and Hoang, and a (dense) thank you Jie for your impressive creativity and kindness.

Merci à Philippe, David, Thomas et Julien pour m'avoir guidé aux quatre coins de la France dans la quête du poumsae.

Merci à tous mes amis et à tous mes proches, dispersés aux quatre coins du globe, loin des yeux mais toujours proches du cœur. Merci à Djo pour ton écoute, ton support, ton amitié précieuse, et pour m'aider à distinguer définition et propriété.

Merci à mes deux filleules, Siloé et Constantine pour illuminer ce monde de votre récente présence. J'espère qu'un jour vous lirez ces lignes.

Merci à mes parents pour leur soutien précieux : ma Maman chérie, experte en violettes, et mon Papa chéri, expert du MAP.

Merci à mes quatre fantastiques frères pour votre amour, votre intelligence, votre humour et votre bienveillance.

Merci à Misstick pour m'avoir guidé dans la quête du saut à l'élastique.

Un merci spécial à Pauline pour ton amour, ta fidélité, ton soutien, et pour avoir grandement contribué à améliorer la qualité de ma soutenance.

# Contents

# List of Acronyms

**CS** correlative sparsity ..... 104

**csp** correlative sparsity pattern ..... 104

**GP** geometric programming ..... 90

**GMP** generalized moment problem ..... 2

**GNS** Gelfand-Naimark-Segal ..... 6

**LMI** linear matrix inequality ..... 64

**LP** linear programming ..... 2

**moment-SOS** moment-sums of squares ..... 1

**POP** polynomial optimization problem ..... 2

**REP** relative entropy programming ..... 90

**RS** reachable set ..... 8

**SAGE** sum of arithmetic-geometric-mean-exponentials ..... 52

**SDP** semidefinite programming ..... 2

**SONC** sum of nonnegative circuits ..... 52

**SOHS** sum of Hermitian squares ..... 39

**SOS** sum of squares ..... 2

**TS** term sparsity ..... 127

**tsp** correlative sparsity pattern ..... 127

# Introduction

This manuscript summarizes the main theoretical and algorithmic frameworks, the objectives and research outcomes that I obtained in the last few years, together with new perspectives. These results have been obtained since my recruitment as a CNRS junior researcher in 2015, and some of them relate to my corresponding research project defended at the entrance competition. Along the three main chapters, the reader will find many illustrating examples and figures. The proofs of the different theoretical statements have been removed for the sake of clarity and can be found in complementary references, available in open access platforms such as arXiv or HAL.

**Research field and motivation.** Simple mistakes arising in the design of modern cyber-physical systems can have tragic impacts, from human and economic points of view. In particular, for embedded systems, one tries to avoid incidents such as the Patriot missile crash in 1991, the FDIV Pentium bug in 1994 or more recently the collision of Google's self-driving car in 2016. To ensure the safety of such systems, program verification tools allow to validate assertions related to program executions. In the linear setting, there are numerous efficient and safe algorithms dedicated to static analysis, producing invariants, or bounding the error due to finite precision representation. These verification tools include both programming and specification languages but also an interface with decision procedures relying on optimization algorithms. Modern verification software still have limited capacities of handling nonlinear optimization problems involving polynomials. This leads to either inaccurate bounds or high analysis time. My research aims to overcome these limitations by widening the application range of verification tools to the (nonlinear) polynomial case.

Certified optimization techniques have successfully tackled challenging verification problems in various fundamental and industrial applications. The formal verification of thousands of nonlinear inequalities arising in the famous proof of Kepler conjecture [J2] was achieved in August 2014.[1] I was involved in the related project called Flyspeck during my PhD in 2010-2013. In energy networks, it is now possible to compute the solution of large-scale power flow problems with up to thousand variables [148]. This success follows from growing research efforts in polynomial optimization, an emerging field extensively developed in the last two decades. One key advantage of these techniques is the ability to model a wide range of problems using optimization formulations, which can be in turn solved with efficient numerical tools. My methodology heavily relies on such methods, including the moment-sums of squares (moment-SOS) approach by Lasserre [180] which provides numerical certificates for positive polynomials as well as recently developed alternative methods. However, such optimization methods still encompass many major issues on both practical and theoretical sides: scalability, unknown complexity bounds, ill-conditionning of numerical solvers, lack of exact certification, convergence guarantees. This manuscripts presents results in this line of research with the long-term perspective of obtaining scientific breakthroughs to handle certification of nonlinear systems arising in real-world applications.

**Polynomial optimization** focuses on minimizing or maximizing a polynomial under a set of polynomial inequality constraints. A polynomial is an expression involving addition, subtraction and multiplication of variables and coefficients. An example of polynomial in two variables $x_1$ and $x_2$ with rational coefficients is $f(x_1, x_2) = 1/3 + x_1^2 + 2x_1x_2 + x_2^2$. *Semialgebraic* sets are defined with conjunctions and disjunctions of polynomial inequalities with real coefficients. For instance the

[1] https://code.google.com/p/flyspeck/wiki/AnnouncingCompletion

two-dimensional unit disk is a semialgebraic set defined as the set of all points $(x_1, x_2)$ satisfying the (single) inequality $1 - x_1^2 - x_2^2 \geq 0$.

In general, computing the *exact* solution of a polynomial optimization problem (POP) over a semialgebraic set is an NP-hard problem. In practice, one can at least try to compute an *approximation* of the solution by considering a *relaxation* of the problem instead of the problem itself. The approximated solution may not satisfy all the problem constraints but still gives useful information about the exact solution. I illustrate this by considering the minimization of the above polynomial $f(x_1, x_2)$ on the unit disk. One can replace this disk by a larger set, for instance the product of intervals $[-1, 1] \times [-1, 1]$. Using basic interval arithmetics, one easily shows that $f$ belongs to $[-4/3, 4/3]$. Next, one can replace the monomials $x_1^2$, $x_1 x_2$ and $x_2^2$ by three new variables $y_1$, $y_2$ and $y_3$, respectively. One can relax the initial problem by linear programming (LP), with a cost of $1/3 + y_1 + 2y_2 + y_3$ and one single linear inequality constraint $1 - y_1 - y_3 \geq 0$. By hand-solving or by using an LP solver, one finds again a lower bound of $-4/3$. Even if LP gives more accurate bounds than interval arithmetics in general, this does not yield any improvement on this example.

One way to obtain more accurate lower bounds is to rely on more sophisticated techniques from the field of convex optimization, e.g., semidefinite programming (SDP). In the seminal paper [180] published in 2001, Lasserre introduced a hierarchy of relaxations allowing to obtain a converging sequence of lower bounds for the minimum of a polynomial over a semialgebraic set. Each lower bound is computed by SDP. A symmetric matrix is said to be semidefinite positive when all its eigenvalues are nonnegative. In SDP, one optimizes a linear function under the constraint that a given matrix is semidefinite positive. Thus, SDP can be seen as a generalization of LP in some sense. SDP itself is relevant to a wide range of applications (combinatorial optimization [112], control theory [47], matrix completion [191]) and can be solved efficiently, namely in time polynomial in its input size, by freely available software, e.g., SeDuMi [279] or MOSEK [289].

The idea behind Lasserre's hierarchy is to tackle the *infinite-dimensional* initial problem by solving several *finite-dimensional* primal-dual SDP problems. The primal is a *moment* problem, that is an optimization problem where variables are the moments of a Borel measure. The first moment is related to means, the second moment is related to variances, etc. Lasserre showed in [180] that POP can be cast as a particular instance of the generalized moment problem (GMP). In a nutshell, the primal moment problem approximates Borel measures. The dual is a sum of squares (SOS) problem, where the variables are the coefficients of SOS polynomials (e.g. $(1/\sqrt{3})^2 + (x_1 + x_2)^2$). It is known that not all positive polynomials can be written with SOS decompositions. However, when the set of constraints satisfies certain assumptions (slightly stronger than compactness) then one can represent positive polynomials with weighted SOS decompositions. In a nutshell, the dual SOS problem approximates positive polynomials. The moment-SOS approach can be used on the example with either three moment variables or SOS of degree 2 to obtain a lower bound of $1/3$. For this example, the exact solution is obtained at the first step of the hierarchy. There is no need to go further, i.e., to consider primal with moments of greater order (e.g. the integrals of $x_1^3$, $x_1^2 x_2$, $x_1^4$) or dual with SOS polynomials of degree 4 or 6. The reason is that for convex quadratic problems, the first step of the hierarchy gives the exact solution!

For more general problems involving polynomials, there are several difficulties encountered while using the moment-SOS hierarchy. My research is structured in 3 related interconnected layers:

**(1)** **Modeling**: we are interested in relying on the moment-SOS hierarchy to analyze dynamical polynomial systems, either in the discrete-time or continuous-time setting. Examples treated in this manuscript include approximation of reachable sets, supports of invariant measures or boundaries of semialgebraic sets. We also wish to model optimization problems involving noncommuting variables, for example matrices of finite or infinite size, to model quantum physics operators.

**(2) Certification**: by solving the dual SOS problem, one can theoretically compute a positivity certificate for a given polynomial. For practical problems, the situation is rather different. One computes such SOS certificates with SDP solvers often implemented using finite-precision arithmetic. When relying on non-exact solvers, the initial polynomial is *approximately* equal to the SOS certificate. We are interested in designing algorithms which output exact certificates for either unconstrained or constrained optimization problems.

**(3) Scalability** : When the initial problem involves $n$ variables, the $r$-th step relaxation of the moment-SOS hierarchy involves the rapidly prohibitive cost of $\binom{n+r}{r}$ SDP variables. We are interested in improving the scalability of the hierarchy by exploiting the specific sparsity structure of the polynomials involved in real-world problems. Important applications arise from various fields, including computer arithmetic (roundoff error bounds), quantum information (noncommutative optimization), optimal power-flow and deep learning.

The organization of my research program was naturally following my PhD thesis [R6], which focused on obtaining computer-assisted proofs for general nonlinear optimization problems, by means of polynomial approximation of transcendental functions and exact certification, in relation with **(2)**. During my first post-doctoral stay, I started to model problems related to set estimation, such as Pareto curves, images of semialgebraic sets, in relation with **(1)**. During my second post-doctoral stay, I focused more specifically on scalability issues encountered in optimization problems coming from computer arithmetic, in relation with **(3)**. After joining in October 2015 the Tempo team at CNRS VERIMAG, I worked on several modeling topics mentioned in **(1)**. During my long-term visit in the joint PolSys team at CNRS LIP6, I focused on exact certification aspects mentioned in **(2)**. Finally, I was affiliated to the MAC team at CNRS LAAS in 2019, whose main goals include providing constructive theoretical conditions for extracting solutions to various control and optimization problems, while designing effective computational algorithms. In this context, my research is devoted to modeling, certification and efficient solving of problems with polynomial data, arising from modern real-world applications such as energy networks, quantum information, control systems, and deep learning.

**Document outline** In the sequel, I report on several research outcomes obtained from October 2015 to December 2020. The exhaustive list of publications is displayed at the end of the manuscript. The references that I co-authored are cited with a prefix as follows: J for journal, C for proceeding in peer-reviewed international conferences and R for research report (submitted to a journal or a conference but unpublished yet).

1. Chapter 1 is dedicated to modeling problems involving polynomials as optimization programs. First, we recall preliminary notions on positive polynomials, Borel measures, and their respective approximation with SOS and truncated moment sequences. We explain in Section 1.2 how to approximate as closely as desired the reachable set of discrete-time polynomial systems with a hierarchy of SDP. We derive similar converging hierarchies to approximate the support/density of invariant measures for polynomial systems in Section 1.3, and to approximate the moments of the boundary measure of semialgebraic sets in Section 1.4. We illustrate the practical convergence behavior of each hierarchy with numerical experiments. Another converging hierarchy is given in Section 1.5 to optimize over trace polynomials, i.e., polynomials in noncommuting variables and traces of their products. The results presented in this chapter are available in [J10, J19, J8, J4, R3].

2. Chapter 2 focuses on certified or "exact" POP, despite the fact that one relies mostly on numerical "inexact" solvers to compute approximate bounds. Section 2.1 interprets some wrong results, due to numerical inaccuracies, already observed when solving SDP relaxations for

POP on a double precision floating point SDP solver. Then, we describe, analyze and compare, from the theoretical and practical points of view, several algorithms to obtain exact nonnegativity certificates for polynomials either in the unconstrained or constrained case. In Section 2.2, we provide two algorithms computing weighted SOS decomposition with rational coefficients for univariate polynomials with rational coefficients. We show how to extend this to the multivariate case in Section 2.3. At last, we consider in Section 2.4 alternative certificates and give two algorithms computing exact sums of nonnegative circuits and sums of arithmetic-geometric-exponentials decompositions. The results presented in this chapter have been published in [J5, J17, C8, C9, J16, C10]. Our certification algorithms have been implemented in the `RealCertify` library available in Maple.

3. Chapter 3 is dedicated to exploit the sparsity structure of the input data to solve large-scale POP. First, we recall in Section 3.1 some background on correlative sparsity, occurring when there are flew correlations between the variables of the input problem. Then we apply this framework in Section 3.2 to provide efficiently upper bounds on roundoff errors of floating-point nonlinear programs, involving polynomials. A very distinct application is described for optimization of polynomials in noncommuting variables in Section 3.3. A converging hierarchy of semidefinite relaxations for eigenvalue and trace optimization is provided. A complementary framework is presented in Section 3.4, where we show how to exploit term sparsity of the input polynomials to obtain a new converging hierarchy of SDP relaxations. Our theoretical framework is then applied to compute lower bounds for POP coming from the networked systems literature. Finally, we explain how to combine correlative and term sparsity in Section 3.5. The results outlined in this chapter are available in [J13, J14, J3, J22, J21, R14, C13, R13, R5]. Our sparsity exploiting algorithms have been implemented in the TSSOS library available in Julia and are the focus of a dedicated article [R5].

4. Chapter 4 summarizes the main future investigation tracks related to the research outcomes from the three contribution chapters. I outline further research topics together with potentially useful references.

At last, I provide a CV detailing my PhD candidate and Postdoctoral fellow supervision, teaching, developed software, conference organization, research projects and grants I am involved in.

# CHAPTER 1
# Modeling with polynomial optimization

## Contents

This chapter focuses on modeling various problems arising from dynamical systems and noncommutative optimization, with the moment-SOS hierarchy.

- We consider in Section 1.2 the problem of approximating the reachable set of a discrete-time polynomial system from a semialgebraic set of initial conditions under general semialgebraic set constraints. Assuming inclusion in a given simple set like a box or an ellipsoid, we provide a method to compute certified outer approximations of the reachable set. The proposed method consists of building a hierarchy of relaxations for an infinite-dimensional moment problem. Under certain assumptions, the optimal value of this problem is the volume of the reachable set and the optimum solution is the restriction of the Lebesgue measure on this set. Then, one can outer approximate the reachable set as closely as desired with a hierarchy of super level sets of increasing degree polynomials. For each fixed degree, finding the coefficients of the polynomial boils down to computing the optimal solution of a convex semidefinite program. When the degree of the polynomial approximation tends to infinity, we provide strong convergence guarantees of the super level sets to the reachable set. We also present some application examples together with numerical results.

- In Section 1.3, we consider the problem of approximating numerically the moments and the supports of measures which are invariant with respect to the dynamics of continuous- and discrete-time polynomial systems, under semialgebraic set constraints. First, we address the problem of approximating the density and hence the support of an invariant measure which is absolutely continuous with respect to the Lebesgue measure. Then, we focus on the approximation of the support of an invariant measure which is singular with respect to the Lebesgue measure. Each problem is handled through an appropriate reformulation into a linear optimization problem over measures, solved in practice with two hierarchies of finite-dimensional semidefinite moment-SOS relaxations. Under specific assumptions, the first moment-SOS hierarchy allows to approximate the moments of an absolutely continuous invariant measure as close as desired and to extract a sequence of polynomials converging weakly to the density of this measure. The second hierarchy allows to approximate as close as desired in the Hausdorff metric the support of a singular invariant measure with the level sets of the Christoffel polynomials associated to the moment matrices of this measure. We also present some application examples together with numerical results for several dynamical systems admitting either absolutely continuous or singular invariant measures. This work was jointly pursued during the (unofficial) supervision of the postdoc of M. Forets (now at UTEC) when I was affiliated to CNRS VERIMAG.

- Given a compact basic semialgebraic set we provide in Section 1.4 a numerical scheme to approximate as closely as desired, any finite number of moments of the Hausdorff measure on the boundary of this set. This also allows one to approximate interesting quantities like length, surface, or more general integrals on the boundary, as closely as desired from above and below.
- Section 1.5 is motivated by recent progress in quantum information theory, and aims at modeling optimization problems over trace polynomials, i.e., polynomials in noncommuting variables and traces of their products. A novel Positivstellensatz certifying positivity of trace polynomials subject to trace constraints is presented, and a hierarchy of semidefinite relaxations converging monotonically to the optimum of a trace polynomial subject to tracial constraints is provided. This hierarchy can be seen as a tracial analog of the Pironio, Navascués and Acín (NPA) hierarchy for optimization of noncommutative polynomials. The Gelfand-Naimark-Segal (GNS) construction is applied to extract optimizers of the trace optimization problem if flatness and extremality conditions are satisfied. These conditions are sufficient to obtain finite convergence of our hierarchy. The main techniques used are inspired by real algebraic geometry, operator theory, and noncommutative algebra.

These contributions are in collaboration with researchers working in polynomial optimization: D. Henrion (Senior researcher, LAAS), J.-B. Lasserre (Senior researcher, LAAS), program verification: P.-L. Garoche (Professor, ENAC), Xavier Thirioux (Assistant Professor, INPT/IRIT) as well as in real algebraic geometry: I. Klep (Professor, University of Ljubljana) and his former PhD student J. Volčič (Postdoc, Texas A & M University).

In what follows, some preliminary notions of polynomial optimization are given in Section 1.1, before presenting the new contribution in Section 1.2.

## 1.1 Preliminary notions

### Polynomials and sums of squares

Let $\mathbb{N}$ (resp. $\mathbb{N}_{>0}$) stands for the set of nonnegative (resp. positive) integers. Given $r, n \in \mathbb{N}$, let $\mathbb{R}[\mathbf{x}]$ (resp. $\mathbb{R}[\mathbf{x}]_{2r}$) stands for the vector space of real-valued $n$-variate polynomials (resp. of degree at most $2r$) in the variable $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$. Let $\mathbb{C}[\mathbf{x}]$ be the vector space of complex-valued $n$-variate polynomials. A basic compact semialgebraic set $\mathbf{X}$ is a finite conjunction of polynomial super levelsets. Namely, given $m \in \mathbb{N}$ and polynomials $g_1, \dots, g_m \in \mathbb{R}[\mathbf{x}]$, one has

$$\mathbf{X} := \{\mathbf{x} \in \mathbb{R}^n : g_1(\mathbf{x}) \geq 0, \dots, g_m(\mathbf{x}) \geq 0\} . \tag{1.1}$$

Let $\Sigma[\mathbf{x}]$ stand for the cone of polynomial SOS and let $\Sigma[\mathbf{x}]_r$ denote the cone of SOS polynomials of degree at most $2r$, namely $\Sigma[\mathbf{x}]_r := \Sigma[\mathbf{x}] \cap \mathbb{R}[\mathbf{x}]_{2r}$.

For the ease of further notation, we set $g_0(\mathbf{x}) := 1$, and $r_j := \lceil (\deg g_j)/2 \rceil$, for all $j = 0, \dots, m$. Given a basic compact semialgebraic set $\mathbf{X}$ as above and an integer $r$, let $\mathcal{M}(\mathbf{X})_r$ be the $r$-truncated quadratic module generated by $g_0, \dots, g_m$:

$$\mathcal{M}(\mathbf{X})_r := \left\{ \sum_{j=0}^{m} s_j(\mathbf{x}) g_j(\mathbf{x}) : s_j \in \Sigma[\mathbf{x}]_{r-r_j}, \, j = 0, \dots, m \right\} .$$

To guarantee the convergence behavior of the relaxations presented in the sequel, we need to ensure that polynomials which are positive on $\mathbf{X}$ lie in $\mathcal{M}(\mathbf{X})_r$ for some $r \in \mathbb{N}$. The existence of such SOS-based representations is guaranteed by Putinar's Positivstellensaz (see, e.g., [182, Section 2.5]), when the following condition holds:

**Assumption 1.1.1** *There exists a large enough integer $N$ such that one of the polynomials describing the set $\mathbf{X}$ is equal to $N - \|\mathbf{x}\|_2^2$.*

This assumption is slightly stronger than compactness. Indeed, compactness of $\mathbf{X}$ already ensures that each variable has finite lower and upper bounds. One (easy) way to ensure that Assumption 1.1.1 holds is to add a redundant constraint involving a well-chosen $N$ depending on these bounds, in the definition of $\mathbf{X}$.

## Borel measures

Given a compact set $\mathbf{A} \subset \mathbb{R}^n$, we denote by $\mathscr{M}(\mathbf{A})$ the vector space of finite signed Borel measures supported on $\mathbf{A}$, namely real-valued functions from the Borel sigma algebra $\mathcal{B}(\mathbf{A})$. The support of a measure $\mu \in \mathscr{M}(\mathbf{A})$ is defined as the closure of the set of all points $\mathbf{x}$ such that $\mu(\mathbf{B}) \neq 0$ for any open neighborhood $\mathbf{B}$ of $\mathbf{x}$. We note $\mathscr{C}(\mathbf{A})$ the Banach space of continuous functions on $\mathbf{A}$ equipped with the sup-norm. Let $\mathscr{C}(\mathbf{A})'$ stand for the topological dual of $\mathscr{C}(\mathbf{A})$ (equipped with the sup-norm), i.e., the set of continuous linear functionals of $\mathscr{C}(\mathbf{A})$. By a Riesz identification theorem (see for instance [196]), $\mathscr{C}(\mathbf{A})'$ is isomorphically identified with $\mathscr{M}(\mathbf{A})$ equipped with the total variation norm denoted by $\|\cdot\|_{\mathrm{TV}}$. Let $\mathscr{C}_+(\mathbf{A})$ (resp. $\mathscr{M}_+(\mathbf{A})$) stand for the cone of nonnegative elements of $\mathscr{C}(\mathbf{A})$ (resp. $\mathscr{M}(\mathbf{A})$). The topology in $\mathscr{C}_+(\mathbf{A})$ is the strong topology of uniform convergence in contrast with the weak-star topology in $\mathscr{M}_+(\mathbf{A})$. See [252, Section 21.7] and [27, Chapter IV] or [201, Section 5.10] for functional analysis, measure theory and applications in convex optimization.

With $\mathbf{X}$ a basic compact semialgebraic set, the restriction of the Lebesgue measure on a subset $\mathbf{A} \subseteq \mathbf{X}$ is $\lambda_{\mathbf{A}}(d\mathbf{x}) := \mathbf{1}_{\mathbf{A}}(\mathbf{x})\, d\mathbf{x}$, where $\mathbf{1}_{\mathbf{A}} : \mathbf{X} \to \{0,1\}$ stands for the indicator function of $\mathbf{A}$, namely $\mathbf{1}_{\mathbf{A}}(\mathbf{x}) = 1$ if $\mathbf{x} \in \mathbf{A}$ and $\mathbf{1}_{\mathbf{A}}(\mathbf{x}) = 0$ otherwise. A sequence $\mathbf{y} := (y_\beta)_{\beta \in \mathbb{N}^n} \in \mathbb{R}^{\mathbb{N}^n}$ is said to have a representing measure on $\mathbf{X}$ if there exists $\mu \in \mathscr{M}(\mathbf{X})$ such that $y_\beta = \int \mathbf{x}^\beta \mu(d\mathbf{x})$ for all $\beta \in \mathbb{N}^n$.

The moments of the Lebesgue measure on $\mathbf{A}$ are denoted by

$$y_\beta^{\mathbf{A}} := \int \mathbf{x}^\beta \lambda_{\mathbf{A}}(d\mathbf{x}) \in \mathbb{R}, \quad \beta \in \mathbb{N}^n \tag{1.2}$$

where we use the multinomial notation $\mathbf{x}^\beta := x_1^{\beta_1} x_2^{\beta_2} \ldots x_n^{\beta_n}$. The Lebesgue volume of $\mathbf{A}$ is $\operatorname{vol} \mathbf{A} := y_0^{\mathbf{A}} = \int \lambda_{\mathbf{A}}(d\mathbf{x})$.

Given $\mu, \nu \in \mathscr{M}(\mathbf{A})$, the notation

$$\mu \leq \nu$$

stands for $\nu - \mu \in \mathscr{M}_+(\mathbf{A})$, and we say that $\mu$ is *dominated* by $\nu$. Given $\mu \in \mathscr{M}_+(\mathbf{A})$, there exists a unique Lebesgue decomposition $\mu = \nu + \psi$ with $\nu, \psi \in \mathscr{M}_+(\mathbf{A})$, $\nu \ll \lambda$ and $\psi \perp \lambda$. Here, the notation $\nu \ll \lambda$ means that $\nu$ is absolutely continuous with respect to (w.r.t.) $\lambda$, that is, for every $\mathbf{A} \in \mathcal{B}(\mathbf{X})$, $\lambda(\mathbf{A}) = 0$ implies $\nu(\mathbf{A}) = 0$. The notation $\psi \perp \lambda$ means that $\psi$ is singular w.r.t. $\lambda$, that is, there exist disjoint sets $\mathbf{A}, \mathbf{B} \in \mathcal{B}(\mathbf{X})$ such that $\mathbf{A} \cup \mathbf{B} = \mathbf{X}$ and $\psi(\mathbf{A}) = \lambda(\mathbf{B}) = 0$.

Given $\mu \in \mathscr{M}_+(\mathbf{X})$, the so-called pushforward measure or *image measure*, see, e.g., [4, Section 1.5], of $\mu$ under $f$ is defined as follows:

$$f_\# \mu(\mathbf{A}) := \mu(f^{-1}(\mathbf{A})) = \mu(\{\mathbf{x} \in \mathbf{X} : f(\mathbf{x}) \in \mathbf{A}\})$$

for every set $\mathbf{A} \in \mathcal{B}(\mathbf{X})$. The main property of the pushforward measure is the change-of-variable formula: $\int_{\mathbf{A}} v(\mathbf{x}) f_\# \mu(d\mathbf{x}) = \int_{f^{-1}(\mathbf{A})} v(f(\mathbf{x})) \mu(d\mathbf{x})$, for all $v \in \mathscr{C}(\mathbf{A})$.

### Moment and localizing matrices

For all $r \in \mathbb{N}$, we set $\mathbb{N}_r^n := \{\beta \in \mathbb{N}^n : \sum_{j=1}^n \beta_j \leq r\}$, whose cardinality is $\binom{n+r}{r}$. Then a polynomial $g \in \mathbb{R}[\mathbf{x}]$ is written as follows:

$$\mathbf{x} \mapsto g(\mathbf{x}) \; = \; \sum_{\beta \in \mathbb{N}^n} g_\beta \, \mathbf{x}^\beta \, ,$$

and $g$ is identified with its vector of coefficients $\mathbf{g} = (g_\beta)$ in the canonical basis $(\mathbf{x}^\beta)$, $\beta \in \mathbb{N}^n$.

Given a real sequence $\mathbf{y} = (y_\beta)_{\beta \in \mathbb{N}^n}$, let us define the linear functional $L_\mathbf{y} : \mathbb{R}[\mathbf{x}] \to \mathbb{R}$ by $L_\mathbf{y}(g) := \sum_\beta g_\beta y_\beta$, for every polynomial $g$.

Then, we associate to $\mathbf{y}$ the so-called *moment matrix* $\mathbf{M}_r(\mathbf{y})$, that is the real symmetric matrix with rows and columns indexed by $\mathbb{N}_r^n$ and the following entrywise definition:

$$(\mathbf{M}_r(\mathbf{y}))_{\beta,\gamma} := L_\mathbf{y}(\mathbf{x}^{\beta+\gamma}) \, , \quad \forall \beta, \gamma \in \mathbb{N}_r^n \, .$$

Given $g \in \mathbb{R}[\mathbf{x}]$, we also associate to $\mathbf{y}$ the so-called *localizing matrix*, that is the real symmetric matrix $\mathbf{M}_r(g\,\mathbf{y})$ with rows and columns indexed by $\mathbb{N}_r^n$ and the following entrywise definition:

$$(\mathbf{M}_r(g\,\mathbf{y}))_{\beta,\gamma} := L_\mathbf{y}(g(\mathbf{x})\,\mathbf{x}^{\beta+\gamma}) \, , \quad \forall \beta, \gamma \in \mathbb{N}_r^n \, .$$

Let $\mathbf{X}$ be a basic compact semialgebraic set as in (1.1). Then it is easy to check that if $\mathbf{y}$ has a representing measure $\mu \in \mathscr{M}_+(\mathbf{X})$ then $\mathbf{M}_r(g_j\,\mathbf{y}) \succeq 0$, for all $j = 0, \ldots, m$ (the notation $\succeq 0$ stands for positive semidefinite).

## 1.2 Reachable sets of polynomial systems

Given a dynamical polynomial system described by a continuous-time or discrete-time equation, the (forward) reachable set (RS) is the set of all states that can be reached from a set of initial conditions under general state constraints. This set appears in different fields such as optimal control, hybrid systems or program analysis. In general, computing or even approximating the RS is a challenge. Note that the RS is typically non-convex and non-connected, even in the case when the set of initial conditions is convex and the dynamics are linear. We are interested in the polynomial discrete-time system defined by

- a set of initial constraints assumed to be compact basic semialgebraic:

$$\mathbf{X}^0 := \{\mathbf{x} \in \mathbb{R}^n : g_1^0(\mathbf{x}) \geq 0, \ldots, g_{m^0}^0(\mathbf{x}) \geq 0\} \tag{1.3}$$

  defined by given polynomials $g_1^0, \ldots, g_{m^0}^0 \in \mathbb{R}[\mathbf{x}]$, $m^0 \in \mathbb{N}_{>0}$;

- a polynomial transition map $f : \mathbb{R}^n \to \mathbb{R}^n$, $\mathbf{x} \mapsto f(\mathbf{x}) := (f_1(\mathbf{x}), \ldots, f_n(\mathbf{x})) \in \mathbb{R}^n[\mathbf{x}]$ of degree $d := \max\{\deg f_1, \ldots, \deg f_n\}$.

Given $T \in \mathbb{N}$, let us define the set of all admissible trajectories after at most $T$ iterations of the polynomial transition map $f$, starting from any initial condition in $\mathbf{X}^0$:

$$\mathbf{X}^T := \mathbf{X}^0 \cup f(\mathbf{X}^0) \cup f(f(\mathbf{X}^0)) \cup \cdots \cup f^T(\mathbf{X}^0) \, ,$$

with $f^T$ denoting the $T$-fold composition of $f$. Then, we consider the RS of all admissible trajectories:

$$\mathbf{X}^\infty := \lim_{T \to \infty} \mathbf{X}^T$$

and we make the following assumption in the sequel:

**Assumption 1.2.1** *The RS $\mathbf{X}^\infty$ is included in a given basic compact semialgebraic set as in (1.1):*

$$\mathbf{X} := \{\mathbf{x} \in \mathbb{R}^n : g_1(\mathbf{x}) \geq 0, \ldots, g_m(\mathbf{x}) \geq 0\} \tag{1.4}$$

*defined by polynomials $g_1, \ldots, g_m \in \mathbb{R}[\mathbf{x}]$, $m \in \mathbb{N}$.*

**Example 1.2.1** *Let us consider $\mathbf{X}^0 = [1/2, 1]$ and $f(x) = x/4$. Then $\mathbf{X}^\infty = [1/2, 1] \cup [1/8, 1/4] \cup [1/32, 1/16] \ldots$ is included in the basic compact semialgebraic set $\mathbf{X} = [0, 1]$, so Assumption 1.2.1 holds. Note that $\mathbf{X}^\infty$ is not connected within $\mathbf{X}$.*

We denote the closure of $\mathbf{X}^\infty$ by $\bar{\mathbf{X}}^\infty$ . Obviously $\mathbf{X}^\infty \subseteq \bar{\mathbf{X}}^\infty$ and the inclusion can be strict. To circumvent this difficulty later on, we make the following assumption in the remainder of this section.

**Assumption 1.2.2** *The volume of the RS is equal to the volume of its closure, i.e.,* $\operatorname{vol} \mathbf{X}^\infty = \operatorname{vol} \bar{\mathbf{X}}^\infty$.

**Example 1.2.2** *Let $\mathbf{X}^0 = [1/2, 1]$ and $f(x) = x/2$. Then $\mathbf{X}^\infty = [1/2, 1] \cup [1/4, 1/2] \cup [1/8, 1/4] \ldots = (0, 1]$ is a half-closed interval within $\mathbf{X} = [0, 1]$. Note that $\bar{\mathbf{X}}^\infty = \mathbf{X}$, so that $\operatorname{vol} \mathbf{X}^\infty = \operatorname{vol} \bar{\mathbf{X}}^\infty = 1$ and Assumption 1.2.2 is satisfied.*

From now on, the over approximation set $\mathbf{X}$ of the set $\mathbf{X}^\infty$ is assumed to be "simple" (e.g. a ball or a box), meaning that $\mathbf{X}$ fulfills the following condition:

**Assumption 1.2.3** *The moments (1.2) of the Lebesgue measure on $\mathbf{X}$ are available analytically.*

**Remark 1.2.1** *Since we are interested in characterizing the RS of polynomial systems with bounded trajectories, Assumption 1.2.1 and Assumption 1.2.3 are not restrictive. As mentioned above, Assumption 1.1.1 can be ensured by using Assumption 1.2.1. While relying on Assumption 1.2.2, we restrict ourselves to discrete-time systems where the boundary of the RS has zero Lebesgue volume.*

To illustrate these concepts, let us consider the discrete-time polynomial system defined by

$$x_1^+ := \frac{1}{2}(x_1 + 2x_1x_2) \,, \quad x_2^+ := \frac{1}{2}(x_2 - 2x_1^3) \,,$$

with initial state constraints $\mathbf{X}^0 := \{\mathbf{x} \in \mathbb{R}^2 : (x_1 - \frac{1}{2})^2 + (x_2 - \frac{1}{2})^2 \leq 4^{-2}\}$ and general state constraints within the unit ball $\mathbf{X} = \{\mathbf{x} \in \mathbb{R}^2 : \|x\|_2^2 \leq 1\}$. On Figure 1.1, the colored sets of points are obtained by simulation for the first 7 iterates. More precisely, each colored set correspond to (under approximations of) the successive image sets $f(\mathbf{X}^0), \ldots, f^7(\mathbf{X}^0)$ of the points obtained by uniform sampling of $\mathbf{X}^0$ under $f, \ldots, f^7$ respectively. The set $\mathbf{X}^0$ is blue and the set $f^7(\mathbf{X}^0)$ is red, while intermediate sets take intermediate colors. The dotted circle represents the boundary of the unit ball $\mathbf{X}$.

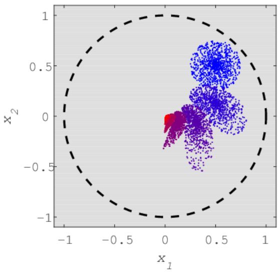


Figure 1.1: Sampling of $\mathbf{X}^\infty$ (color dot points)

## Classical approaches

A classical approach relies on Lyapunov theory (see, e.g., [274, § 5.7]) in order to approximate from outside. This can be done in a continuous-time setting (with possible extension to discrete-time systems), i.e., when the state variable is constrained from an initial condition to satisfy an ordinary differential equation $\dot{\mathbf{x}} = f(\mathbf{x}, t)$. The idea is to search for a Lyapunov function $v$ (also called value or Bellman function in the context of optimal control) which is negative on the set of initial conditions and with negative derivative of states satisfying some general constraints. These inequalities provide sufficient conditions for the RS to be included in the sublevel set of $v$. In the case where the set of initial (resp. general) state constraints are defined by polynomial inequalities, the difficulty of computing such a function $v$ can be practically addressed. This is done while reducing the search space to polynomials of bounded degree and by replacing the inequalities satisfied by $v$ (and its total derivative) by stronger equality constraints involving $v$ and (weighted) SOS of polynomials. Since the weights are the polynomials defining the set of initial and general constraints, computing $v$ together with these SOS polynomials boils down to solving an SDP of fixed size. This general framework has been used in [242] for the safety verification of hybrid systems. In this case, the function $v$ is called a "barrier certificate" and can be constructed by computing an SOS decomposition. The zero level set of $v$ separates a given unsafe region from all possible trajectories starting from a prescribed set of initial conditions.
These dual Lyapunov certificates relying on SOS decompositions also allow to obtain approximations of the (backward) RS (also called region of attraction) [60]. In [294], the authors proved the existence of a Lyapunov function, whose sublevel set is the region of attraction of a given equilibrium point of a continuous-time system. When the degree of the approximation $v$ is fixed in advance, one can obtain convergence guarantees by increasing the degree of the SOS polynomials. However, one has no guarantee that when the degree of $v$ goes to infinity, the approximation conservatism asymptotically vanishes. In addition, the conservatism of such approximations relying on dual Lyapunov certificates is not easy to estimate in a systematic way.

Here, we propose a characterization of the RS as the solution of an infinite-dimensional LP problem. Most materials from this section have been published in [J19]. This characterization is done by considering a hierarchy of converging convex programs through moment relaxations of the LP. Doing so, one can compute tight outer approximations of the RS. Such outer approximations yield invariants for the discrete-time system, which are sets where systems trajectories are confined.

The general methodology is deeply inspired from previous research efforts. The idea of formulation relying on LP optimization over probability measures appears in [180], with a hierarchy of SDP also called the moment-SOS hierarchy, whose optimal values converge from below to the infimum of a multivariate polynomial. One can see outer approximations of sets as the analogue of lower approximations of real-valued functions. In [130], the authors leverage on these techniques to address the problem of computing outer approximations by single polynomial super level sets of basic compact semialgebraic sets described by the intersection of a finite number of given polynomial super level sets. Further work focused on approximating semialgebraic sets for which such a description is not explicitly known or difficult to compute: in [185], the author derives converging outer approximations of sets defined with existential quantifiers; in [J10], the authors approximate the image of a compact semialgebraic set under a polynomial map. The current study can be seen as an extension of [J10] where instead of considering only one iteration of the map, we consider infinitely many iterations starting from a set of initial conditions.

This methodology has also been successfully applied for several problems arising in the context of polynomial systems control. Similar convergent hierarchies appear in [129], where the authors approximate the region of attraction (ROA) of a controlled polynomial system subject to compact semialgebraic constraints in continuous time, or in [231] where the authors consider maximal positively invariant sets for continuous-time polynomial systems. This framework is extended to hybrid systems in [267]. Note that the ROA is not a semialgebraic set in general. The authors of [162] build upon the infinite-dimensional LP formulation of the ROA problem while providing a similar framework to characterize the maximum controlled invariant (MCI) for discrete and continuous time polynomial dynamical systems. The framework used for ROA and MCI computation both rely on occupation measures. These allow to measure the time spent by solutions of differential or difference equations. As solutions of a linear transport equation called the Liouville Equation, occupation measures also capture the evolution of the semialgebraic set describing the initial conditions. As mentioned in [129], the problem of characterizing the (forward) RS in a continuous setting and finite horizon could be done as ROA computation by using a time-reversal argument.

The modeling power of this approach also extends to the analysis of attractors of dynamical systems, e.g., by approximating the moments and support of invariant measures in both continuous and discrete-time settings [165, J8], see Section 1.3. In the present study, we handle the problem in a discrete setting and infinite horizon. Our contribution follows a similar approach but requires to describe the solution set of another Liouville Equation.

### Forward reachable sets and Liouville's Equation

For a given *terminal time* $T \in \mathbb{N}_{>0}$ and an initial measure $\mu_0 \in \mathscr{M}_+(\mathbf{X}^0)$, let us define the measures $\mu_1, \dots, \mu_T, \mu \in \mathscr{M}_+(\mathbf{X})$ as follows:

$$\begin{aligned} \mu_{t+1} &:= f_\# \mu_t = f_\#^{t+1} \mu_0 \,, \; t = 0, \dots, T-1 \,, \\ \nu &:= \sum_{t=0}^{T-1} \mu_t = \sum_{t=0}^{T-1} f_\#^t \mu_0 \,. \end{aligned} \tag{1.5}$$

The measure $\nu$ is a (discrete-time) occupation measure: if $\mu_0 = \delta_{\mathbf{x}_0}$ is the Dirac measure at $\mathbf{x}_0 \in \mathbf{X}^0$ then $\mu_t = \delta_{x_t}$ and $\nu = \delta_{x_0} + \delta_{x_1} + \cdots + \delta_{x_{T-1}}$ , i.e., $\nu$ measures the time spent by the state trajectory in any subset of $\mathbf{X}$ after $T$ iterations, if initialized at $\mathbf{x}_0$.

**Lemma 1.2.4** *See [J19] For any $T \in \mathbb{N}_{>0}$ and $\mu_0 \in \mathscr{M}(\mathbf{X}^0)$, there exist $\mu_T, \nu \in \mathscr{M}(\mathbf{X})$ solving the discrete Liouville Equation:*

$$\mu_T + \nu = f_\# \nu + \mu_0 \,. \tag{1.6}$$

Now let

$$\mathbf{Y}^0 := \mathbf{X}^0, \quad \mathbf{Y}^t := f^t(\mathbf{X}^0) \backslash \mathbf{X}^{t-1}, \; t = 1, \dots, T.$$

Note that the RS is equal to

$$\mathbf{X}^T = \cup_{t=0}^T \mathbf{Y}^t.$$

Further results involve statements relying on the following technical assumption:

**Assumption 1.2.5** $\lim_{T\to\infty} \sum_{t=0}^T t \operatorname{vol} \mathbf{Y}^t < \infty$.

This assumption seems to be strong or unjustified at that point. Moreover, if we do not know if the assumption is satisfied a priori, there is an a posteriori validation based on duality theory. Thanks to this validation procedure, we could check that the assumption was satisfied in all the examples we processed. Moreover, we were not able to find a discrete-time polynomial system violating this assumption.

In the sequel, we prove that equation (1.6) holds when $\mu_T = \lambda_{\mathbf{X}^T}$, the restriction of the Lebesgue measure over the RS. We rely on the auxiliary result from [J10, Lemma 4.1]:

**Lemma 1.2.6** *Let $\mathbf{S}, \mathbf{B} \subseteq \mathbf{X}$ be such that $f(\mathbf{S}) \subseteq \mathbf{B}$. Given a measure $\mu_1 \in \mathscr{M}_+(\mathbf{B})$, there is a measure $\mu_0 \in \mathscr{M}_+(\mathbf{S})$ such that $f_\# \mu_0 = \mu_1$ if and only if there is no continuous function $v \in \mathscr{C}(\mathbf{B})$ such that $v(f(\mathbf{x})) \geq 0$ for all $\mathbf{x} \in \mathbf{S}$ and $\int_{\mathbf{B}} v(\mathbf{y}) d\mu_1(\mathbf{y}) < 0$.*

**Lemma 1.2.7** *For any $T \in \mathbb{N}_{>0}$, there exist $\mu_0^T \in \mathscr{M}_+(\mathbf{X}^0)$ and $\nu^T \in \mathscr{M}_+(\mathbf{X})$ such that the restriction of the Lebesgue measure over $\mathbf{X}^T$ solves the discrete Liouville Equation:*

$$\lambda_{\mathbf{X}^T} + \nu^T = f_\# \nu^T + \mu_0^T \,. \tag{1.7}$$

*In addition, if Assumption 1.2.5 holds, then there exist $\mu_0 \in \mathscr{M}_+(\mathbf{X}^0)$ and $\nu \in \mathscr{M}_+(\mathbf{X})$ such that the restriction of the Lebesgue measure over $\mathbf{X}^\infty$ solves the discrete Liouville Equation:*

$$\lambda_{\mathbf{X}^\infty} + \nu = f_\# \nu + \mu_0 \,. \tag{1.8}$$

**Remark 1.2.2** *In Lemma 1.2.7, the measure $\mu_0^T$ (resp. $\mu_0$) can be thought as distribution of mass for the initial states of trajectories reaching $\mathbf{X}^T$ (resp. $\mathbf{X}^\infty$) but it has a total mass which is not required to be normalized to one.*

*The mass of $\nu^T$ measures the volume averaged w.r.t. $\mu_0$ occupied by state trajectories reaching $\mathbf{X}^T$ after $T$ iterations, by contrast with the mass of $\lambda_{\mathbf{X}^T}$ which measures the volume of $\mathbf{X}^T$.*

*The mass of $\nu$ measures the volume averaged w.r.t. $\mu_0$ occupied by state trajectories reaching the RS $\mathbf{X}^\infty$, by contrast with the mass of $\lambda_{\mathbf{X}^\infty}$ which measures the exact RS volume.*

## Primal-dual LP formulation

To approximate the set $\mathbf{X}^\infty$, one considers the infinite-dimensional LP, for any $T \in \mathbb{N}_{>0}$:

$$\begin{aligned} p^T := \sup_{\mu_0,\mu,\hat{\mu},\nu,a} \quad & \int_{\mathbf{X}} \mu \\ \text{s.t.} \quad & \int_{\mathbf{X}} \nu + a = T \operatorname{vol} \mathbf{X} \,, \\ & \mu + \nu = f_\# \nu + \mu_0 \,, \\ & \mu + \hat{\mu} = \lambda_{\mathbf{X}} \,, \\ & \mu_0 \in \mathscr{M}_+(\mathbf{X}^0) \,, \quad \mu, \hat{\mu}, \nu \in \mathscr{M}_+(\mathbf{X}) \,, \quad a \in \mathbb{R}_+ \,. \end{aligned} \tag{1.9}$$

The first equality constraint ensures that the mass of the occupation measure $\nu$ is bounded (by $T \operatorname{vol} \mathbf{X}$). The second one ensures that Liouville's equation is satisfied by the measures $\mu_0$, $\nu$ and $\mu$, as in Lemma 1.2.7. The last one ensures that $\mu$ is dominated by the restriction of the Lebesgue measure on $\mathbf{X}$ implying that the mass of $\mu$ (and thus the optimal value $p^T$) is bounded by $\operatorname{vol} \mathbf{X}$. The next result explains how the solution of LP (1.9) relates to $\lambda_{\mathbf{X}^\infty}$, the restriction of the Lebesgue measure to the RS.

**Theorem 1.2.1** *For any $T \in \mathbb{N}_{>0}$, LP (1.9) has an optimal solution $(\mu_0^*, \mu^*, \hat{\mu}^*, \nu^*, a^*)$ such that $\mu^* = \lambda_{\mathbf{S}^T}$ for some set $\mathbf{S}^T$ satisfying $\mathbf{X}^T \subseteq \mathbf{S}^T \subseteq \bar{\mathbf{X}}^\infty$ and $\operatorname{vol} \mathbf{S}^T = p^T$.*

*In addition if Assumption 1.2.5 holds then there exists $T_0 \in \mathbb{N}$ such that for all $T \geq T_0$ one has $\mathbf{S}^T = \bar{\mathbf{X}}^\infty$, LP (1.9) has a unique optimal solution with $\mu^* = \lambda_{\mathbf{X}^\infty}$ and $p^T = \operatorname{vol} \mathbf{X}^\infty$.*

From now on, we refer to $\mathbf{S}^T$ as the support of the optimal solution $\mu^*$ of LP (1.9) which satisfies the condition of Lemma 1.2.1, i.e., $\mathbf{X}^T \subseteq \mathbf{S}^T \subseteq \bar{\mathbf{X}}^\infty$.

In the sequel, we formulate LP (1.9) as an infinite-dimensional conic problem on appropriate vector spaces. By construction, a feasible solution of problem (1.9) satisfies:

$$\int_{\mathbf{X}} \nu(d\mathbf{x}) + a = \int_{\mathbf{X}} T\lambda(d\mathbf{x}) \,, \tag{1.10}$$

$$\int_{\mathbf{X}} v(\mathbf{x})\, \mu(d\mathbf{x}) + \int_{\mathbf{X}} v(\mathbf{x})\, \nu(d\mathbf{x}) = \int_{\mathbf{X}} v(f(\mathbf{x}))\, \nu(d\mathbf{x}) + \int_{\mathbf{X}^0} v(\mathbf{x})\, \mu_0(d\mathbf{x}) \,, \tag{1.11}$$

$$\int_{\mathbf{X}} w(\mathbf{x})\, \mu(d\mathbf{x}) + \int_{\mathbf{X}} w(\mathbf{x})\, \hat{\mu}(d\mathbf{x}) = \int_{\mathbf{X}} w(\mathbf{x})\, \lambda(d\mathbf{x}) \,, \tag{1.12}$$

for all continuous test functions $v, w \in \mathscr{C}(\mathbf{X})$.

Then, we cast problem (1.9) as a particular instance of a primal LP in the canonical form given in [27, p. 7.1.1] and consider its dual. With our notation, the dual LP reads:

$$\begin{aligned}
d^T := \inf_{u,v,w} \quad & \int_{\mathbf{X}} (w(\mathbf{x}) + Tu)\, \lambda_{\mathbf{X}}(d\mathbf{x}) \\
\text{s.t.} \quad & v(\mathbf{x}) \geq 0, \quad \forall \mathbf{x} \in \mathbf{X}^0, \\
& w(\mathbf{x}) \geq 1 + v(\mathbf{x}), \quad \forall \mathbf{x} \in \mathbf{X}, \\
& w(\mathbf{x}) \geq 0, \quad \forall \mathbf{x} \in \mathbf{X}, \\
& u + v(f(\mathbf{x})) \geq v(\mathbf{x}), \quad \forall \mathbf{x} \in \mathbf{X}, \\
& u \geq 0, \\
& u \in \mathbb{R}, \quad v, w \in \mathscr{C}(\mathbf{X}).
\end{aligned} \tag{1.13}$$

**Theorem 1.2.2** *For a fixed $T \in \mathbb{N}_{>0}$, there is no duality gap between primal LP (1.9) and dual LP (1.13), i.e., $p^T = d^T$ and there exists a minimizing sequence $(u_k, v_k, w_k)_{k \in \mathbb{N}}$ for the dual LP (1.13). In addition, if $u_k = 0$ for some $k \in \mathbb{N}$, then Assumption 1.2.5 holds and $p^T = d^T = \operatorname{vol} \mathbf{X}^\infty$.*

**Remark 1.2.3** *When $u = 0$, the first and third inequalities satisfied by $-v$ in dual LP (1.13) can be seen as a discrete-time analogue of the conditions satisfied by the barrier certificate in [242], that is $-v \leq 0$ on $\mathbf{X}^0$ and $-v \circ f \leq -v$ on $\mathbf{X}$, the latter one being similar to the barrier condition $\nabla(-v) \cdot f \leq -v$.*

In the sequel, we propose a method to compute an outer approximation of $\mathbf{X}^\infty$ as the super level set of a polynomial. Given its degree $2r$, such a polynomial may be produced using standard numerical optimization tools. Under certain assumptions, the resulting outer approximation can be as tight as desired and converges (with respect to the $\mathscr{L}_1$ norm on $\mathbf{X}$) to the set $\mathbf{X}^\infty$ as $r$ tends to infinity.

## Primal-dual hierarchies of SDP approximations

Given a positive $m \in \mathbb{N}$, let us note $[m] = \{1, \ldots, m\}$. With $\mathbf{X}^0$ a basic compact semialgebraic set as in 1.3, let us define $r_j^0 := \lceil (\deg g_j^0)/2 \rceil$, for all $j \in [m^0]$, and with $\mathbf{X}$ a basic compact semialgebraic set as in (1.1), we set $r_j := \lceil (\deg g_j)/2 \rceil, j \in [m]$. For each $r \geq r_{\min} := \max\{r_1^0, \ldots, r_{m^0}^0, r_1, \ldots, r_m\}$, let $\mathbf{y}_0 = (y_{0\beta})_{\beta \in \mathbb{N}_{2r}^n}$ be the finite sequence of moments up to degree $2r$ of the measure $\mu_0$. Similarly, let $\mathbf{y}$, $\hat{\mathbf{y}}$ and $\mathbf{z}$ stand for the sequences of moments up to degree $2r$, respectively associated with $\mu$,

$\hat{\mu}$ and $\nu$. The infinite primal LP (1.9) can be relaxed with the following SDP:

$$\begin{aligned}
p_r^T := \sup_{\mathbf{y}_0,\mathbf{y},\hat{\mathbf{y}},\mathbf{z},a} \quad & y_0 \\
\text{s.t.} \quad & z_0 + a = Ty_0^{\mathbf{X}}\,, \\
& y_\beta + z_\beta = L_{\mathbf{z}}(f(\mathbf{x})^\beta) + y_{0\beta}\,, \quad \forall \beta \in N_{2r}^n\,, \\
& y_\beta + \hat{y}_\beta = y_\beta^{\mathbf{X}}, \quad \forall \beta \in N_{2r}^n\,, \\
& \mathbf{M}_{rd-r_j^0}(g_j^0\,\mathbf{y}_0) \succeq 0, \quad j = 0,\ldots,m^0\,, \\
& \mathbf{M}_{r-r_j}(g_j\,\mathbf{y}) \succeq 0\,, \mathbf{M}_{r-r_j}(g_j\,\hat{\mathbf{y}}) \succeq 0\,, \mathbf{M}_{r-r_j}(g_j\,\mathbf{z}) \succeq 0\,, j = 0,\ldots,m\,, \\
& a \geq 0\,.
\end{aligned} \tag{1.14}$$

Consider also the following SDP, which is a strengthening of the infinite dual LP (1.13) and also the dual of Problem (1.14):

$$\begin{aligned}
d_r^T := \inf_{u,v,w} \quad & \sum_{\beta\in\mathbb{N}_{2r}^n} w_\beta y_\beta^{\mathbf{X}} + uTy_0^{\mathbf{X}} \\
\text{s.t.} \quad & v \in \mathcal{M}(\mathbf{X}^0)_r\,, \\
& w - 1 - v \in \mathcal{M}(\mathbf{X})_r\,, \\
& u + v \circ f - v \in \mathcal{M}_{rd}(\mathbf{X})\,, \\
& w \in \mathcal{M}(\mathbf{X})_r\,, \\
& u \in \mathbb{R}^+\,, \\
& v, w \in \mathbb{R}[\mathbf{x}]_{2r}\,,
\end{aligned} \tag{1.15}$$

where $\mathcal{M}(\mathbf{X}^0)_r$, $\mathcal{M}(\mathbf{X})_r$ (resp. $\mathcal{M}(\mathbf{X})_{rd}$) are the $r$-truncated (resp. $rd$) quadratic module respectively generated by $g_0^0,\ldots,g_m^{m^0}$ and $g_0,\ldots,g_m$, as defined in Section 1.1.

**Theorem 1.2.3** *Let $r \geq r_{\min}$. Suppose that the three sets $\mathbf{X}^0$, $\mathbf{S}^T$ and $\mathbf{X}\backslash\mathbf{S}^T$ have nonempty interior. Then:*

1. *$p_r^T = d_r^T$, i.e., there is no duality gap between the primal SDP program (1.14) and the dual SDP program (1.15).*

2. *The dual SDP program (1.15) has an optimal solution $(u_r, v_r, w_r) \in \mathbb{R} \times \mathbb{R}[\mathbf{x}]_{2r} \times \mathbb{R}[\mathbf{x}]_{2r}$, and the sequence $(w_r + u_r T)$ converges to $\mathbf{1}_{\mathbf{S}^T}$ in $\mathscr{L}_1$ norm on $\mathbf{X}$:*

$$\lim_{r\to\infty} \int |w_r(\mathbf{x}) + u_r T - \mathbf{1}_{\mathbf{S}^T}(\mathbf{x})|\, \lambda_{\mathbf{X}}(d\mathbf{x}) = 0. \tag{1.16}$$

3. *Defining the sets*

$$\mathbf{X}_r^T := \{\mathbf{x} \in \mathbf{X} : v_r(\mathbf{x}) + u_r T \geq 0\}\,,$$

*it holds that*

$$\mathbf{X}_r^T \supseteq \mathbf{X}^T\,.$$

4. *In addition, if $u_r = 0$ then the sequence $(w_r)$ converges to $\mathbf{1}_{\bar{\mathbf{X}}^\infty}$ in $\mathscr{L}_1$ norm on $\mathbf{X}$. Defining the sets*

$$\mathbf{X}_r^\infty := \{\mathbf{x} \in \mathbf{X} : v_r(\mathbf{x}) \geq 0\}\,,$$

*its holds that*

$$\mathbf{X}_r^\infty \supseteq \bar{\mathbf{X}}^\infty \supseteq \mathbf{X}^\infty .$$

*and*

$$\lim_{r\to\infty} \operatorname{vol}(\mathbf{X}_r^\infty \backslash \mathbf{X}^\infty) = \operatorname{vol}(\mathbf{X}_r^\infty \backslash \bar{\mathbf{X}}^\infty) = 0 .$$

**Remark 1.2.4** *Theorem 1.2.3 states that one can over approximate the reachable states of the system after any arbitrary finite number of discrete-time steps (third item). In addition, Theorem 1.2.3 provides a sufficient condition to obtain a hierarchy of over approximations converging in volume to the RS (fourth item). If $u_r = 0$, then the sequence of optimal values of SDP (1.15) is nonincreasing and converges to the volume of the RS. If one defines the piecewise polynomial $\overline{v}_r := \min_{k\leq r} v_k$, then one shows as in [135, Theorem 1] that we obtain a nonincreasing sequence of functions converging to the indicator function of the RS: one has $\overline{v}_r \downarrow \mathbf{1}_{\overline{\mathbf{X}}^\infty}$ almost everywhere, almost uniformly and in Lebesgue measure.*

## Special case: linear systems with ellipsoid constraints

Given $A \in \mathbb{R}^{n\times n}$, let us consider a discrete-time linear system $\mathbf{x}_{t+1} = \mathbf{A}\,\mathbf{x}_t$ with a set of initial constraints defined by the ellipsoid $\mathbf{X}^0 := \{\mathbf{x} \in \mathbb{R}^n : 1 \geq \mathbf{x}^T\,\mathbf{V}_0\,\mathbf{x}\}$ with $\mathbf{V}_0 \in \mathbb{R}^{n\times n}$ a positive definite matrix.

Similarly the set of state constraints is defined by the ellipsoid $\mathbf{X} = \{\mathbf{x} \in \mathbb{R}^n : 1 \geq \mathbf{x}^T\,\mathbf{G}\,\mathbf{x}\}$ with $\mathbf{G} \in \mathbb{R}^{n\times n}$ a positive definite matrix. Since one has $\mathbf{X}^0 \subseteq \mathbf{X}$, it follows that $\mathbf{V}_0 \succeq \mathbf{G}$.

Then, one can look for a quadratic function $v(\mathbf{x}) := 1 - \mathbf{x}^T\,\mathbf{V}\,\mathbf{x}$, with $\mathbf{V} \in \mathbb{R}^{n\times n}$ a positive definite matrix solution of the following SDP optimization problem:

$$\begin{aligned} \sup_{\mathbf{V}\in\mathbb{R}^{n\times n}} \quad & \langle \mathbf{M}\mathbf{V} \rangle \\ \text{s.t.} \quad & \mathbf{V}_0 \succeq \mathbf{V} \succeq \mathbf{A}^T\mathbf{V}\mathbf{A}\,, \\ & \mathbf{V} \succ 0\,, \end{aligned} \tag{1.17}$$

where $\langle\cdot\rangle$ stands for the matrix trace, and $\mathbf{M}$ is the second-order moment matrix of the Lebesgue measure on $\mathbf{X}$, i.e., the matrix with entries

$$(\mathbf{M})_{\alpha,\beta} = y^{\mathbf{X}}_{\alpha+\beta}, \quad ,\alpha,\beta \in \mathbb{N}^n, |\alpha| + |\beta| = 2.$$

Note that in this special case SDP (1.17) can be retrieved from SDP (1.15) and one can over approximate the RS with the superlevel set of $v$ or $w - 1$:

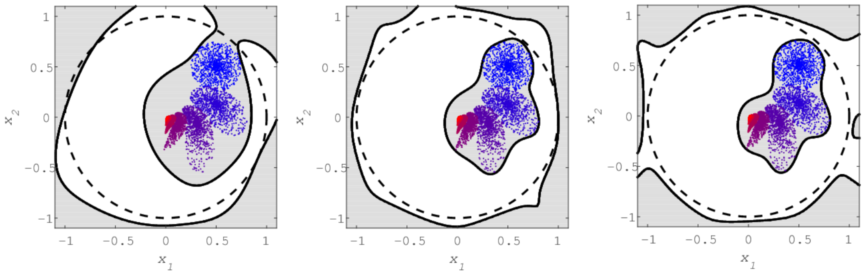
**Lemma 1.2.8** *SDP (1.17) is equivalent to SDP (1.15) with $r := 1$, $u_r := 0$, $v(\mathbf{x}) := 1 - \mathbf{x}^T\,\mathbf{V}\,\mathbf{x}$ and $w(\mathbf{x}) = 1 + v(\mathbf{x})$. Thus, one has:*

$$\{\mathbf{x} \in \mathbf{X} : v(\mathbf{x}) \geq 0\} = \{\mathbf{x} \in \mathbf{X} : w(\mathbf{x}) \geq 1\} \supseteq \mathbf{X}^\infty .$$

## Numerical experiments

Here, we present experimental benchmarks that illustrate our method. For a given positive integer $r$, we compute the polynomial solution $w_r$ of the dual SDP program (1.15). This dual SDP is modeled using the YALMIP toolbox [200] available within MATLAB and interfaced with the SDP solver MOSEK [7]. Performance results were obtained with an Intel Core i7-5600U CPU (2.60 GHz) running under Debian 8.

For all experiments, we could find an optimal solution of the dual SDP program (1.15) either by adding the constraint $u = 0$ or by setting $T = 100$. In the latter case, the optimal solution is such that $u_r \simeq 0$ and the polynomial solution $w_r$ is the same than in the former case, up to small numerical errors (in practice the value of $u_r$ is less than 1e–5). This implies that Assumption 1.2.5 is satisfied, i.e., the constraint of the mass of the occupation measure is not saturated, and yielding valid outer approximations of $\mathbf{X}^\infty$. The implementation is freely available on-line[1]. We first consider the toy example described at the beginning of this section. On Figure 1.2, we represent in light gray the outer approximations $\mathbf{X}_r^\infty$ of $\mathbf{X}^\infty$ obtained by our method, for increasing values of the relaxation order $r$ (from $r = 3$ to 7). Figure 1.2 shows that the over approximations are already quite tight for low degrees.



(a) $r = 3$ (b) $r = 5$ (c) $r = 7$

Figure 1.2: Outer approximations $\mathbf{X}_r^\infty$ (light gray) of $\mathbf{X}^\infty$ (color dot samples) for the toy example, $r \in \{3, 5, 7\}$.

Next, we consider the discretized version (taken from [33, Section 5]) of the FitzHugh-Nagumo model [90], which is originally a continuous-time polynomial system modelling the electrical activity of a neuron:

$$
\begin{aligned}
x_1^+ &:= x_1 + 0.2(x_1 - x_1^3/3 - x_2 + 0.875)\,,\\
x_2^+ &:= x_2 + 0.2(0.08(x_1 + 0.7 - 0.8x_2))\,,
\end{aligned}
$$

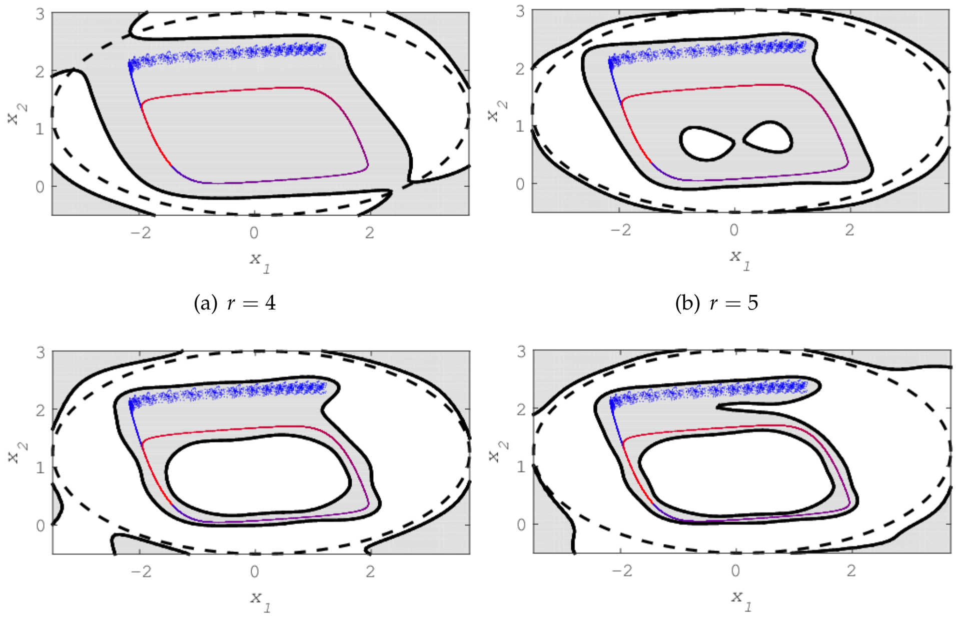
with initial state constraints $\mathbf{X}^0 := [1, 1.25] \times [2.25, 2.5]$ and state constraints $\mathbf{X} = \{\mathbf{x} \in \mathbb{R}^2 : (\frac{x_1 - 0.1}{3.6})^2 + (\frac{x_2 - 1.25}{1.75})^2 \leq 1\}$. Figure 1.3 illustrates that the outer approximations provide useful indications on the system behavior, in particular for higher values of $r$. Indeed $\mathbf{X}_5^\infty$ and $\mathbf{X}_6^\infty$ capture the presence of the central "hole" made by periodic trajectories and $\mathbf{X}_7^\infty$ shows that there is a gap between the first discrete-time steps and the iterations corresponding to these periodic trajectories.

Lastly, we consider the discretized version of the Phytoplankton growth model (also taken from [33, Section 5]). This model is obtained after making assumptions, corroborated experimentally by biologists in order to represent such growth phenomena [37], yielding the following discrete-time polynomial system:

$$
\begin{aligned}
x_1^+ &:= x_1 + 0.01(1 - x_1 - 0.25x_1x_2)\,,\\
x_2^+ &:= x_2 + 0.01(2x_3 - 1)x_2\,,\\
x_3^+ &:= x_3 + 0.01(0.25x_1 - 2x_3^2)\,,
\end{aligned}
$$

with initial state constraints $\mathbf{X}^0 := [-0.3, -0.2]^2 \times [-0.05, 0.05]$ and state constraints $\mathbf{X} = [-0.5, 1.5] \times [-0.5, 0.5]^2$. Figure 1.4 illustrates the system convergence behavior towards an equilibrium point for initial conditions near the origin. One way to obtain more accurate information

[1] https://homepages.laas.fr/vmagron/files/reachsdp.tar.gz



(a) $r = 4$

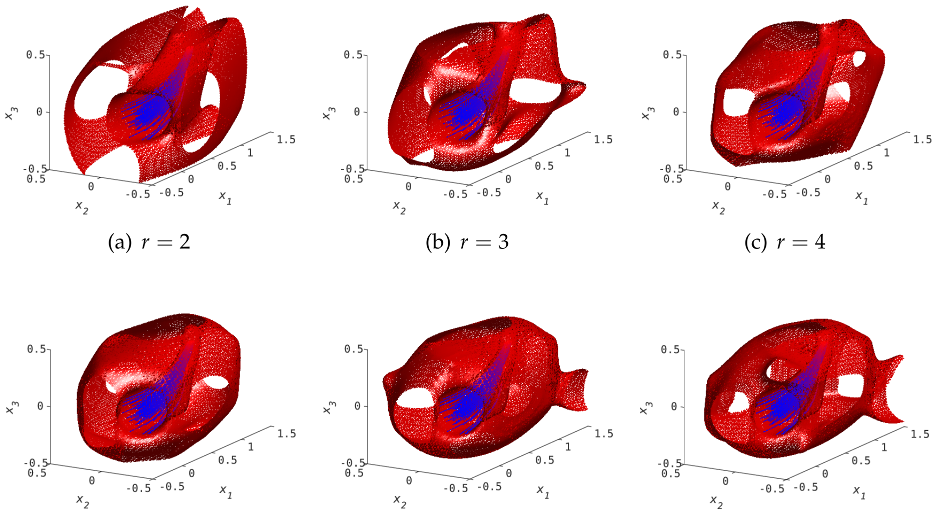
(b) $r = 5$

(c) $r = 6$

(d) $r = 7$

Figure 1.3: Outer approximations $\mathbf{X}_r^\infty$ (light gray) of $\mathbf{X}^\infty$ (color dot samples) for the FitzHugh-Nagumo model, $r \in \{4, 5, 6, 7\}$.

on such systems would be to design a subdivision procedure (e.g. with branch-and-bound techniques), which boils down to zooming on specific areas of the RS.



(a) $r = 2$

(b) $r = 3$

(c) $r = 4$

(d) $r = 5$

(e) $r = 6$

(f) $r = 7$

Figure 1.4: Outer approximations $\mathbf{X}_r^\infty$ (red) of $\mathbf{X}^\infty$ (color dot samples) for the Phytoplankton growth model, from $r = 2$ to 7.

## 1.3 Invariant measures for polynomial systems

Given a polynomial system described by a discrete-time (difference) or continuous-time (differential) equation under general semialgebraic constraints, we propose numerical methods to approximate the moments and the supports of the measures which are invariant under the sytem dynamics.

The characterization of invariant measures allows to determine important features of long term dynamical behaviors [172].

We develop our approach in parallel for discrete-time and continuous-time systems. As in Section 1.2, we have a polynomial transition map $f : \mathbb{R}^n \to \mathbb{R}^n, \quad \mathbf{x} \mapsto f(\mathbf{x}) := (f_1(\mathbf{x}), \ldots, f_n(\mathbf{x})) \in \mathbb{R}^n[\mathbf{x}]$ and we have a set $\mathbf{X}$ of basic compact semialgebraic state constraints as in (1.1). As in Section 1.2, we assume that $\mathbf{X}$ satisfies both Assumptions 1.1.1 (Archimedean quadratic module) and 1.2.3 (available moments of the Lebesgue measure). We consider either the discrete-time system:

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t)\,, \quad \mathbf{x}_t \in \mathbf{X}\,, \quad t \in \mathbb{N}\,, \tag{1.18}$$

or the continuous-time system:

$$\dot{\mathbf{x}}(t) = \frac{d\mathbf{x}(t)}{dt} = f(\mathbf{x}(t))\,, \quad \mathbf{x}(t) \in \mathbf{X}\,, \quad t \in [0, \infty)\,. \tag{1.19}$$

Let us define the linear operator $\mathcal{L}_f^{\text{disc}} : \mathscr{C}(\mathbf{X})' \to \mathscr{C}(\mathbf{X})'$ by:

$$\mathcal{L}_f^{\text{disc}}(\mu) := f_{\#}\mu - \mu$$

and the linear operator $\mathcal{L}_f^{\text{cont}} : \mathscr{C}^1(\mathbf{X})' \to \mathscr{C}(\mathbf{X})'$ by:

$$\mathcal{L}_f^{\text{cont}}(\mu) := \operatorname{div}(f\mu) = \sum_{i=1}^n \frac{\partial(f_i\mu)}{\partial x_i}$$

where the derivatives of measures are understood in the sense of distributions, that is, through their action on test functions of $\mathscr{C}^1(\mathbf{X})$. In the sequel, we use the more concise notation $\mathcal{L}_f$ to refer to $\mathcal{L}_f^{\text{disc}}$ (resp. $\mathcal{L}_f^{\text{cont}}$) in the context of discrete-time (resp. continuous-time) systems.

**Definition 1.3.1** ***(Invariant measure)*** *We say that a measure $\mu$ is invariant w.r.t. $f$ when $\mathcal{L}_f(\mu) = 0$ and refer to such a measure as an invariant measure. We also omit the reference to the map $f$ when it is obvious from the context and write $\mathcal{L}(\mu) = 0$.*

When considering discrete-time systems as in (1.18), a measure $\mu$ is called invariant w.r.t. $f$ when it satisfies $\mathcal{L}_f^{\text{disc}}(\mu) = 0$. When considering continuous-time systems as in (1.19), a measure is called invariant w.r.t. $f$ when it satisfies $\mathcal{L}_f^{\text{cont}}(\mu) = 0$.

It was proved in [166] that a continuous map of a compact metric space into itself has at least one invariant probability measure. A probability measure $\mu$ is ergodic w.r.t. $f$ if for all $\mathbf{A} \in \mathcal{B}(\mathbf{X})$ with $f^{-1}(\mathbf{A}) = \mathbf{A}$, one has either $\mu(\mathbf{A}) = 0$ or $\mu(\mathbf{A}) = 1$. The set of invariant probability measures is a convex set and the extreme points of this set consist of the so-called ergodic measures. For more material on dynamical systems and invariant measures, we refer the interested reader to [172].

### Classical approaches

One classical way to approximate such features is to perform numerical integration of the equation satisfied by the system state after choosing some initial conditions. However, the resulting trajectory could exhibit some chaotic behaviors or great sensitivity with respect to the initial conditions.

Numerical computation of invariant sets and measures of dynamical systems have previously been studied using domain subdivision techniques, where the density of an invariant measure is recovered as the solution of fixed point equations of the discretized Perron-Frobenius operator [76, 14]. The underlying method, integrated in the software package GAIO [74], consists of covering the invariant set by boxes and then approximating the dynamical behaviour by a Markov chain based on transition probabilities between elements of this covering. More recently, in [75] the authors have developed a multilevel subdivision scheme that can handle uncertain ordinary differential equations as well.

By contrast with most of the existing work in the literature, our method does not rely neither on time nor on space discretization. In our approach, the invariant measures are modeled with finitely many moments, leading to approximate recovery of densities in the absolutely continuous case or supports in the singular case. Our contribution is in the research trend aiming at characterizing the behavior of dynamical nonlinear systems through LP, whose unkown are measures supported on the system constraints.

In our case, we first focus on the characterization of densities of absolutely continuous invariant measures with respect to some reference measure (for instance the Lebesgue measure). For this first problem, our method is inspired by previous contributions looking for moment conditions ensuring that the underlying unknown measure is absolutely continuous [177] with a bounded density in a Lebesgue space, as a follow-up of the volume approximation results of [130]. When the density function is assumed to be square-integrable, one can rely on [133] to build a sequence of polynomial approximations converging to this density in the $\mathscr{L}^2$-norm.

We focus later on the characterization of supports of singular invariant measures. For this second problem, we rely on previous works [130, 181, 187] aiming at extracting as much information as possible on the support of a measure from the knowledge of its moments. The numerical scheme proposed in [130] allows to approximate as close as desired the moments of a measure uniformly supported on a given semialgebraic set. The framework from [181] uses similar techniques to compute effectively the Lebesgue decomposition of a given measure, while [187] relies on the Christoffel function associated to the moment matrix of this measure. When the measure is uniform or when the support of the measure satisfies certain conditions, the sequence of level sets of the Christoffel function converges to the measure support with respect to the Hausdorff distance.

Previous work [128] shows how to use the Lasserre hierarchy to characterize invariant measures for one-dimensional discrete polynomial dynamical systems. We extend significantly this work in the sense that we now characterize invariant measures on more general multidimensional semialgebraic sets, in both discrete and continuous settings, and we establish convergence guarantees under certain assumptions. In the concurrent work [165], the authors are also using the Lasserre hiearchy for approximately computing extremal measures, i.e., invariant measures optimal w.r.t. a convex criterion. They have weaker convergence guarantees than ours, but the problem is formulated in a more general setting allowing to use the criterion to single-out some class of extremal measures, including physical measures, ergodic measures or atomic measures (for instance periodic orbits) or invariant densities.

## A first observation

For a given invariant measure $\mu \in \mathscr{M}(\mathbf{X})$, one has $\mathcal{L}(\mu) = 0$. It follows from the Stone-Weierstrass Theorem that monomials are dense in continuous functions on the compact set $\mathbf{X}$. The equation $\mathcal{L}(\mu) = 0$ is then equivalent to

$$L_{\mathbf{y}}(f(\mathbf{x})^\alpha) - L_{\mathbf{y}}(\mathbf{x}^\alpha) = 0\,, \quad \forall \alpha \in \mathbb{N}^n\,,$$

in the context of a discrete-time system (1.18) and

$$\sum_{i=1}^{n} L_{\mathbf{y}}\left(\frac{\partial(\mathbf{x}^{\alpha})}{\partial x_i} f_i(\mathbf{x})\right) = 0\,, \quad \forall \alpha \in \mathbb{N}^n\,,$$

in the context of a continuous-time system (1.19).

Hence, we introduce the linear functionals $\mathscr{I}_{\mathbf{y}}^{\text{disc}} : \mathbb{R}[\mathbf{x}] \to \mathbb{R}$ defined by

$$\mathscr{I}_{\mathbf{y}}^{\text{disc}}(g) := L_{\mathbf{y}}(g \circ f) - L_{\mathbf{y}}(g)$$

and $\mathscr{I}_{\mathbf{y}}^{\text{cont}} : \mathbb{R}[\mathbf{x}] \to \mathbb{R}$ defined by

$$\mathscr{I}_{\mathbf{y}}^{\text{cont}}(g) := L_{\mathbf{y}}(\operatorname{grad} g \cdot f)$$

for every polynomial $g$ and where $\operatorname{grad} g := (\frac{\partial g}{\partial x_i})_{i=1,\ldots,n}$. In the sequel, we use the more concise notation $\mathscr{I}_{\mathbf{y}}$ to refer to $\mathscr{I}_{\mathbf{y}}^{\text{disc}}$ (resp. $\mathscr{I}_{\mathbf{y}}^{\text{cont}}$) in the context of discrete-time (resp. continuous-time) systems.

## Absolutely continuous invariant measures

For $p \in \mathbb{N} \cup \{\infty\}$, let $\mathscr{L}^p(\mathbf{X})$ (resp. $\mathscr{L}_+^p(\mathbf{X})$) be the space of (resp. nonnegative) Lebesgue integrable functions $f$ on $\mathbf{X}$, i.e., such that $\|f\|_p := (\int_{\mathbf{X}} |f(\mathbf{x})|^p \lambda(d\mathbf{x}))^{1/p} < \infty$. Let $\mathscr{L}^\infty(\mathbf{X})$ (resp. $\mathscr{L}_+^\infty(\mathbf{X})$) be the space of (resp. nonnegative) Lebesgue integrable functions $f$ on $\mathbf{X}$ which are essentially bounded on $\mathbf{X}$, i.e., such that $\|f\|_\infty := \operatorname{ess\,sup}_{x \in \mathbf{X}} |f(x)| < \infty$. Two integers $p$ and $q$ are said to be conjugate if $1/p + 1/q = 1$, and by Riesz's representation theorem (see, e.g., [196, Theorem 2.14]), the dual space of $\mathscr{L}^q(\mathbf{X})$ for $1 \leq q < \infty$ (i.e., the set of continuous linear functionals on $\mathscr{L}^q(\mathbf{X})$) is isometrically isomorphic to $\mathscr{L}^p(\mathbf{X})$.

For $\mu \in \mathscr{M}(\mathbf{X})$, if $\mu \ll \lambda$ then there exists a measurable function $h$ on $\mathbf{X}$ such that $d\mu = h\,d\lambda$ and the function $h$ is called the density of $\mu$. If $h \in \mathscr{L}^p(\mathbf{X})$, by a slight abuse of notation, we write $\mu \in \mathscr{L}^p(\mathbf{X})$ and $\|\mu\|_p := \|h\|_p$. If in addition $\mu$ is invariant w.r.t. $f$ then we say that $f$ has an invariant density in $\mathscr{L}^p(\mathbf{X})$.

Next, we state some conditions fulfilled by the moments of an absolutely continuous measure with a density in $\mathscr{L}^p(\mathbf{X})$. In the case of invariant measures, we rely on these conditions to provide an infinite-dimensional LP characterization. We show how to approximate the solution of this LP by using a hierarchy of finite-dimensional SDP relaxations. We also explain how to approximate the invariant density.

**Theorem 1.3.1** *Let $p$ and $q$ be conjugate with $1 \leq q < \infty$. Consider a sequence $\mathbf{y} \subset \mathbb{R}$. The following statements are equivalent:*

*(i) $\mathbf{y}$ has a representing measure $\mu \in \mathscr{L}_+^p(\mathbf{X})$ with $\|\mu\|_p \leq \gamma < \infty$ for some $\gamma \geq 0$;*

*(ii) there exists $\gamma \geq 0$ such that for all $g \in \mathbb{R}[\mathbf{x}]$ it holds*

$$|L_{\mathbf{y}}(g)| \leq \gamma \left(\int_{\mathbf{X}} |g|^q d\lambda\right)^{1/q}\,, \tag{1.20}$$

*and for all polynomial $g$ nonnegative on $\mathbf{X}$, it holds $L_{\mathbf{y}}(g) \geq 0$.*

Theorem 1.3.1 provides necessary and sufficient conditions satisfied by the moments of an absolutely continuous Borel measure with a density in $\mathscr{L}^p_+(\mathbf{X})$. We now state further characterizations when $p = q = 2$ in Theorem 1.3.2 and when $p = \infty$ and $q = 1$ in Theorem 1.3.3. For a given sequence $\mathbf{y} = (y_\alpha)_\alpha$ and $r \in \mathbb{N}$, the notation $\mathbf{y}^r$ stands for the truncated sequence $(y_\alpha)_{|\alpha| \leq 2r}$. The notation $\succeq 0$ means positive semidefinite. Recall that $\mathbf{y}^{\mathbf{X}}$ stands for the sequence of Lebesgue moments on $\mathbf{X}$, as defined after (1.2) in Section 1.1.

**Theorem 1.3.2** *Consider a sequence* $\mathbf{y} \subset \mathbb{R}$. *The following statements are equivalent:*

*(i)* $\mathbf{y}$ *has a representing measure* $\mu \in \mathscr{L}^2_+(\mathbf{X})$ *with* $\|\mu\|_2 \leq \gamma < \infty$ *for some* $\gamma \geq 0$*;*

*(ii) there exists* $\gamma \geq 0$ *such that for all* $r \in \mathbb{N}$*:*

$$\begin{pmatrix} \mathbf{M}_r(\mathbf{y}^{\mathbf{X}}) & \mathbf{y}^r \\ (\mathbf{y}^r)^T & \gamma^2 \end{pmatrix} \succeq 0 \tag{1.21}$$

*and*

$$\mathbf{M}_{r-r_j}(g_j \mathbf{y}) \succeq 0\,, \quad j = 0, \ldots, m\,. \tag{1.22}$$

**Theorem 1.3.3** *Consider a sequence* $\mathbf{y} \subset \mathbb{R}$. *The following statements are equivalent:*

*(i)* $\mathbf{y}$ *has a representing measure* $\mu \in \mathscr{L}^\infty_+(\mathbf{X})$ *with* $\|\mu\|_\infty \leq \gamma$ *for some* $\gamma \geq 0$*;*

*(ii) there exists* $\gamma \geq 0$ *such that for all* $r \in \mathbb{N}$*:*

$$\gamma \mathbf{M}_r(\mathbf{y}^{\mathbf{X}}) \succeq \mathbf{M}_r(\mathbf{y}) \tag{1.23}$$

$$\mathbf{M}_{r-r_j}(g_j \mathbf{y}) \succeq 0\,, \quad j = 0, 1, \ldots, m\,. \tag{1.24}$$

From now on, we will restrict to the case where $p = 2$ or $p = \infty$ while relying on one of the two characterizations stated in the two previous theorems. Let us consider the following infinite-dimensional conic program:

$$\begin{aligned} \rho_{\mathrm{ac}} := \sup_{\mu} \quad & \int_{\mathbf{X}} \mu \\ \text{s.t.} \quad & \mathcal{L}(\mu) = 0\,, \\ & \|\mu\|_p \leq 1\,, \\ & \mu \in \mathscr{L}^p_+(\mathbf{X})\,. \end{aligned} \tag{1.25}$$

**Theorem 1.3.4** *Problem (1.25) admits an optimal solution. If the optimal value* $\rho_{ac}$ *is positive, then the optimal solution is a nonzero invariant measure.*

**Assumption 1.3.2** *There exists a unique invariant probability measure $\mu_{ac} \in \mathscr{L}^p(\mathbf{X})$ for some $p \geq 1$.*

Note that Assumption 1.3.2 is equivalent to supposing that there exists a unique ergodic probability measure.

**Theorem 1.3.5** *If Assumption 1.3.2 holds, then problem (1.25) admits a unique optimal solution $\mu_{ac}^{\mathrm{opt}} := \rho_{ac}\, \mu_{ac}$.*

The choice of maximizing the mass of the invariant measure in problem (1.25) is motivated by the following reasons:

- If we consider to solve only the feasibility constraints associated to problem (1.25), one could end up with a solution being the zero measure, even under Assumption 1.3.2.
- Enforcing the feasibility constraints by adding the condition for $\mu$ to be a probability measure (i.e., $\int_{\mathbf{X}} \mu = \|\mu\|_1 = 1$) would not provide any guarantee to obtain a feasible solution as the inequality constraints $\|\mu\|_p \leq 1$ may not be fulfilled since $\|\mu\|_1 \leq \operatorname{vol} \mathbf{X} \|\mu\|_p \leq \|\mu\|_p$ when $\mu \in \mathscr{L}^p(\mathbf{X})$ for some $p \geq 1$.

Let

$$\mathbf{C}_r^2(\mathbf{y}) := \begin{pmatrix} \mathbf{M}_r(\mathbf{y}^{\mathbf{X}}) & \mathbf{y}^r \\ (\mathbf{y}^r)^T & 1 \end{pmatrix}, \quad \mathbf{C}_r^\infty(\mathbf{y}) := \mathbf{M}_r(\mathbf{y}^{\mathbf{X}}) - \mathbf{M}_r(\mathbf{y}),$$

and from now on, let $p = 2$ or $p = \infty$ and $r \in \mathbb{N}$ be fixed, with $r \geq r_{\min}$. We build the following hierarchy of finite-dimensional SDP relaxations for problem (1.25):

$$\begin{aligned} \rho_{\mathrm{ac}}^r := \sup_{\mathbf{y}} \quad & y_0 \\ \text{s.t.} \quad & \mathscr{I}_{\mathbf{y}}(\mathbf{x}^\alpha) = 0, \quad \forall \alpha \in \mathbb{N}_{2r}^n, \\ & \mathbf{C}_r^p(\mathbf{y}) \succeq 0, \\ & \mathbf{M}_{r-r_j}(g_j\, \mathbf{y}) \succeq 0, \quad j = 0, 1, \ldots, m. \end{aligned} \tag{1.26}$$

**Lemma 1.3.3** *Problem (1.26) has a compact feasible set and an optimal solution $\mathbf{y}^r$.*

Let us denote by $\mathbb{R}[\mathbf{x}]'$ the dual set of $\mathbb{R}[\mathbf{x}]$, i.e., the linear functionals acting on $\mathbb{R}[\mathbf{x}]$.

**Lemma 1.3.4** *Let Assumption 1.3.2 hold and let $\mu_{ac}^{\mathrm{opt}}$ be the unique optimal solution of problem (1.25). For every $r \geq r_{\min}$, let $\mathbf{y}^r$ be an arbitrary optimal solution of problem (1.26) and by completing with zeros, consider $\mathbf{y}^r$ as an element of $\mathbb{R}[\mathbf{x}]'$. Then the sequence $(\mathbf{y}^r)_{r \geq r_{\min}} \subset \mathbb{R}[\mathbf{x}]'$ converges pointwise to $\mathbf{y}^{\mathrm{opt}} \in \mathbb{R}[\mathbf{x}]'$, that is, for any fixed $\alpha \in \mathbb{N}^n$:*

$$\lim_{r \to +\infty} y_\alpha^r = y_\alpha^{\mathrm{opt}}. \tag{1.27}$$

*Moreover, $\mathbf{y}^{\mathrm{opt}}$ has representing measure $\mu_{ac}^{\mathrm{opt}}$. In addition, one has:*

$$\lim_{r \to +\infty} \rho_{ac}^r = \rho_{ac} = \|\mu_{ac}^{\mathrm{opt}}\|_1. \tag{1.28}$$

**Remark 1.3.1** *Note that without the uniqueness hypothesis made in Assumption 1.3.2, we are not able to guarantee the pointwise convergence of the sequence of optimal solutions* $(\mathbf{y}^r)_{r\geq r_{\min}}$ *to* $\mathbf{y}^{\mathrm{opt}}$.

**Remark 1.3.2** *One could consider the dual of SDP (1.26), which is an optimization problem over polynomial SOS. One way to prove the non-existence of invariant densities in* $\mathscr{L}^p(\mathbf{X})$ *for* $p \in \{2, \infty\}$ *is to use the output of this dual program, yielding SOS certificates of infeasibility.*

Recall that $p = 2$ or $\infty$. Given a solution $\mathbf{y}^r$ of the SDP (1.26) at finite order $r \geq r_{\min}$, let $h^r \in \mathbb{R}[\mathbf{x}]_{2r}$ be the polynomial with vector of coefficients $\mathbf{h}^r$ given by:

$$\mathbf{h}^r := \mathbf{M}_r(\mathbf{y}^{\mathbf{X}})^{-1}\mathbf{y}^r \tag{1.29}$$

where the moment matrix $\mathbf{M}_r(\mathbf{y}^{\mathbf{X}})$ is positive definite hence invertible for all $r \in \mathbb{N}$. Note that the degree of the extracted invariant density depends on the SDP relaxation order $r$, and higher relaxation orders lead to higher degree approximations.

**Lemma 1.3.5** *Let Assumption 1.3.2 hold. For every* $r \geq r_{\min}$*, let* $\mathbf{y}^r$ *be an optimal solution of SDP (1.26), let* $h^r$ *be the corresponding polynomial obtained as in (1.29) and let* $\mu_{ac}^{\mathrm{opt}}$ *be the unique optimal solution of problem (1.25) with density* $h_{ac}^{\mathrm{opt}}$*. Then, the following convergence holds:*

$$\lim_{r\to+\infty} \int_{\mathbf{X}} g(\mathbf{x})\, h^r(\mathbf{x}) d\lambda = \int_{\mathbf{X}} g(\mathbf{x})\, h_{ac}^{\mathrm{opt}}(\mathbf{x}) d\lambda\,,$$

*for all* $g \in \mathbb{R}[\mathbf{x}]$.

In [J8, § 3.5], this methodology is extended to piecewise polynomial systems. The idea, inspired from [1], consists in using the piecewise structure of the dynamics and the state-space partition to decompose the invariant measure into a sum of local invariant measures supported on each partition cell while being invariant w.r.t. the local dynamics.

## Singular invariant measures

In the sequel, we focus on computing the support of singular measures for either discrete-time or continuous-time polynomial systems. Our approach is inspired from the framework presented in [181], yielding a numerical scheme to obtain the Lebesgue decomposition of a measure $\mu$ w.r.t. $\lambda$, for instance when $\lambda$ is the Lebesgue measure. By contrast with [181] where all moments of $\mu$ and $\lambda$ are *a priori* given, we only know the moments of the Lebesgue measure $\lambda$ in our case but we impose an additional constraint on $\mu$ to be an invariant probability measure.

We start by considering the infinite-dimensional linear optimization problem:

$$\begin{aligned} \rho_{\mathrm{sing}} = \sup_{\mu,\nu,\hat{\nu},\psi} \quad & \int_{\mathbf{X}} \nu \\ \text{s.t.} \quad & \int_{\mathbf{X}} \mu = 1\,, \quad \mathcal{L}(\mu) = 0\,, \\ & \nu + \psi = \mu\,, \quad \nu + \hat{\nu} = \lambda_{\mathbf{X}}\,, \\ & \mu, \nu, \hat{\nu}, \psi \in \mathscr{M}_+(\mathbf{X})\,. \end{aligned} \tag{1.30}$$

**Assumption 1.3.6** *There exists a unique invariant probability measure* $\mu^{\mathrm{opt}} \in \mathscr{M}_+(\mathbf{X})$.

For a measure $\nu$ with density $h \in \mathscr{L}_+^\infty(\mathbf{X})$, let us denote by $\min\{1, \nu\}$ the measure with density $x \mapsto \min\{1, h(x)\} \in \mathscr{L}_+^\infty(\mathbf{X})$.

**Theorem 1.3.6** *Under Assumption 1.3.6, LP (1.30) has a unique optimal solution* $(\mu^{\mathrm{opt}}, \nu_1^{\mathrm{opt}}, \lambda_{\mathbf{X}} - \nu_1^{\mathrm{opt}}, \mu^{\mathrm{opt}} - \nu_1^{\mathrm{opt}})$*, where* $(\nu^{\mathrm{opt}}, \mu^{\mathrm{opt}} - \nu^{\mathrm{opt}})$ *is the Lebesgue decomposition of* $\mu^{\mathrm{opt}}$ *w.r.t.* $\lambda_{\mathbf{X}}$ *and* $\nu_1^{\mathrm{opt}} := \min\{1, \nu^{\mathrm{opt}}\} \in \mathscr{L}_+^\infty(\mathbf{X})$.

Now we explain the rationale behind LP (1.30). When there is no absolutely continuous invariant probability measure supported on $\mathbf{X}$, then LP (1.30) has an optimal solution $(\mu^{\mathrm{opt}}, 0, \lambda_{\mathbf{X}}, \mu^{\mathrm{opt}})$ with $\mu^{\mathrm{opt}}$ being the unique singular invariant probability measure. In this case, the value of LP (1.30) is $\rho_{\mathrm{sing}} = 0$. Note that in the general case where Assumption 1.3.6 does not hold, there may be several invariant probability measures. In this case, LP (1.30) still admits an optimal solution and the optimal value is the maximal mass of the $\nu$-component among all invariant probability measures.

By contrast with problem (1.25) for absolutely continuous invariant densities, we enforce the feasibility constraints by adding the condition for $\mu$ to be a probability measure. The reason is that if we remove this condition, the value $\rho_{\mathrm{sing}} = 0$ could still be obtained with another optimal solution $(0, 0, \lambda_{\mathbf{X}}, 0)$, and we could not retrieve the unique invariant probability measure $\mu^{\mathrm{opt}}$. For every $r \geq r_{\min}$, we consider the following optimization problem:

$$
\begin{aligned}
\rho^r_{\mathrm{sing}} := \sup_{\mathbf{u}, \mathbf{v}, \hat{\mathbf{v}}, \mathbf{y}} \quad & v_0 \\
\text{s.t.} \quad & u_0 = 1\,, \quad \mathscr{I}_{\mathbf{u}}(\mathbf{x}^\alpha) = 0\,, \quad \forall \alpha \in \mathbb{N}^n_{2r}\,, \\
& v_\alpha + y_\alpha = u_\alpha\,, \quad v_\alpha + \hat{v}_\alpha = y^{\mathbf{X}}_\alpha\,, \quad \forall \alpha \in \mathbb{N}^n_{2r}\,, \\
& \mathbf{M}_{r-r_j}(g_j\, \mathbf{u})\,, \mathbf{M}_{r-r_j}(g_j\, \mathbf{v}) \succeq 0\,, \quad j = 0, \dots, m\,, \\
& \mathbf{M}_{r-r_j}(g_j\, \hat{\mathbf{v}})\,, \mathbf{M}_{r-r_j}(g_j\, \mathbf{y}) \succeq 0\,, \quad j = 0, \dots, m\,.
\end{aligned}
\tag{1.31}
$$

Problem (1.31) is a finite-dimensional SDP relaxation of LP (1.30), implying that $\rho^r_{\mathrm{sing}} \geq \rho_{\mathrm{sing}}$ for every $r \geq r_{\min}$.

**Theorem 1.3.7** *Problem (1.31) has a compact feasible set and an optimal solution, denoted by* $(\mathbf{u}^{\mathrm{opt}}, \mathbf{v}^{\mathrm{opt}}, \hat{\mathbf{v}}^{\mathrm{opt}}, \mathbf{y}^{\mathrm{opt}})$.

**Theorem 1.3.8** *Let Assumption 1.3.6 hold. For every* $r \geq r_{\min}$*, let* $(\mathbf{u}^r, \mathbf{v}^r, \hat{\mathbf{v}}^r, \mathbf{y}^r)$ *be an arbitrary optimal solution of SDP (1.31) and by completing with zeros, consider* $\mathbf{u}^r$*,* $\mathbf{v}^r$*,* $\hat{\mathbf{v}}^r$*,* $\mathbf{y}^r$ *as elements of* $\mathbb{R}[\mathbf{x}]'$*. The sequence* $(\mathbf{u}^r, \mathbf{v}^r, \hat{\mathbf{v}}^r, \mathbf{y}^r)_{r \geq r_{\min}} \subset (\mathbb{R}[\mathbf{x}]')^4$ *converges pointwise to* $(\mathbf{u}^{\mathrm{opt}}, \mathbf{v}^{\mathrm{opt}}, \hat{\mathbf{v}}^{\mathrm{opt}}, \mathbf{y}^{\mathrm{opt}}) \subset (\mathbb{R}[\mathbf{x}]')^4$*, that is, for any fixed* $\alpha \in \mathbb{N}^n$:

$$
\lim_{r \to \infty} u^r_\alpha = u^{\mathrm{opt}}_\alpha\,, \quad \lim_{r \to \infty} v^r_\alpha = v^{\mathrm{opt}}_\alpha\,, \quad \lim_{r \to \infty} \hat{v}^r_\alpha = z^{\mathbf{X}}_\alpha - v^{\mathrm{opt}}_\alpha\,, \quad \lim_{r \to \infty} y^r_\alpha = u^{\mathrm{opt}}_\alpha - v^{\mathrm{opt}}_\alpha\,. \tag{1.32}
$$

*Moreover, with* $(\mu^{\mathrm{opt}}, \nu_1^{\mathrm{opt}}, \lambda_{\mathbf{X}} - \nu_1^{\mathrm{opt}}, \mu^{\mathrm{opt}} - \nu_1^{\mathrm{opt}})$ *being the unique optimal solution of LP (1.30),* $\mathbf{u}^{\mathrm{opt}}$ *is the moment sequence of the unique invariant probability measure* $\mu^{\mathrm{opt}}$*,* $\mathbf{v}^{\mathrm{opt}}$ *and* $\mathbf{y}^{\mathrm{opt}}$ *are the respective moment sequences of* $\nu_1^{\mathrm{opt}}$*,* $\hat{\nu}^{\mathrm{opt}} = \lambda_{\mathbf{X}} - \nu_1^{\mathrm{opt}}$*,* $\mu^{\mathrm{opt}} - \nu_1^{\mathrm{opt}}$.

*In addition, one has:*

$$
\lim_{r \to \infty} \rho^r_{sing} = \rho_{sing}\,.
$$

The meaning of Theorem 1.3.8 is similar to the one of Theorem 3.4 in [181]. By noting $(\nu^{\mathrm{opt}}, \psi^{\mathrm{opt}})$ the Lebesgue decomposition of the unique invariant probability measure $\mu^{\mathrm{opt}}$, one has $\nu^{\mathrm{opt}}$ (resp. $\psi^{\mathrm{opt}}$) which is absolutely continuous (resp. singular) w.r.t. $\lambda_{\mathbf{X}}$. We have the two following cases:

1. If the decomposition $(\nu^{\mathrm{opt}}, \psi^{\mathrm{opt}})$ is feasible for LP (1.30), then $\nu^{\mathrm{opt}} \in \mathscr{L}_+^\infty(\mathbf{X})$ with $\|\nu^{\mathrm{opt}}\|_\infty \leq 1$. So, we can obtain all the moment sequences associated to $\nu^{\mathrm{opt}}$ and $\psi^{\mathrm{opt}}$ by computing $\mathbf{v}^r$ and $\mathbf{y}^r$ through solving SDP (1.31) as $r \to \infty$. In [181], the sup-norm must be less than an arbitrary fixed $\gamma > 0$ while in the present study we select $\gamma = 1$ as we consider an invariant probability measure $\mu^{\mathrm{opt}}$. In particular, when there is no invariant measure which is absolutely continuous w.r.t. $\lambda$, one has $\nu^{\mathrm{opt}} = \nu_1^{\mathrm{opt}} = 0$, $\psi^{\mathrm{opt}} = \mu^{\mathrm{opt}}$ and we obtain in the limit the moment sequence $\mathbf{y}^{\mathrm{opt}}$ of the singular measure $\mu^{\mathrm{opt}}$.

2. If the decomposition $(\nu^{\mathrm{opt}}, \psi^{\mathrm{opt}})$ is not feasible for LP (1.30), one has either $\nu^{\mathrm{opt}} \notin \mathscr{L}_+^\infty(\mathbf{X})$ or $\nu^{\mathrm{opt}} \in \mathscr{L}_+^\infty(\mathbf{X})$ with $\|\nu^{\mathrm{opt}}\|_\infty > 1$. Then one can define $\nu' = \min\{1, \nu^{\mathrm{opt}}\} \in \mathscr{L}_+^\infty(\mathbf{X})$ and $\psi' = \mu^{\mathrm{opt}} - \nu'$, such that $(\nu', \psi')$ is feasible for LP (1.30). In this case, the invariant probability measure $\mu^{\mathrm{opt}}$ is equal to $\nu' + \psi'$, but $\psi'$ is not singular w.r.t. $\lambda$.

**Definition 1.3.7** ***(Christoffel polynomial)*** *Assume that $\mu \in \mathcal{M}_+(\mathbf{X})$ is such that its moments are all finite and that for all $r \in \mathbb{N}$, the moment matrix $\mathbf{M}_r(\mathbf{u})$ is positive definite. With $\mathbf{v}_r(\mathbf{x})$ denoting the vector of monomials of degree less or equal than $r$, sorted by graded lexicographic order, the Christoffel polynomial is the function $p_{\mu,r} : \mathbf{X} \to \mathbb{R}$ such that*

$$\mathbf{x} \mapsto p_{\mu,r}(\mathbf{x}) := \mathbf{v}_r(\mathbf{x})^T \mathbf{M}_r(\mathbf{u})^{-1} \mathbf{v}_r(\mathbf{x}).$$

The following assumption is similar to [187, Assumption 3.6 (b)]. It provides the existence of a sequence of thresholds $(\alpha_r)_{r\in\mathbb{N}}$ for the Christoffel function associated to a given measure $\mu$ in order to approximate the support $\mathbf{S}$ of this measure. Here, we do not assume as in [187, Assumption 3.6 (a)] that the closure of the interior of $\mathbf{S}$ is equal to $\mathbf{S}$.

**Assumption 1.3.8** *Given a measure $\mu \in \mathcal{M}_+(\mathbf{X})$ with support $\mathbf{S} \subseteq \mathbf{X}$, $\mathbf{S}$ has nonempty interior and there exist three sequences $(\delta_r)_{r\in\mathbb{N}}$, $(\alpha_r)_{r\in\mathbb{N}}$, $(d_r)_{r\in\mathbb{N}}$ such that:*

- *$(\delta_r)_{r\in\mathbb{N}}$ is a decreasing sequence of positive numbers converging to 0.*
- *For every $r \in \mathbb{N}$, $d_r$ is the smallest integer such that:*

$$2^{3 - \frac{\delta_r d_r}{\delta_r + \operatorname{diam} \mathbf{S}}} d_r^n \left(\frac{e}{n}\right)^n \exp\left(\frac{n^2}{d_r}\right) \leq \alpha_r \,. \tag{1.33}$$

- *For every $r \in \mathbb{N}$, $\alpha_r$ is defined as follows:*

$$\alpha_r := \frac{\delta_r^n \omega_n}{\operatorname{vol} \mathbf{S}} \frac{(d_r+1)(d_r+2)(d_r+3)}{(d_r+n+1)(d_r+n+2)(2d_r+n+6)}\,,$$

  *where $\operatorname{diam} \mathbf{S}$ denotes the diameter of the set $\mathbf{S}$ and $\omega_n := \frac{2\pi^{\frac{n+1}{2}}}{\Gamma(\frac{n+1}{2})}$ is the surface of the n-dimensional unit sphere in $\mathbb{R}^{n+1}$.*

**Remark 1.3.3** *Regarding Assumption 1.3.8, as mentioned in [187, Remark 3.7], $d_r$ is well defined for all $r \in \mathbb{N}$ and the sequence $(d_r)_{r\in\mathbb{N}}$ is nondecreasing. This comes from the fact that the left-hand side of (1.33) goes to 0 as $d_r \to \infty$ and that $\alpha_r \to \frac{\delta_r^n \omega_n}{2 \operatorname{vol} \mathbf{S}}$ as $d_r \to \infty$. Since $\operatorname{diam} \mathbf{S} \leq \operatorname{diam} \mathbf{X} \leq 1$ and $\operatorname{vol} \mathbf{S} \leq \operatorname{vol} \mathbf{X} \leq 1$, replacing $\operatorname{diam} \mathbf{S}$ and $\operatorname{vol} \mathbf{S}$ by 1 in (1.33) yields a result similar to Theorem 1.3.9. Assume that one has a given sequence $(\delta_r)_{r\in\mathbb{N}}$. Then, since $(d_r)_{r\in\mathbb{N}}$ is a nondecreasing sequence of integers, one can simply start from $d_0 := 1$, and increase the value of $d_0$ until (1.33) holds. This yields a simple recursive method to compute $(d_r)_{r\in\mathbb{N}}$ as well as the corresponding threshold $\alpha_r$ for the Christoffel polynomial.*

**Theorem 1.3.9** *Let Assumption 1.3.6 hold and let $\mathbf{S} \subseteq \mathbf{X}$ be the support of the invariant probability measure $\mu^{\text{opt}}$. Suppose that there exist sequences $(\delta_r)_{r\in\mathbb{N}}$, $(\alpha_r)_{r\in\mathbb{N}}$ and $(d_r)_{r\in\mathbb{N}}$ such that $\mu^{\text{opt}}$, $\mathbf{S}$, $(\delta_r)_{r\in\mathbb{N}}$, $(\alpha_r)_{r\in\mathbb{N}}$ and $(d_r)_{r\in\mathbb{N}}$ fulfill Assumption 1.3.8. For every $r \in \mathbb{N}$, let:*

$$\mathbf{S}^r := \{\mathbf{x} \in \mathbf{X} : p_{\mu^{\text{opt}},d_r}(\mathbf{x}) \leq \frac{\binom{d_r+n}{n}}{\alpha_r}\}\,. \tag{1.34}$$

*Then* $\lim_{r\to\infty} \sup_{\mathbf{x}\in\mathbf{S}^r} \operatorname{dist}(\mathbf{x}, \mathbf{S}) = 0$.

The proof can be found in [J8] and relies on Assumption 1.3.8 and [187, Lemma 6.6].

**Remark 1.3.4** *In the case when the invariant measure is discrete singular, its moment matrix may not be invertible. Indeed, if the invariant measure is a convex combination of r atoms, then the rank of the moment matrix is upper bounded by r. Thus, we cannot approximate the support of such a measure with the level sets of the Christoffel polynomial. In this case, one way to recover the support of the measure is to rely on the numerical linear algebra algorithm proposed in [131] for detecting global optimality and extracting solutions of moment problems. This algorithm is implemented in the* GloptiPoly *[132] software and has been already used in previous work [128] to recover finite cycles in the context of discrete-time systems. Note that if the invariant measure has a non trivial absolutely continuous component $\nu$ (in its Lebesgue decomposition), our approach based on Christoffel polynomials also allows to approximate the support of $\nu$, while relying on the inverse of $\mathbf{M}_r(\mathbf{v})$ at relaxation order r. However, in this case, we are more interested in approximating the density of $\nu$, by relying on the first Lasserre hierarchy provided for absolutely continuous invariant densities.*

## Numerical experiments

Here, we present experimental benchmarks that illustrate our method. For invariant densities, we compute the optimal solution $\mathbf{y}^r$ of the primal SDP program (1.26) for a given positive integer $r$ and $p = 2$ or $\infty$, as well as the approximate polynomial density $h_p^r$ defined in (1.29). For singular measures, we compute the optimal solution $\mathbf{u}^r$ of the primal SDP program (1.31) for a given positive integer $r$ as well as $\mathbf{S}^r$, the sublevel set of the Christoffel polynomial defined in (1.34). In practice, accordingly with the discussion from Remark 1.3.3, the computation of $\alpha_r$ in (1.34) relies on the following iterative procedure: we select $d_r = r$, $\delta_r = 1$ and increment the value of $\delta_r$ until the inequality (1.33) from Assumption 1.3.8 is satisfied. SDPs (1.26) and (1.31) are both modeled through GloptiPoly [132] via Yalmip toolbox [200] available within Matlab and interfaced with the SDP solver Mosek [7]. The sources of our code are available online[2].

For each problem, we apply a preprocessing step which consists in scaling data (dynamics, general state constraints) so that the constraint sets become unit boxes. Note that our theoretical framework (including convergence of the SDP relaxations) only works after assuming that there exists a unique invariant probability measure (Assumption 1.3.2 and Assumption 1.3.6). Even though this may not hold for some of the considered systems, numerical experiments show that satisfying results can be obtained when approximating invariant densities and supports of singular measures.

First, let us consider the one-dimensional discrete-time polynomial system defined by

$$t^+ = A(t) := t + w \mod 1\,,$$

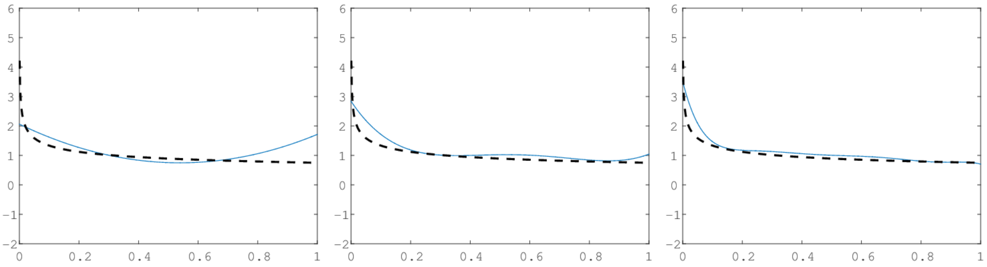
[2] http://homepages.laas.fr/vmagron/invsdp.tar.gz

with $t$ being constrained in the interval $[0,1]$ and $w \in \mathbb{R}\backslash\mathbb{Q}$ be an arbitrary irrational number. This dynamics corresponds to the circle rotation with an irrational angle $w$ and thus it has a unique invariant measure equal to the restriction of the Lebesgue measure on $[0,1]$ [303], i.e., Assumption 1.3.2 is fulfilled here.

Let us consider the square integrable probability density $h^{\text{opt}}(t) := \frac{3}{4}t^{-1/4}$ and let $F(t) := \int_0^t h^{\text{opt}}(s)ds = t^{3/4}$ be its cumulative distribution function. Since $F$ is invertible, then $F^{-1}(t) = t^{4/3}$ is distributed according to $h^{\text{opt}}$ and hence the following dynamical system

$$x^+ = F^{-1} \circ A \circ F(x)\,, \tag{1.35}$$

with $x$ being constrained in the interval $\mathbf{X} = [0,1]$, has the invariant measure with density $h^{\text{opt}} \in \mathscr{L}^2(\mathbf{X})$.



(a) $r = 4$ (b) $r = 6$ (c) $r = 8$

Figure 1.5: Approximate invariant density for the dynamics from (1.35) with corresponding approximations $h_2^r$ (solid curve) of the exact density $h^{\text{opt}}$ (dashed curve) for $r \in \{4,6,8\}$ and $w = \frac{\sqrt{99}}{10}$.

Now, let us take $w \in (0,1)$. In this case one has $F^{-1} \circ A \circ F(x) = (x^{3/4} + w)^{4/3}$ if $x^{3/4} + w \leq 1$ and $F^{-1} \circ A \circ F(x) = (x^{3/4} + w - 1)^{4/3}$ otherwise. In order to cast the dynamical system from (1.35) as a piecewise polynomial system, we introduce two additional (so-called *lifting*) variables $y, z$ and consider the system defined as follows:

$$x^+ = y \quad \text{for } (x,y,z) \in \mathbf{X}_1 \cup \mathbf{X}_2 \tag{1.36}$$

where $\mathbf{X}_1$ and $\mathbf{X}_2$ are defined by

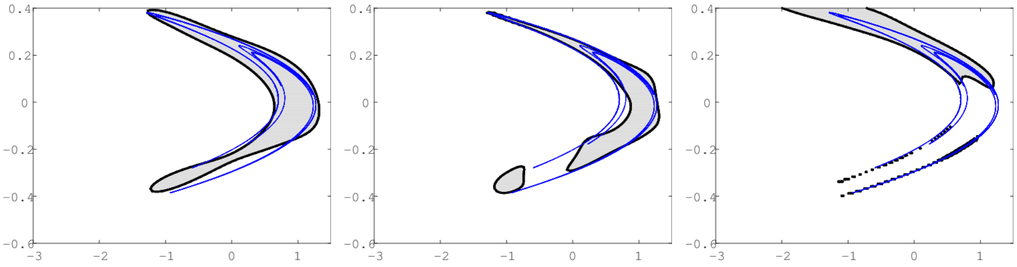
$$\begin{aligned}\mathbf{X}_1 &:= \{(x,y,z) \in \mathbb{R}^3 : z(1-w-z) \geq 0, z^4 = x^3, (z+w)^4 = y^3\}\,,\\ \mathbf{X}_2 &:= \{(x,y,z) \in \mathbb{R}^3 : (1-z)(z+w-1) \geq 0, z^4 = x^3, (z+w-1)^4 = y^3\}\,.\end{aligned}$$

Note that the collection $\{\mathbf{X}_1, \mathbf{X}_2\}$ is a partition of $[0,1]^3$. The variable $z$ represents $x^{3/4}$, for all $x \in [0,1]$. The variable $y$ represents either $(x^{3/4}+w)^{4/3}$ on $\mathbf{X}_1$ or $(x^{3/4}+w-1)^{4/3}$ on $\mathbf{X}_2$. Using the extension of our results to piecewise polynomial systems, we performed numerical experiments with the irrational number $w = \frac{\sqrt{99}}{10}$. The approximate density $h_2^r$ obtained in (1.29) from the $r$ first moments (for $r = 4,6,8$) and the exact density $h^{\text{opt}}$ are displayed on Figure 1.5. These numerical results indicate that the density approximations become tighter when the value of $r$ increases.

The Hénon map is a famous example of two-dimensional discrete-time systems that exhibit a chaotic behavior. The system is defined as follows:

$$\begin{aligned}x_1^+ &= 1 - ax_1^2 + x_2\,,\\ x_2^+ &= bx_1\,.\end{aligned}$$

with general state constraints within the box $\mathbf{X} = [-3, 1.5] \times [-0.6, 0.4]$. For $a = 1.4$ and $b = 0.3$,



(a) $r = 4$ (b) $r = 6$ (c) $r = 8$

Figure 1.6: Hénon attractor (blue) and approximations $\mathbf{S}^r$ (light gray) for the support of the invariant measure, with $r \in \{4, 6, 8\}$, $a = 1.4$ and $b = 0.3$.

Hénon proves in [127] that the sequence of points obtained by iteration of this map from an initial point can either diverge or tend to a strange attractor being the product of a one-dimensional manifold by a Cantor set.

Figure 1.6 displays set approximations $\mathbf{S}^r$ obtained from (1.34) of the support of the measure invariant w.r.t. the Hénon map, for $r \in \{4, 6, 8\}$. For comparison purpose, we also represented the "true" Hénon attractor by displaying the sequence of points obtained after hundred iterations of the map while starting from random sampled initial conditions within the disk of radius 0.1 and center $(-1, 0.4)$. These numerical experiments show that the level sets of the Christoffel polynomial provide fairly tight approximations of the attractor for modest values of the relaxation order.

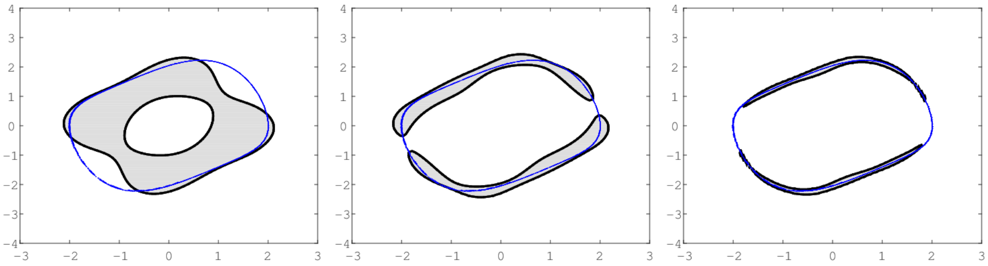
The Van der Pol oscillator is an example of an oscillating system with nonlinear damping [291]. The dynamics are given by the following second-order ordinary differential equation:

$$\ddot{x}_1 - a(1 - x_1^2)\dot{x}_1 + x_1 = 0 \,.$$

By setting $x_2 = \dot{x_1}$, one can reformulate this one-dimensional system into a two-dimensional continuous-time system:

$$\begin{aligned} \dot{x_1} &= x_2 \,, \\ \dot{x_2} &= a(1 - x_1^2)x_2 - x_1 \,. \end{aligned}$$

When $a > 0$, there exists a limit cycle for the system. Here, we consider $a = 0.5$ and general state constraints within the box $\mathbf{X} = [-3, 3] \times [-4, 4]$.



(a) $r = 4$ (b) $r = 6$ (c) $r = 8$

Figure 1.7: Van der Pol attractor (blue) and approximations $\mathbf{S}^r$ (light gray) for the support of the invariant measure, with $r \in \{4, 6, 8\}$ and $a = 0.5$.

Figure 1.7 shows set approximations $\mathbf{S}^r$ obtained from (1.34) of the support of the measure invariant w.r.t. the Van der Pol map. As for the Hénon map, we also represent the "true" limit cycle after performing a numerical integration of the Van der Pol system from $t_0 = 0$ to $T = 20$ with random sampled initial conditions within the disk of radius 0.1 and center $(1, -1)$. This numerical approximation is done with the `ode45` procedure available inside MATLAB. Once again, the plots exhibit a quite fast convergence behavior of the approximations $\mathbf{S}^r$ of the invariant measure support to the limit cycle when $r$ increases.

## 1.4 Boundaries of semialgebraic sets

The materials from this section have been published in [J4]. Given $m \in \mathbb{N}_{>0}$, let us define $[m] := \{1, \ldots, m\}$. We focus on the Hausdorff boundary measure of a basic compact semialgebraic set $\mathbf{\Omega} \subset \mathbb{R}^n$

$$\mathbf{\Omega} := \{\mathbf{x} \in \mathbb{R}^n : g_j(\mathbf{x}) \leq b_j, \quad j \in [m]\}, \tag{1.37}$$

with $g_j \in \mathbb{R}[\mathbf{x}]$, for each $j \in [m]$. For later purpose, we also note $g_0(x) := -1$ and $b_0 := 0$.

Throughout this section, we suppose that the following condition is fulfilled:

**Assumption 1.4.1** $\mathbf{\Omega} \subset \mathbf{B} := (-1, 1)^n$.

Since $\mathbf{\Omega}$ is compact, Assumption 1.4.1 is satisfied, possibly after rescaling of the data. From now on, let $\lambda_{\mathbf{B}}$ be the Lebesgue measure on $\mathbf{B}$. Its moments are denoted $\mathbf{y}^{\mathbf{B}} = (y_\alpha^{\mathbf{B}})_{\alpha \in \mathbb{N}^n}$, i.e.,

$$y_\alpha^{\mathbf{B}} := \int_{\mathbf{B}} \mathbf{x}^\alpha \lambda_{\mathbf{B}}(d\mathbf{x}) = \int_{\mathbf{B}} x_1^{\alpha_1} \cdots x_n^{\alpha_n} \lambda_{\mathbf{B}}(d\mathbf{x}) = \prod_{i=1}^n \frac{1 + (-1)^{\alpha_i}}{\alpha_i + 1}, \quad \alpha \in \mathbb{N}^n, \tag{1.38}$$

thus are available analytically. In particular $|y_\alpha^{\mathbf{B}}| \leq y_0^{\mathbf{B}} = 2^n$, for all $\alpha \in \mathbb{N}^n$. Other choices of $\mathbf{B}$ (e.g. an Euclidean ball, a box, an ellipsoid) are possible as soon as all moments of $\lambda_{\mathbf{B}}$ can be obtained easily or in closed form. As in [130] and other problems previously described in this chapter, Assumption 1.4.1 is required to be able to compute moments of the Lebesgue measure on $\mathbf{\Omega}$, which in turn is required to compute moments on the boundary measure on $\partial\mathbf{\Omega}$.

Our main contribution is a systematic numerical scheme to approximate as closely as desired any (fixed) finite number of its moments, in particular its mass (the length or area of $\partial\mathbf{\Omega}$) and integrals of polynomials on $\partial\mathbf{\Omega}$.

Besides being a challenge of its own in computational mathematics, computation of moments (e.g. the mass) of the boundary measure has also important applications: either practical ones, e.g., in computational geometry [292] (perimeter, surface area), or in control for computing the length of trajectories in the context of robot motion planning [59], or theoretical ones such as (real) periods computation [161, 266]. Periods are integrals of rational functions with rational coefficients over semialgebraic sets and the moments of the Hausdorff boundary measure of $\mathbf{\Omega}$ are special cases of periods where the integrated rational functions are monomials. In some applications, e.g., tomography [115], the moments of the Lebesgue measure on $\mathbf{\Omega}$ are available after appropriate measurements. In this case the methodology slightly simplifies as they now appear as data instead of variables.

Of course certain *line* or *surface* specific integrals on $\partial\mathbf{\Omega}$ reduce to surface of volume integrals of a related function on $\mathbf{\Omega}$ via Green's (or Stokes') theorem, in which case one may invoke the arsenal of techniques of multivariate integration on specific domains $\mathbf{\Omega}$ like Monte-Carlo and/or cubatures techniques. We refer to [52, 63, 66, 192] for some previous studies on convex and non convex polytopes. There are few available systematic numerical schemes in the literature for computing “volume" or moments of the boundary $\partial\mathbf{\Omega}$ of a basic semialgebraic set $\mathbf{\Omega}$. Our work seems to be one of the first such attempts, at least at this level of generality.

## Classical approaches

Among existing techniques for numerical volume computation and integration, Monte Carlo algorithms [51, 313, 194] generate points uniformly in a box which contains $\mathbf{\Omega}$ and approximate the volume by the ratio of the number of points that fall into $\mathbf{\Omega}$, and similarly for integration. Cubature formulae [70, 308] provide a way to perform numerical integration on simple sets, such as simplices, boxes or balls, or tetrahedra [146]. However, such formula are not available for arbitrary semialgebraic sets. When the set $\partial\mathbf{\Omega}$ is defined explicitly by polynomial equality constraints, the algorithm from [49] allows one to compute integrals and perform sampling from distributions on $\partial\mathbf{\Omega}$, based on intersecting with random linear spaces. Let us also mention the recent work [168] which provides arbitrary precision approximations for the volume of $\mathbf{\Omega}$. This algorithm is based on computing the Picard-Fuchs differential equations of appropriate periods and critical point properties.

Concerning volume computation of a semialgebraic set $\mathbf{\Omega}$, its GMP formulation as an infinite-dimensional LP has been developed in [130]. It requires knowledge of a simple set $\mathbf{B} \supset \mathbf{\Omega}$ such that all moments of the Lebesgue measure on $\mathbf{B}$ are available (e.g., $\mathbf{B}$ is a Euclidean ball, an ellipsoid, a box). When $\mathbf{\Omega}$ is the level set of a single homogeneous polynomial, a related and alternative method has been proposed in [186], which results in solving a hierarchy of generalized eigenvalue problems (rather than a sequence of SDP problems) with respect to a pair of Hankel matrices of increasing size. Here, our approach consists of extracting information on the boundary measure on $\partial\mathbf{\Omega}$ from some available moments, here moments of the Lebesgue measure on $\mathbf{\Omega}$. The latter are either already available (e.g., from measurements or computation) or must also be approximated. A similar approach has been previously exploited in [48] for the moment problem with holonomic functions and in [99] for inverse moment problems for convex polytopes. In the recent work [80] the authors propose to reconstruct mixtures of gaussian distributions from knowledge of the so-called *derivatives* of moments. Finally we also refer to [6] for a related work in the bi-dimensional case, using generalized polarization tensors.

Our contribution is in the spirit of the work [130] as we also formulate our problem as a GMP, i.e., an infinite-dimensional LP on appropriate spaces of measures. Indeed, to compute the moments of the boundary measure $\sigma$ on $\partial\mathbf{\Omega}$ we relate them (linearly) to moments of the Lebesgue measure $\lambda$ on $\mathbf{\Omega}$ via Stokes' theorem.

## Stokes' theorem

Let $\mathbf{\Omega}$ be a smooth manifold with boundary $\partial\mathbf{\Omega}$. Given a polynomial $g \in \mathbb{R}[\mathbf{x}]$, Stokes' theorem with vector field $X$ states that:

$$\int_{\mathbf{\Omega}} \operatorname{div}(X\, g(\mathbf{x}))\, d\mathbf{x} \;=\; \int_{\partial\mathbf{\Omega}} \langle X, \vec{n}_{\mathbf{x}} \rangle\, g(\mathbf{x})\, d\sigma\,, \tag{1.39}$$

where $\partial\mathbf{\Omega}$ stands for the boundary of $\mathbf{\Omega}$, $\sigma$ is the $(n-1)$-dimensional Hausdorff boundary measure on $\partial\mathbf{\Omega}$, and $\vec{n}_{\mathbf{x}}$ is the outward pointing normal to $\partial\mathbf{\Omega}$; see, e.g., Taylor [287, Proposition 3.2, p. 128]. Then Whitney [305, Theorem 14A] generalized Stokes' theorem to rough domains $\mathbf{\Omega}$ (e.g. with corners). For instance, in our case:

$$\partial\mathbf{\Omega} \;=\; \cup_{j=1}^{m} \mathbf{\Omega}_j \text{ with } \mathbf{\Omega}_j \;=\; \{\mathbf{x} \in \mathbf{\Omega} : g_j(\mathbf{x}) \;=\; b_j\,\}. \tag{1.40}$$

In the remaining part of this section, we derive preliminary results and required assumptions that are core components of our framework. Throughout this section, we make the following non-degeneracy assumption:

**Assumption 1.4.2**

$$\forall \mathbf{x} \in \mathbf{\Omega}_j : \quad \|\nabla g_j(\mathbf{x})\| \neq 0, \quad j \in [m], \tag{1.41}$$

*and therefore, with* $\mathbf{x} \mapsto t_j(\mathbf{x}) := \|\nabla g_j(\mathbf{x})\|^2$*,*

$$t_j(\mathbf{x}) \,\geq\, a_j\,, \qquad \forall \mathbf{x} \in \mathbf{\Omega}_j\,, \quad j \in [m], \tag{1.42}$$

*for some* $a_j > 0$*,* $j \in [m]$*.*

We also make the following technical assumption on the polynomials $g_j$ that define the boundary $\partial\mathbf{\Omega}$.

**Assumption 1.4.3** *Let* $\mathbf{\Omega}_j$ *be as in* (1.40) *and let* $\sigma_j$ *be the restriction to* $\mathbf{\Omega}_j$ *of the Hausdorff measure* $\sigma$ *on* $\partial\mathbf{\Omega}$*,* $j \in [m]$*.*

*Then* $\sigma(\mathbf{\Omega}_j \cap \mathbf{\Omega}_k) = 0$ *for all pairs* $(j,k)$*,* $j \neq k$*. In particular, for every* $j \in [m]$*,* $\sigma_j(\{\mathbf{x} : g_k(\mathbf{x}) = b_k\}) = 0$ *for all* $k \neq j$*, and:*

$$\sigma(\partial\mathbf{\Omega}) \,=\, \sigma(\cup_{j=1}^m \mathbf{\Omega}_j) \,=\, \sum_{j=1}^m \sigma(\mathbf{\Omega}_j) \,=\, \sum_{j=1}^m \sigma_j(\mathbf{\Omega}_j).$$

Developing (1.39) yields:

$$\int_{\mathbf{\Omega}} \mathrm{div}(X)\, g(\mathbf{x}) + \langle X, \nabla g(\mathbf{x})\rangle\, d\mathbf{x} \,=\, \int_{\partial\mathbf{\Omega}} \langle X, \vec{n}_{\mathbf{x}}\,\rangle\, g(\mathbf{x})\, d\sigma(\mathbf{x})\,. \tag{1.43}$$

Next, select the vector field $X = \mathbf{x}$ and note that $\langle \mathbf{x}, \nabla \mathbf{x}^\alpha\rangle = |\alpha|\mathbf{x}^\alpha$ for all $\alpha \in \mathbb{N}^n$. Then under Assumption 1.4.2-1.4.3, letting $g(\mathbf{x}) := \mathbf{x}^\alpha$ in (1.43) with $\alpha \in \mathbb{N}^n$ fixed, arbitrary, yields:

$$(n + |\alpha|) \int_{\mathbf{\Omega}} \mathbf{x}^\alpha\, d\mathbf{x} \;=\; \sum_{j=1}^m \int_{\mathbf{\Omega}_j} \left\langle \mathbf{x}, \frac{\nabla g_j(\mathbf{x})}{\|\nabla g_j(\mathbf{x})\|} \right\rangle \mathbf{x}^\alpha\, d\sigma_j(\mathbf{x}), \tag{1.44}$$

$$=\; \sum_{j=1}^m \int_{\mathbf{\Omega}_i} \mathbf{x}^\alpha\, \langle \mathbf{x}, \nabla g_j(\mathbf{x})\rangle\, \frac{d\sigma_j(\mathbf{x})}{\|\nabla g_j(\mathbf{x})\|}\,. \tag{1.45}$$

Note that the outgoing normal vector $\vec{n}_{\mathbf{x}}$ at $\mathbf{x} \in \mathbf{\Omega}_j$ is $\frac{\nabla g_j(\mathbf{x})}{\|\nabla g_j(\mathbf{x})\|}$, as a consequence of the inequality "$g_j(\mathbf{x}) \leq b_j$" whenever $\mathbf{x} \in \mathbf{\Omega}$.

When $\alpha = 0$ (1.43) has a simple geometric interpretation, easy to visualize in dimension $n = 2$. Indeed with $0 \in \mathrm{int}(\mathbf{\Omega})$, (i) $\langle \mathbf{x}, \vec{n}_{\mathbf{x}}\rangle$ is the "height" from 0 to the hyperplane tangent to $\mathbf{\Omega}$ at the point $M \in \partial\mathbf{\Omega}$ (with coordinate $\mathbf{x}$), and (ii) $d\sigma(\mathbf{x})$ is the infinitesimal "length" $[M, M']$ on $\partial\mathbf{\Omega}$ around $M$, and so $\frac{1}{n}\langle \mathbf{x}, \vec{n}_{\mathbf{x}}\rangle d\sigma(\mathbf{x})$ is the infinitesimal "area" of the triangle $(O, M, M')$, that is the length of base $[M, M']$ times the height $[0, M]$; see Figure 1.8.

There is also a non-geometric interpretation. When the $g_j$'s are polynomials then so are the functions $\mathbf{x} \mapsto \langle \mathbf{x}, \nabla g_j(\mathbf{x})\rangle$'s, which yields a simple interpretation for (1.44). Indeed (1.44) states that for each $\alpha \in \mathbb{N}^n$, the moment $\int_{\mathbf{\Omega}} \mathbf{x}^\alpha d\mathbf{x}$ is some *linear combination* of moments of the measure $d\hat{\sigma} = f d\sigma$ on $\partial\mathbf{\Omega}$, with density $f(\mathbf{x}) := \|\nabla g_j(\mathbf{x})\|^{-1}$ on $\mathbf{\Omega}_j$, $j \in [m]$, (recall (1.41)).

**Remark 1.4.1** *Assumption 1.4.3 will also allow us later on (in Theorem 1.4.1) to characterize the boundary measure* $\sigma_j$ *of* $\mathbf{\Omega}_j$ *as the unique optimal solution of an infinite-dimensional LP over measures (LP* (1.50)*). Assumption 1.4.3 may fail if for instance two polynomials* $b_j - g_j$ *and* $b_k - g_k$ *have a common factor* $h \in \mathbb{R}[\mathbf{x}]$ *which is nonnegative on* $\mathbf{\Omega}$*. That is,* $b_j - g_j = \tilde{g}_j\, h$*,* $b_k - g_k = \tilde{g}_k\, h$*, and therefore*

$$H \,:=\, \{\mathbf{x} \in \mathbf{\Omega} : h(\mathbf{x}) \,=\, 0\} \,\subset\, \mathbf{\Omega}_j \cap \mathbf{\Omega}_k.$$

*So it may happen that* $\sigma(H) > 0$*, which violates the condition* $\sigma(\mathbf{\Omega}_j \cap \mathbf{\Omega}_k) = 0$ *in Assumption 1.4.3, and which implies that* (1.44) *is not correct. In such a simple case it is easy to provide an equivalent reformulation of* $\mathbf{\Omega}$ *which prevents the assumption from failing; just replace* $b_j - g_j \geq 0$ *with* $\tilde{g}_j \geq 0$*, replace* $b_k - g_k \geq 0$ *with* $\tilde{g}_k \geq 0$*, and introduce the additional constraint* $h \geq 0$*. However for more general semialgebraic sets, such a reformulation might be delicate and beyond the scope of this work. In practical applications, assuming that* $\sigma(\Omega_j \cap \Omega_k) = 0$ *for every arbitrary pair* $(j,k)$*,* $j \neq k$*, seems to us quite reasonable.*

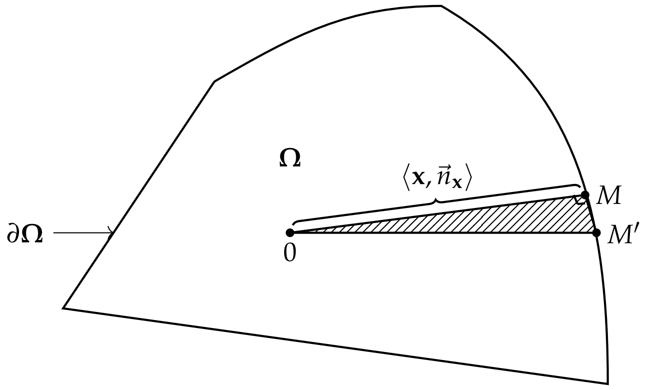

$\boldsymbol{\Omega}$
$\langle \mathbf{x}, \vec{n}_{\mathbf{x}} \rangle$
$M$
$M'$
$\partial\boldsymbol{\Omega}$
0

Figure 1.8: Geometric interpretation.

If $g_j$ is homogeneous of degree $d_j$, for each $j \in [m]$, then $\langle \mathbf{x}, \nabla g_j(\mathbf{x}) \rangle = d_j \, g_j(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{R}^n$, and therefore (1.44) yields:

$$(n + |\alpha|) \int_{\boldsymbol{\Omega}} \mathbf{x}^{\alpha} \, d\mathbf{x} = \sum_{j=1}^{m} d_j \, b_j \int_{\boldsymbol{\Omega}_j} \mathbf{x}^{\alpha} \, \frac{d\sigma_j(\mathbf{x})}{\|\nabla g_j(\mathbf{x})\|}, \quad \alpha \in \mathbb{N}^n \,. \tag{1.46}$$

Recall that $\lambda_{\boldsymbol{\Omega}}$ is the restriction of the Lebesgue measure on $\boldsymbol{\Omega}$.

## A first observation

**Lemma 1.4.4** *Let $\boldsymbol{\Omega}$ in (1.37) be compact with $b_j > 0$ and where $g_j \in \mathbb{R}[\mathbf{x}]$ is homogeneous of degree $d_j$, $j \in [m]$. Let Assumption 1.4.1 and 1.4.2 hold. Let $\mathbf{y} = (y_\alpha)_{\alpha \in \mathbb{N}^n}$ be the sequence of moments of $\lambda_{\boldsymbol{\Omega}}$. Then the sequence $\hat{\mathbf{y}} = (\hat{y}_\alpha)_{\alpha \in \mathbb{N}^n} = ((n + |\alpha|) y_\alpha)_{\alpha \in \mathbb{N}^n}$ is the sequence of moments uniquely represented by the boundary measure $\phi$ on $\partial\boldsymbol{\Omega}$, absolutely continuous with respect to the Hausdorff measure $\sigma$ on $\partial\boldsymbol{\Omega}$, which reads:*

$$\begin{aligned} d\phi(\mathbf{x}) \quad &:= \quad \sum_{j=1}^{m} \frac{b_j \, d_j}{\|\nabla g_j(\mathbf{x})\|} \, d\sigma_j(\mathbf{x}), \\ &= \quad \frac{b_j \, d_j}{\|\nabla g_j(\mathbf{x})\|} \, d\sigma(\mathbf{x}), \quad \forall \mathbf{x} \in \boldsymbol{\Omega}_j, \quad j \in [m]. \end{aligned} \tag{1.47}$$

To recover the boundary measure $\sigma$ from $\mathbf{y} = (y_\alpha)_{\alpha \in \mathbb{N}^n}$ we proceed in two steps. Define

$$d\phi_j(\mathbf{x}) = \frac{d\sigma_j(\mathbf{x})}{\|\nabla g_j(\mathbf{x})\|}, \quad j \in [m], \tag{1.48}$$

so that by (1.43) and (1.47), for every $\alpha \in \mathbb{N}^n$:

$$(n + |\alpha|) \, y_\alpha = \sum_{j=1}^{m} b_j d_j \int_{\boldsymbol{\Omega}_j} \mathbf{x}^{\alpha} \, d\phi_j(\mathbf{x}) = \int_{\partial\boldsymbol{\Omega}} \mathbf{x}^{\alpha} \sum_{j=1}^{m} d_j b_j \, d\phi_j. \tag{1.49}$$

The first step consists of exploiting the latter linear equality constraints (1.49) to approximate as closely as desired the moments of $\phi_j$ by solving a first hierarchy of SDP. The second step exploits the linear relations (1.48) between $\phi_j$ and the boundary measure $\sigma_j$ to derive a second hierarchy of SDP. This in turn allows one to compute the moments of $\sigma_j$ as closely as desired. In the sequel, we provide more details on both steps, in the case where the $g_j$'s are homogeneous polynomials. We refer the interested reader to [J4, § 4] for the general case where the $g_j$'s are not necessarily homogeneous.

### Step 1: computing moments of the measures $\phi_j$ in (1.48)

Consider the following infinite-dimensional LP:

$$\begin{aligned} \rho = \sup_{\mu,\mu_j} \int_{\mathbf{\Omega}} d\mu \ : & \\ \text{s.t.}\ \mu \leq \lambda_{\mathbf{B}}\,;\ \ (n+|\alpha|)\int_{\mathbf{\Omega}} \mathbf{x}^\alpha d\mu &= \sum_{j=1}^m d_j\, b_j \int_{\mathbf{\Omega}_j} \mathbf{x}^\alpha d\mu_j,\ \alpha \in \mathbb{N}^n\,, \\ \mu \in \mathscr{M}_+(\mathbf{\Omega})\,, \mu_j \in \mathscr{M}_+(\mathbf{\Omega}_j),\ j \in [m]\,. & \end{aligned} \tag{1.50}$$

We recall that from [130, theorem 3.1], the uniform measure $\lambda_{\mathbf{\Omega}}$ is the unique optimal solution of the following infinite-dimensional LP problem:

$$\rho' = \sup_{\mu} \{ \int_{\mathbf{\Omega}} d\mu \ : \ \mu \leq \lambda_{\mathbf{B}}\,;\ \mu \in \mathscr{M}_+(\mathbf{\Omega})\ \}. \tag{1.51}$$

and $\rho' = \mathrm{vol}\,(\mathbf{\Omega})$. A dual of (1.51) is the infinite-dimensional LP:

$$\inf_{g\in\mathbb{R}[\mathbf{x}]} \{ \int_{\mathbf{B}} g\, d\lambda_{\mathbf{B}} \ : \ g \geq 1_{\mathbf{\Omega}} \quad \text{on } \mathbf{B}\ \}, \tag{1.52}$$

with same optimal value as (1.51) as strong duality holds.

**Theorem 1.4.1** *Assume that $b_j > 0$ for all $j \in [m]$ and let Assumption 1.4.1, 1.4.2 and 1.4.3 hold. Then $(\lambda_{\mathbf{\Omega}}, \phi_1, \ldots, \phi_m)$ with $\phi_j$ on $\mathbf{\Omega}_j$ as in (1.48), $j \in [m]$, is the unique optimal solution of LP (1.50) and $\rho = \mathrm{vol}\,(\mathbf{\Omega})$,*

Let $r_j := \lceil d_j/2 \rceil$, for all $j \in [m]$ and let $r^1_{\min} := \max\{1, r_1, \ldots, r_m\}$. In practice, fix an integer $r \geq r^1_{\min}$ and consider the following SDP:

$$\begin{aligned} \rho^r = \sup_{\mathbf{y},\mathbf{v}_j} \quad & \{ y_0 \ : \\ \text{s.t.} \quad & (n+|\alpha|)\, L_{\mathbf{y}}(\mathbf{x}^\alpha) = \sum_{j=1}^m d_j\, b_j\, L_{\mathbf{v}_j}(\mathbf{x}^\alpha), \quad |\alpha| \leq 2r\,, \\ & \mathbf{M}_r(\mathbf{y}^{\mathbf{B}}) \succeq \mathbf{M}_r(\mathbf{y}) \succeq 0\,,\ \mathbf{M}_{r-r_j}((b_j - g_j)\,\mathbf{y}) \succeq 0\,, \\ & \mathbf{M}_r(\mathbf{v}_j) \succeq 0\,,\ \mathbf{M}_{r-r_j}((b_j - g_j)\,\mathbf{v}_j) = 0\,, \quad j \in [m]\,, \\ & \mathbf{M}_{r-r_l}((b_l - g_l)\,\mathbf{v}_j)) \succeq 0\,, \quad l \neq j, \quad l, j \in [m]\,\}\,, \end{aligned} \tag{1.53}$$

where $\mathbf{y} = (y_\alpha)_{\alpha\in\mathbb{N}^n_{2r}}$, and $\mathbf{v}_j = (v_{j,\alpha})_{\alpha\in\mathbb{N}^n_{2r}}$, $j \in [m]$.

The sequence of SDP (1.53) indexed by $r \geq r^1_{\min}$ form a hierarchy of convex relaxations of (1.50) whose size increases with $r$. Let us briefly explain why. The objective function $y_0$ $(= L_{\mathbf{y}}(1))$ corresponds to the mass of the objective function $\int_{\mathbf{\Omega}} d\mu$ of LP (1.50). The equality constraints of (1.53) directly come from the ones in LP (1.50). As mentioned in Section 1.1, if $\mathbf{y}$ has a representing measure $\mu \in \mathscr{M}_+(\mathbf{\Omega})$, then $\mathbf{M}_{r-r_j}((b_j - g_j)\,\mathbf{y}) \succeq 0$. Therefore any solution of LP (1.50) necessarily satisfies the inequality constraints of (1.53), and so $\rho^r \geq \rho$ for all $r \geq r^1_{\min}$.

Next, $(\rho^r)_{r\in\mathbb{N}}$ is a monotone non increasing sequence and the result below shows that by solving the hierarchy of SDP (1.53), then asymptotically one recovers the desired solution.

**Theorem 1.4.2** *Assume that $b_j > 0$ for all $j \in [m]$ and let Assumption 1.4.1, 1.4.2 and 1.4.3 hold. Let $(\phi_1, \ldots, \phi_m)$ be as in (1.48). For each $r \geq r^1_{\min}$, the SDP (1.53) has an optimal solution $(\mathbf{y}^r, \mathbf{v}^r_1, \ldots, \mathbf{v}^r_m)$. In addition:*

$$\lim_{r\to\infty} y^r_\alpha = \int_{\mathbf{\Omega}} \mathbf{x}^\alpha \, d\lambda, \quad \forall \alpha \in \mathbb{N}^n, \tag{1.54}$$

$$\lim_{r\to\infty} v^r_{j,\alpha} = \int_{\mathbf{\Omega}_j} \mathbf{x}^\alpha \, d\phi_j, \quad \forall \alpha \in \mathbb{N}^n; \quad j \in [m]. \tag{1.55}$$

*In particular, as $r \to \infty$, $y^r_0 \downarrow \rho = \mathrm{vol}(\mathbf{\Omega})$.*

For a proof, see [J4, § 7.1].

**Remark 1.4.2** *(a) In (1.53) one may use another criteria different from $y_0$. For instance on may choose to maximize the trace of $\mathbf{M}_r(\mathbf{y})$ which in some cases accelerates the convergence (1.54)-(1.55).*

*(b) In the case where one already knows moments $\mathbf{y} = (y_\alpha)_{\alpha\in\mathbb{N}^n}$ of the Lebesgue measure $\lambda_{\mathbf{\Omega}}$ on $\mathbf{\Omega}$ (e.g. from measurements), then in (1.50) and (1.53), the left-hand-side in the moment equality constraints is now the constant $(n + |\alpha|)\, y_\alpha$. Then in (1.50) one replaces the criterion $\sup \int_{\mathbf{\Omega}} d\mu$ with, e.g., $\sup \sum_j \mu_j(\mathbf{\Omega}_j)$ or $\inf \sum_j \mu_j(\mathbf{\Omega}_j)$. In fact, under Assumption 1.4.3, the feasible set is the singleton $(\phi_1, \ldots, \phi_m)$.*

*The same modification is done in the semidefinite relaxations (1.53) and if one chooses $\sup_{\mathbf{v}_j} \sum_j v_{j,0}$ as criterion then in Theorem 1.4.2, as $d \to \infty$,*

$$\sum_{j=1}^m v^r_{j,0} \downarrow \sum_{j=1}^m \phi_j(\mathbf{\Omega}) = \lim_{r\to\infty} \rho^r.$$

*On the other hand, if one chooses $\inf_{\mathbf{v}_j} \sum_j v_{j,0}$ as criterion then as $r \to \infty$,*

$$\sum_{j=1}^m v^r_{j,0} \uparrow \sum_{j=1}^m \phi_j(\mathbf{\Omega}) = \lim_{r\to\infty} \rho^r.$$

### Step 2: extracting the boundary measure $\sigma_j$ on $\mathbf{\Omega}_j$

from the measure $\phi_j$, for every $j \in [m]$. To do so we use its moments $\mathbf{v}_j = (v_{j,\alpha})_{\alpha\in\mathbb{N}^n}$, obtained in Step 1. For each $j \in [m]$, define the set $\Theta_j \subset \mathbf{\Omega}_j \times \mathbb{R}_+$ by:

$$\Theta_j := \{\, (\mathbf{x}, z) \in \mathbf{\Omega}_j \times \mathbb{R}_+ : \underbrace{z^2 - \|\nabla g_j(\mathbf{x})\|^2}_{\theta_j(\mathbf{x},z)} = 0 \,\}, \quad j \in [m]. \tag{1.56}$$

Observe that if $z^2 = \|\nabla g_j(\mathbf{x})\|^2$ and $z \geq 0$, then $z = \|\nabla g_j(\mathbf{x})\|$. So let $\psi_j$ be a measure on $\Theta_j$ with marginal $\psi_{j,\mathbf{x}} = \phi_j$ on $\mathbf{\Omega}_j$, and conditional $\hat{\psi}_j(dz|\mathbf{x})$ on $\mathbb{R}_+$. Then disintegrating $\psi_j$ yields:

$$\begin{aligned} \int_{\Theta_j} \mathbf{x}^\alpha \, z \, d\psi_j(\mathbf{x}, z) &= \int_{\mathbf{\Omega}_j} \mathbf{x}^\alpha \left( \int_{\mathbb{R}_+} z \, \hat{\psi}_j(dz|\mathbf{x}) \right) d\phi_j(\mathbf{x}) \\ &= \int_{\mathbf{\Omega}_j} \mathbf{x}^\alpha \, \|\nabla g_j(\mathbf{x})\| \, d\phi_j(\mathbf{x}) \qquad (1.57) \\ &= \int_{\mathbf{\Omega}_j} \mathbf{x}^\alpha \, d\sigma_j(\mathbf{x}), \quad \forall \alpha \in \mathbb{N}^n \quad [\text{by } (1.48)]. \end{aligned}$$

Recall that $t_j = \|\nabla g_j(\mathbf{x})\|^2$. Note that $\deg(t_j)/2 = \deg(\theta_j)/2 = d_j - 1$ and let $(v_{j,\alpha})_{\alpha\in\mathbb{N}^n}$ be all moments of $\phi_j$ obtained in step 1. As $\mathbf{\Omega}_j \subseteq (-1,1)^n$ is compact, let us select $M > \sup_{\mathbf{x}\in\mathbf{\Omega}_j} \|\nabla g_j(\mathbf{x})\|^2$. In practice, an easy way to obtain such an upper bound $M$ is to apply interval arithmetic on $\|\nabla g_j(\mathbf{x})\|^2$ or to approximate from above its supremum with the hierarchy of SDP relaxations from [180]. After performing the numerical experiments presented later on, it seems that our method is not sensitive to the accuracy of this approximation. Hence one may and will impose the additional redundant constraint $z^2 \leq M$. Then for each $j \in [m]$, consider the hierarchy of SDP indexed by $r \in \mathbb{N}$:

$$\begin{aligned} \rho^r_{\max,j} = \sup_{\mathbf{u}} \quad & \{ u_{0,1} : \\ \text{s.t.} \quad & |u_{\alpha,0} - v_{j,\alpha}| \leq 1/r, \quad |\alpha| \leq 2r, \\ & \mathbf{M}_r(\mathbf{u}) \succeq 0, \mathbf{M}_{r-d_j+1}(\theta_j\, \mathbf{u}) = 0, \\ & \mathbf{M}_{r-1}((M - z^2)\, \mathbf{u}) \succeq 0, \mathbf{M}_{r-1}(z\, \mathbf{u}) \succeq 0 \}, \end{aligned} \tag{1.58}$$

and

$$\begin{aligned} \rho^r_{\min,j} = \inf_{\mathbf{w}} \quad & \{ w_{0,1} : \\ \text{s.t.} \quad & |w_{\alpha,0} - v_{j,\alpha}| \leq 1/r, \quad |\alpha| \leq 2r, \\ & \mathbf{M}_r(\mathbf{w}) \succeq 0, \mathbf{M}_{r-d_j+1}(\theta_j\, \mathbf{w}) = 0, \\ & \mathbf{M}_{r-1}((M - z^2)\, \mathbf{w}) \succeq 0, \mathbf{M}_{r-1}(z\, \mathbf{w}) \succeq 0 \}, \end{aligned} \tag{1.59}$$

where $\mathbf{u} = (u_{\alpha,k})_{(\alpha,k)\in\mathbb{N}^{n+1}_{2r}}$ and $\mathbf{w} = (w_{\alpha,k})_{(\alpha,k)\in\mathbb{N}^{n+1}_{2r}}$. Let $r^2_{\min} := \max\{1, r_1, \ldots, r_m, L_j\}$.

**Theorem 1.4.3** *Assume that $b_j > 0$ for all $j \in [m]$ and let Assumption 1.4.1, 1.4.2 and 1.4.3 hold.*

*(a) SDP (1.58) (resp. SDP (1.59)) has a feasible solution for each $r \geq r^2_{\min}$ whenever $(v_{j,\alpha}) = (v^{d(r)}_{j,\alpha})_{\alpha\in\mathbb{N}^n_{2r}}$ is an optimal solution of (1.53) at step 1, for sufficiently large $d(r)$.*

*(b) SDP (1.58) (resp. SDP (1.59)) has an optimal solution $\mathbf{u}^r = (u^r_{\alpha,k})$ (resp. $\mathbf{w}^r = (w^r_{\alpha,k})$). In addition:*

$$\lim_{r\to\infty} u^r_{\alpha,1} = \lim_{r\to\infty} w^r_{\alpha,1} = \int_{\mathbf{\Omega}_j} \mathbf{x}^\alpha \, d\sigma_j, \quad \forall \alpha \in \mathbb{N}^n. \tag{1.60}$$

*In particular, $\rho^r_{\max,j} = u^r_{0,1} \downarrow \sigma_j(\mathbf{\Omega})$ and $\rho^r_{\min,j} = w^r_{0,1} \uparrow \sigma_j(\mathbf{\Omega})$, as $r \to \infty$.*

The proof can be found in [J4, § 7.2].

**Remark 1.4.3** *If $\mathbf{v}_j = (v_{j,\alpha})_{\alpha\in\mathbb{N}^n_{2r}}$ is the exact vector of moments of $\phi_j$ on $\mathbf{\Omega}_j$ (up to degree $2r$), then (1.58) has a feasible solution even with the stronger constraint $u_{\alpha,0} = v_{j,\alpha}$ for all $\alpha \in \mathbb{N}^n_{2r}$. It suffices to consider the moments $\mathbf{u} = (u_{\alpha,k})_{(\alpha,k)\in\mathbb{N}^{n+1}_{2r}}$ of the measure $d\phi(\mathbf{x}, z) = d\delta_{\|\nabla g_j(\mathbf{x})\|}(z) d\phi_j(\mathbf{x})$, where $\delta_\bullet$ is the Dirac measure. Indeed such a vector $\mathbf{u}$ is feasible by construction. In fact in this case, an infinite sequence $\mathbf{u} = (u_{\alpha,k})_{(\alpha,k)\in\mathbb{N}^{n+1}}$ that satisfies* all *constraints of (1.58) is unique and is the moment sequence of the measure $d\delta_{\|\nabla g_j(\mathbf{x})\|}(z)\, d\phi_j(\mathbf{x})$, and so (1.60) holds. However, the weaker constraint $|u_{\alpha,0} - v_{j,\alpha}| \leq 1/r$ allows to handle the practical case of approximate solutions obtained in step 1, and one still obtains the desired convergence result. In our numerical experiments, one could successfully solve (1.58) after replacing the inequality $|u_{\alpha,0} - v_{j,\alpha}| \leq 1/r$ by the equality $u_{\alpha,0} = v_{j,\alpha}$. Note that the former inequality can be interpreted as a perturbation of the latter equality. In general, such perturbations always occur while using numerical SDP solvers. We refer the interested reader to [J5], outlined in Section 2.1, for more details on the problem of interpreting wrong results (due to numerical inaccuracies) observed when solving SDP relaxations for polynomial optimization on a double precision floating point SDP solver.*

Theorem 1.4.3 states that by solving the hierarchy of SDP (1.58), one may approximate as closely as desired any finite number of moments of the boundary measure $\sigma_j$. In addition, depending on whether one maximizes or minimizes the same criterion $u_{0,1}$, one obtains a monotone sequence of upper bounds or lower bounds that converges to $\sigma_j(\mathbf{\Omega}_j)$. After summing up, this allows to obtain closer and closer approximations

$$\sum_{j=1}^{m} \rho^r_{\min,j} \;\leq\; \sigma(\partial\mathbf{\Omega}) \;\leq\; \sum_{j=1}^{m} \rho^r_{\max,j}$$

of the mass of $\partial\mathbf{\Omega}$ (length if $n=2$ or area if $n=3$).

## Discussion

One may also approximate moments of the $\sigma_j$'s by solving a *single* hierarchy that combines the constraints of (1.53) and (1.58), that is, by solving the following hierarchy of SDP indexed by $r \in \mathbb{N}$:

$$\begin{array}{rll} \rho^r = & \sup\limits_{\mathbf{y},\mathbf{v}_j,\mathbf{u}^j} & \{ y_0 : \\ & \text{s.t.} & (n+|\alpha|)\, L_{\mathbf{y}}(\mathbf{x}^\alpha) = \sum\limits_{j=1}^{m} d_j\, b_j\, L_{\mathbf{v}_j}(\mathbf{x}^\alpha), \quad |\alpha| \leq 2r\,, \\ & & \mathbf{M}_r(\mathbf{y}^{\mathbf{B}}) \succeq \mathbf{M}_r(\mathbf{y}) \succeq 0\,,\, \mathbf{M}_{r-r_j}((b_j - g_j)\,\mathbf{y}) \geq 0\,, \\ & & \mathbf{M}_r(\mathbf{v}_j) \succeq 0\,,\, \mathbf{M}_{r-r_j}((b_j - g_j)\,\mathbf{v}_j) = 0\,, \quad j \leq m\,, \\ & & \mathbf{M}_{r-r_l}((b_l - g_l)\,\mathbf{v}_j)) \succeq 0\,, \quad l \neq j\,, \quad j \leq m \\ & & \\ & & u^j_{\alpha,0} = v_{j,\alpha}\,, \qquad |\alpha| \leq 2r\,,\, j \leq m\,, \\ & & \mathbf{M}_r(\mathbf{u}^j) \,\succeq\, 0\,,\, \mathbf{M}_{r-L_j}(\theta_j\,\mathbf{u}^j) \,=\, 0\,, \quad j \leq m\,, \\ & & \mathbf{M}_{r-1}((M - z^2)\,\mathbf{u}^j) \succeq 0\,,\, \mathbf{M}_{r-1}(z\,\mathbf{u}^j) \succeq 0\,, \quad j \leq m\,\}\,. \end{array} \tag{1.61}$$

Indeed recall Remark 1.4.3. If $\mathbf{y} = (y_\alpha)_{\alpha\in\mathbb{N}}$ is the moment sequence of $\lambda_{\mathbf{\Omega}}$ then an infinite sequence $\mathbf{u}^j = (u^j_{\alpha,k})_{(\alpha,k)\in\mathbb{N}^{n+1}}$ that satisfies all constraints of (1.58), after replacing each inequality $|u_{\alpha,0} - v_{j,\alpha}| \leq 1/r$ by the equality $u_{\alpha,0} - v_{j,\alpha} = 0$, for all $r \in \mathbb{N}$, is unique and is the moment sequence of the measure $d\delta_{\|\nabla g_j(\mathbf{x})\|}(z)\, d\phi_j(\mathbf{x})$ on $\mathbf{\Omega}_j$.

So the SDP (1.61) has always a feasible solution. However it has $m$ additional unknown moment sequences $(\mathbf{u}^j)_{j\leq m}$, hence is harder than (1.53) to solve, and the numerical results can be less accurate. If $m$ is small it still may be an interesting alternative to the two-step procedure. On the other hand, one looses the upper and lower bounds $\rho^r_{\max,j}$ and $\rho^r_{\min,j}$ obtained in (1.58) and (1.59) respectively.

## Numerical experiments

Here, we illustrate our theoretical framework on simple basic compact semialgebraic sets, contained in the unit box $\mathbf{B} := (-1,1)^n$ (possibly after proper scaling). Our numerical experiments are performed with the GloptiPoly [132] library. We used SeDuMi 1.3 [279] to solve the SDP problems. Our code is available online[3].

We first consider the two-dimensional unit disk $\mathbf{\Omega} = \{\mathbf{x} \in \mathbb{R}^2 : \|\mathbf{x}\|_2^2 \leq 1\}$, corresponding to $g_1 = \|\mathbf{x}\|_2^2$ and $b_1 = 1$. The moments of the Hausdorff boundary measure of $\partial\mathbf{\Omega}$ are approximated after solving SDP (1.53) (first step) and SDP (1.58)-(1.59) (second step), successively. The second step allows one to compute the absolute error gap between the optimal value $\rho^r_{\max,1}$ of SDP (1.58)

[3] http://homepages.laas.fr/vmagron/boundary.tar.gz

and $\rho^r_{\min,1}$ of SDP (1.59), which respectively provide an upper bound and a lower bound on the perimeter of the boundary of $\mathbf{\Omega}$. To ensure that the moments of the uniform measure on $\mathbf{\Omega}$ are approximated with good accuracy, we solve the first step with relatively large value of $d$, namely with $d = 10$. These moment approximations are then used as input in the SDP relaxations related to the second step. At $d = 2$, we already obtain $\rho^r_{\max,1} = 6.28319 \gtrsim 2\pi \gtrsim \rho^r_{\min,1} = 6.28317$.

Then, we provide results for the same problem $\mathbf{\Omega} = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_2^2 \leq 1\}$ in larger dimensions $n = 3, 4, 5$, by solving the first and second steps with the same value of $r$. Table 1.1 indicates that the relative error regularly decreases when $r$ increases. These numerical results also show that it is more difficult to obtain accurate bounds, as the moments of the uniform measure on the unit ball are approximated with less good precision. The symbol "$-$" in a column entry means that the SDP solver runs out of memory when trying to solve the corresponding problem. This is a consequence of the fact that state-of-the-art SDP solvers are limited to solve relaxations of POP of modest size (e.g., with $n + r \leq 10$ on standard laptops) as these relaxations involve a number of variables proportional to $\binom{n+2r}{n}$ and matrices of size proportional to $\binom{n+r}{n}$.

Table 1.1: Relative and absolute errors obtained for the mass approximation of the Hausdorff measure on the unit sphere.

| $n$ | $r$ | 2 | 4 | 6 | 8 |
|---|---|---|---|---|---|
| 3 | relative error | 11.9 % | 1.51 % | 0.16 % | 0.02 % |
| | absolute error | 1.50 | 0.19 | 0.02 | 1.9e-3 |
| 4 | relative error | 38.6 % | 7.59 % | 1.36 % | $-$ |
| | absolute error | 7.62 | 1.50 | 0.27 | $-$ |
| 5 | relative error | 95.5 % | 24.0 % | $-$ | $-$ |
| | absolute error | 25.2 | 6.31 | $-$ | $-$ |

For comparison purpose, we perform the "reverse" experiment of the first one, namely we consider the square of length $\sqrt{2}$ given by $\mathbf{\Omega} = \{\mathbf{x} \in \mathbb{R}^2 : \pm x_1 \leq \frac{1}{\sqrt{2}}, \pm x_2 \leq \frac{1}{\sqrt{2}}\}$, contained in the unit disk $\mathbf{B} = \{\mathbf{x} \in \mathbb{R}^2 : \|\mathbf{x}\|_2^2 \leq 1\}$. Here, the set $\mathbf{\Omega}$ is defined by four inequality constraints so SDP (1.53) involves four sequences of variables $\mathbf{v}_1, \dots, \mathbf{v}_4$ instead of a single one. Here, we only need to perform the first step as $\|\nabla g_j\| = 1$, so $r\phi_j = \frac{d\sigma_j}{\|\nabla g_j\|} = r\sigma_j$, for each $j \in [4]$. We obtain at $r = 1$ a very good approximation $\rho^1 = 5.6498$ for the exact perimeter $4 \times \sqrt{2} \simeq 5.6569$.

As for the unit disk, we repeat the same experiments on the so-called "TV screen", defined by $\mathbf{\Omega} = \{\mathbf{x} \in \mathbb{R}^2 : x_1^4 + x_2^4 \leq 1\}$. The approximate value of the perimeter of the boundary is given by numerical integration of $2 \times \int_{-1}^1 \sqrt[4]{1 - t^4} dt \simeq 7.0177$. In Table 1.2, we display the relative errors in percentage when approximating the perimeter of the boundary of $\mathbf{\Omega}$, namely the mass of the boundary measure, for increasing values of the relaxation order $d$. We also display the absolute error gap between the optimal value $\rho^r_{\max,1}$ of SDP (1.58) and $\rho^r_{\min,1}$ of SDP (1.59). Table 1.2 indicates that the quality of the approximations increases significantly when the relaxation order grows. We also implemented SDP (1.61), i.e., the relaxation corresponding to a single hierarchy. In this case, the approximation of the perimeter is less accurate as we obtain a relative error of 17.9% for $r = 3$, 5.98% for $r = 4$ and 5.65% for $r = 5$. With higher relaxation orders, we encountered numerical issues, certainly due to the growing number of SDP constraints and SDP variables.

Eventually, we consider the non-convex two-dimensional "star-shaped" curve, defined to be the boundary of $\mathbf{\Omega} = \{\mathbf{x} \in \mathbb{R}^2 : x_1^4 + x_2^4 - 1.7x_1^2x_2^2 \leq 0.2\}$, displayed in Figure 1.9. Again, using numerical integration scheme, we obtain an approximate value of the perimeter equal to 7.5055. By contrast with the two previous examples, Table 1.3 shows that this is slightly more difficult to obtain accurate approximations for the mass of the boundary measure of this non-convex set, as we need to compute the fifth order relaxation to get a relative error below 0.5 %.

Table 1.2: Relative and absolute errors obtained for the mass approximation of the Hausdorff measure on the boundary of the TV screen.

| $r$ | 3 | 4 | 5 |
|---|---|---|---|
| relative error | 0.18 % | 0.02 % | 0.01 % |
| absolute error | 4.37 | 1.5e-3 | 1.4e-5 |

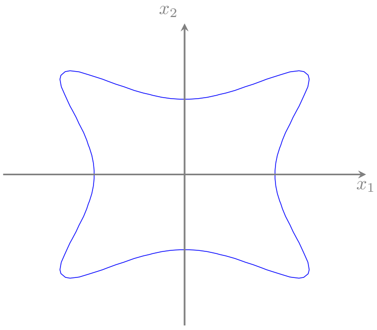


Figure 1.9: The non-convex two-dimensional "star-shaped" curve, given by $\partial\mathbf{\Omega} = \{\mathbf{x} \in \mathbb{R}^2 : x_1^4 + x_2^4 - 1.7x_1^2x_2^2 = 0.2\}$.

Table 1.3: Relative and absolute errors obtained when approximating the mass of the Hausdorff measure on the boundary of a "star-shaped" curve.

| $r$ | 3 | 4 | 5 |
|---|---|---|---|
| relative error | 1.91% | 1.19% | 0.49% |
| absolute error | 6.77 | 0.68 | 0.25 |

## 1.5 Optimization over trace polynomials

The goal of this section is to solve the class of POP with noncommuting variables (e.g. finite/infinite size matrices) and coefficients being some of their trace products. Applications of interest arise from quantum theory and quantum information science [224, 241] as well as control theory [270, 230]. Further motivation relates to the generalized Lax conjecture [193], where the goal is to obtain computer-assisted proofs based on noncommutative SOS in Clifford algebras [225]. The verification of noncommutative polynomial trace inequalities has also been motivated by a conjecture formulated by Bessis, Moussa and Villani (BMV) in 1975 [40], which has been recently proved by Stahl [277] (see also the Lieb and Seiringer reformulation [197]). Further efforts focused on applications arising from bipartite quantum correlations [104], and matrix factorization ranks in [103]. In a related analytic direction, there has been recent progress on multivariate generalizations of the Golden-Thompson inequality and the Araki-Lieb-Thirring inequality [280, 138].

There is a plethora of prior research in quantum information theory involving reformulating problems as optimization of noncommutative polynomials. One famous application is to characterize the set of quantum correlations. Bell inequalities [31] provide a method to investigate entanglement, which allows two or more parties to be correlated in a non-classical way, and is often studied through the set of bipartite quantum correlations. Such correlations consist of the conditional probabilities that two physically separated parties can generate by performing measurements on a shared entangled state. These conditional probabilities satisfy some inequalities classically, but violate them in the quantum realm [65].

In the free noncommutative context (i.e., without traces), a polynomial is positive semidefinite if and only if it can be written as a sum of Hermitian squares (SOHS) [123, 212]. One can rely on such SOHS decompositions to perform eigenvalue optimization of noncommutative polynomials over noncommutative semialgebraic sets, i.e., under noncommutative polynomial inequality constraints. The noncommutative analogues of Lasserre's hierarchy [126, 224, 238, 57, 54] allow one to approximate as closely as desired the optimal value of such eigenvalue minimization problems. In [224], Navascués, Pironio and Acín provide a way to compute bounds on the maximal violation levels of Bell inequalities: they first reformulate the initial problem as an eigenvalue optimization one and then approximate its solution with a converging hierarchy of SDP, based on the noncommutative version of Putinar's Positivstellensatz due to Helton and McCullough [126]. This is the so-called Navascués-Pironio-Acín (NPA for short) hierarchy and can be viewed as the "eigenvalue" version of Lasserre's hierarchy. This leads to a hierarchy of upper bounds on the maximum violation level of Bell inequalities (see also [81, 232]). Further extensions [238, 57, 54] have been provided to optimize the trace of a given polynomial under positivity constraints. `NCSOStools` [58, 53] can compute lower bounds on minimal eigenvalues or traces of noncommutative polynomial objective functions over noncommutative semialgebraic sets.

This work greatly extends these frameworks to the case of optimization problems involving *trace polynomials*, i.e., polynomials in symmetric noncommutative variables $x_1, \dots, x_n$ and traces of their products. Thus naturally each trace polynomial has an adjoint. A *pure trace polynomial* is a trace polynomial that is made only of traces, i.e., has no free variables $x_j$. For instance, the trace of a trace polynomial is a pure trace polynomial, e.g.,

$$\begin{aligned} f &= x_1 x_2 x_1^2 - \operatorname{tr}(x_2)\operatorname{tr}(x_1 x_2)\operatorname{tr}(x_1^2 x_2) x_2 x_1, \\ \operatorname{tr}(f) &= \operatorname{tr}(x_1^3 x_2) - \operatorname{tr}(x_2)\operatorname{tr}(x_1 x_2)^2 \operatorname{tr}(x_1^2 x_2), \\ f^\star &= x_1^2 x_2 x_1 - \operatorname{tr}(x_2)\operatorname{tr}(x_1 x_2)\operatorname{tr}(x_1^2 x_2) x_1 x_2. \end{aligned}$$

The variables $x_1$ and $x_2$ can be both quantum physics operators. One important underlying motivation is that trace polynomials are involved in several problems arising from quantum information theory. For instance, [93] presents a framework to obtain the limit output states for a large class of input states having specific sets of parameters. To obtain these limits, one needs to compute bounds for generalized traces of tensors. One way to model such generalized traces is to consider a reformulation as an optimization problem involving trace polynomials. In this problem, trace polynomials arise as cost functions but they can also appear in the constraints. Convex relaxations of trace polynomial problems can be obtained as in the NPA hierarchy: one can associate a new variable to each word trace (e.g. $\operatorname{tr}(x_1)$, $\operatorname{tr}(x_2)$, $\operatorname{tr}(x_1 x_2)$, etc. in the above example), then incorporate the initial constraints into the semidefinite matrix defined in the NPA hierarchy. Moreover such noncommuting operators in [241], fulfill causal constraints, which leads to equality constraints. This results in a so-called *scalar extension* of the NPA hierarchy, which allows the authors to successfully identify correlations not attainable in the entanglement-swapping scenario. However, [241] does not provide a proof of convergence for this hierarchy. In [142], the author focuses on the multilinear case and obtains a characterization of all multilinear equivariant trace polynomials which are positive on the positive cone. In a closely related work in real algebraic geometry [155], the first and third author derive several Positivstellensätze for trace polynomials positive on semialgebraic sets of *fixed size* matrices. In particular, [155] establishes a Putinar-type Positivstellensatz stating that any positive polynomial admits a weighted SOHS decomposition without denominators. In the dimension-free setting, finite von Neumann algebras and their tracial states provide a natural framework for studying tracial polynomial inequalities. This work characterizes trace polynomials which are positive on tracial semialgebraic sets, where the initial polynomials and constraints involve freely noncommutative variables and traces, and the evaluations are performed on von Neumann algebras.

## Noncommutative polynomials and trace polynomials

Let us denote by $\mathbb{M}_k$ (resp. $\mathbb{S}_k$) the space of all real (resp. symmetric) matrices of order $k$. The normalized trace of a matrix $A \in \mathbb{M}_k$ is given by $\operatorname{tr} A = \frac{1}{k}\sum_{i=1}^{k} a_{i,i}$. For a fixed $n \in \mathbb{N}$, we consider a finite alphabet $x_1, \dots, x_n$ and generate all possible words of finite length in these letters. The empty word is denoted by 1. The resulting set of words is the *free monoid* $\langle \underline{x} \rangle$, with $\underline{x} = (x_1, \dots, x_n)$. We denote by $\mathbb{R}\langle \underline{x} \rangle$ the set of real polynomials in noncommutative variables, abbreviated as *nc polynomials*. The algebra $\mathbb{R}\langle \underline{x} \rangle$ is equipped with the involution $\star$ that fixes $\mathbb{R} \cup \{x_1, \dots, x_n\}$ point-wise and reverses words, so that $\mathbb{R}\langle \underline{x} \rangle$ is the $\star$-algebra freely generated by $n$ symmetric letters $x_1, \dots, x_n$.

We now introduce some algebraic terminology to deal with the trace, following [244] (see also [154, 155]). We denote by $\mathrm{T}$ the commutative polynomial algebra in infinitely many variables $\operatorname{tr}(w)$ with $w \in \langle \underline{x} \rangle$, up to $\star$-cyclic equivalence, that is, $\mathrm{T} := \mathbb{R}[\operatorname{tr}(w)\,, w \in \langle \underline{x} \rangle / \mathrm{cyc}^\star]$. We also let $\mathbb{T} := \mathrm{T}\langle \underline{x} \rangle$ be the free $\mathrm{T}$-algebra on $\underline{x}$. Elements of $\mathrm{T}$ are called *pure trace polynomials*, and elements of $\mathbb{T}$ are *trace polynomials*. For example, $t = \operatorname{tr}(x_1^2) - \operatorname{tr}(x_1)^2 \in \mathrm{T}$ and $x_1^2 - \operatorname{tr}(x_1)x_1 - 2t \in \mathbb{T} = \mathrm{T}\langle x_1 \rangle$. The involution on $\mathbb{T}$, denoted also by $\star$, fixes $\{x_1, \dots, x_n\} \cup \mathrm{T}$ point-wise, and reverses words from $\langle \underline{x} \rangle$. The set of all *symmetric elements* of $\mathbb{T}$ is defined as $\operatorname{Sym} \mathbb{T} := \{f \in \mathbb{T} : f = f^\star\}$. A linear functional $L : \mathbb{T} \to \mathbb{R}$ is said to be *tracial* if $L(\operatorname{tr}(f)) = L(f)$ for all $f \in \mathbb{T}$. We also consider the universal trace map $\tau$ defined by

$$\begin{aligned} \tau : \mathbb{T} &\to \mathrm{T}\,, \\ f &\mapsto \operatorname{tr}(f)\,. \end{aligned}$$

A linear functional $L : \mathbb{T} \mapsto \mathbb{R}$ is tracial if and only if $L \circ \tau = L$. Such an $L$ is determined by $L|_{\mathrm{T}} : \mathrm{T} \to \mathbb{R}$ being an (arbitrary) linear functional. The functional $L$ is called *unital* if $L(1) = 1$ and is called *symmetric* if $L(f^\star) = L(f)$, for all $f$ belonging to the domain of $L$.

## Tracial semialgebraic sets and von Neumann algebras

Given $S \subseteq \operatorname{Sym} \mathbb{T}$, the *matricial tracial semialgebraic set* $\mathcal{D}_S$ associated to $S$ is defined as follows:

$$\mathcal{D}_S := \bigcup_{k \in \mathbb{N}} \{\underline{A} = (A_1, \dots, A_n) \in \mathbb{S}_k^n : g(\underline{A}) \succeq 0 \text{ for all } g \in S\}\,. \tag{1.62}$$

While (1.62) looks like a natural candidate for testing positivity of tracial polynomials, the failure of Connes' embedding conjecture [147] hinders the existence of a reasonable Positivstellensatz for (1.62) by [153]. Instead of just matrices of all finite sizes, one is thus led to include bounded operators, similarly as in the trace-free setting [126]. Since we deal with tracial constraints, the considered bounded operators need to admit traces. The natural framework is therefore given by tracial von Neumann algebras, which we discuss next.

A real von Neumann algebra $\mathcal{F}$ [15] is a unital, weakly closed, real, self-adjoint subalgebra of the (real) algebra of bounded linear operators on a complex Hilbert space, with the property $\mathcal{F} \cap \mathbf{i}\mathcal{F} = \{0\}$ (where $\mathbf{i}$ denotes the imaginary unit). We restrict ourselves to separable Hilbert spaces, implying that all von Neumann algebras have separable preduals. Much of the structure theory of real von Neumann algebras can be transfered from complex von Neumann algebras [284, Chapter 5]. Namely, the complexification of a real von Neumann algebra yields a complex von Neumann algebra with an involutory $*$-antiautomorphism; conversely, the fixed set of an involutory $*$-antiautomorphism on a complex von Neumann algebra is a real von Neumann algebra. A real von Neumann algebra is *finite* if in its complexification, every isometry is a unitary. By [284, Theorem 2.4], a von Neumann algebra is finite if and only if it admits sufficiently many normal tracial states, which will play an important role in this article.

A (real) von Neumann algebra is a factor if its center consists of only the (real) scalar operators. By [284, Theorem 2.6], a factor is finite if and only if it admits a faithful normal tracial state; in this case, such a state is unique, and is called the *trace* of the factor. Finally, a $\mathrm{II}_1$-*factor* is an infinite-dimensional finite factor (other finite factors are of type $\mathrm{I}_n$, which are $n \times n$ complex matrices in the complex setting, and $n \times n$ real matrices or $\frac{n}{2} \times \frac{n}{2}$ quaternion matrices in the real setting). In this article we consider positivity on operator semialgebraic sets. These are defined as follows (cf. [53, Definition 1.59]):

**Definition 1.5.1** *A tracial pair $(\mathcal{F}, \tau)$ consists of a real finite von Neumann algebra $\mathcal{F}$ and a faithful normal tracial state $\tau$ on $\mathcal{F}$ [284, Chapter 5].*

*Given $S \subseteq \operatorname{Sym} \mathbb{T}$ let $\mathcal{D}_S^{\mathcal{F},\tau}$ be the set of all self-adjoint tuples $\underline{X} = (X_1, \dots, X_n) \in \mathcal{F}^n$ making $g(\underline{X})$ a positive semidefinite operator for every $g \in S$; here $\operatorname{tr}$ is evaluated as $\tau$. The von Neumann semialgebraic set $\mathcal{D}_S^{vN}$ generated by $S$ is defined as*

$$\mathcal{D}_S^{vN} := \bigcup_{(\mathcal{F},\tau)} \mathcal{D}_S^{\mathcal{F},\tau},$$

*where the union is over all tracial pairs $(\mathcal{F}, \tau)$. Analogously, we define*

$$\mathcal{D}_S^{\mathrm{II}_1} := \bigcup_{\mathcal{F}} \mathcal{D}_S^{\mathcal{F}},$$

*where the union is over all $\mathrm{II}_1$-factors (which come equipped with unique traces).*

Note that finiteness of $S$ is not needed at this stage. Unlike in the free case [124], these tracial semialgebraic sets are closed neither under direct sums nor reducing subspace compressions; for example, if $g = \operatorname{tr}(x_1)\operatorname{tr}(x_2)$, then

$$\begin{aligned} &g(3,1) > 0 \quad \text{and} \quad g(-1,-2) > 0, \quad \text{but} \quad g(3 \oplus -1, 1 \oplus -2) < 0; \\ &g(-2 \oplus 1, 1 \oplus -2) > 0, \quad \text{but} \quad g(-2,1) < 0 \quad \text{and} \quad g(1,-2) < 0. \end{aligned}$$

To sidestep this technical problem we make use of the following well-known fact that is all but stated in [85, Theorem 2.5].

**Proposition 1.5.2** *Every tracial pair embeds into a $\mathrm{II}_1$-factor.*

## Non-cyclic Positivstellensatz for pure trace polynomials

In this section we provide our first Positivstellensatz, Theorem 1.5.1, for pure trace polynomials based on quadratic modules from real algebraic geometry [210]. Given an archimedean quadratic module $\mathcal{M} \subseteq \mathrm{T}$ (in the usual commutative sense, meaning that for each $f \in \mathrm{T}$ there is $N > 0$ such that $N \pm f \in \mathcal{M}$), we consider the real points of the real spectrum $\operatorname{Sper}_{\mathcal{M}} \mathrm{T}$, namely the set $\chi_{\mathcal{M}}$ defined by

$$\chi_{\mathcal{M}} := \{\varphi : \mathrm{T} \to \mathbb{R} \mid \varphi \text{ homomorphism}, \varphi(\mathcal{M}) \subseteq \mathbb{R}_{\geq 0}, \varphi(1) = 1\}. \tag{1.63}$$

The next proposition is the well-known Kadison-Dubois representation theorem, see, e.g., [210, Theorem 5.4.4].

**Proposition 1.5.3** *Let $\mathcal{M} \subseteq \mathrm{T}$ be an archimedean quadratic module. Then, for all $a \in \mathrm{T}$, one has*

$$\forall \varphi \in \chi_{\mathcal{M}} \quad \varphi(a) \geq 0 \qquad \Leftrightarrow \qquad \forall \varepsilon > 0 \quad a + \varepsilon \in \mathcal{M}.$$

A homomorphism $\varphi \to \mathbb{R}$ is determined by the "tracial moments" $\varphi(\operatorname{tr}(w))$ for $w \in \langle \underline{x} \rangle$. In this sense, the following variant of [113, Theorem 1.3] is a solution of the tracial moment problem. In the given formulation, it is the dimension-free analog of of the extension theorem [155, Theorem 4.8].

**Proposition 1.5.4** *Let $\varphi : \mathrm{T} \to \mathbb{R}$ be a homomorphism. Then there are a tracial pair $(\mathcal{F}, \tau)$ and $\underline{X} = \underline{X}^* \in \mathcal{F}^n$ such that $\varphi(a) = a(\underline{X})$ for all $a \in \mathrm{T}$ if and only if the following holds:*

(a) $\varphi(\mathrm{tr}(pp^\star)) \geq 0$ *for all* $p \in \mathbb{R}\langle \underline{x} \rangle$;

(b) $\liminf_{k\to\infty} \sqrt[2k]{\varphi(\mathrm{tr}(x_j^{2k}))} < \infty$ *for* $j \in [n]$.

**Definition 1.5.5** *Given $S \subseteq \mathrm{T}$ and $N > 0$ let*

$$S(N) := S \cup \{\mathrm{tr}(pp^\star) \mid p \in \mathbb{R}\langle \underline{x} \rangle\} \cup \{N^k - \mathrm{tr}(x_j^{2k}) \mid 1 \leq j \leq n,\ k \in \mathbb{N}\} \subseteq \mathrm{T}. \tag{1.64}$$

*For $S \subseteq \operatorname{Sym} \mathbb{T}$ let*

$$S[N] := S \cup \{N - x_j^2 \mid j \in [n]\} \subset \mathbb{T}. \tag{1.65}$$

**Lemma 1.5.6** *The quadratic module $\mathcal{M}(S(N)) \subseteq \mathrm{T}$ is archimedean for every $S, N$.*

We are now ready to prove our first theorem, the purely tracial analog of the noncommutative Helton-McCullough archimedean Positivstellensatz [126].

**Theorem 1.5.1** *Let $S \subseteq \mathrm{T}$ and $N > 0$ be given. Then for $a \in \mathrm{T}$ the following are equivalent:*

(i) $a(\underline{X}) \geq 0$ *for all* $\underline{X} \in \mathcal{D}_{S[N]}^{vN}$;

(ii) $a(\underline{X}) \geq 0$ *for all* $\underline{X} \in \mathcal{D}_{S[N]}^{\mathrm{II}_1}$;

(iii) $a + \varepsilon \in \mathcal{M}(S(N))$ *for all* $\varepsilon > 0$.

Since $\mathcal{M}(S(N_1)) \subseteq \mathcal{M}(S(N_2))$ for $N_1 \geq N_2$, we obtain the following consequence.

**Corollary 1.5.7** *Let $S \subset \mathrm{T}$ and $a \in \mathrm{T}$. The following are equivalent:*

(i) $a(\underline{X}) \geq 0$ *for all* $\underline{X} \in \mathcal{D}_S^{vN}$;

(ii) $a(\underline{X}) \geq 0$ *for all* $\underline{X} \in \mathcal{D}_S^{\mathrm{II}_1}$;

(iii) $a + \varepsilon \in \mathcal{M}(S(N))$ *for all* $\varepsilon > 0$ *and* $N \in \mathbb{N}$.

## Cyclic Positivstellensatz for trace polynomials

In this section we prove a Positivstellensatz for trace polynomials that is less inspired by the commutative theory than the one from above and relies more on the tracial structure of trace polynomials. First we introduce the notion of a cyclic quadratic module. A subset $\mathcal{M}^{\mathrm{cyc}} \subseteq \operatorname{Sym} \mathbb{T}$ is called a *cyclic quadratic module* if

$$1 \in \mathcal{M}^{\mathrm{cyc}},\ \mathcal{M}^{\mathrm{cyc}} + \mathcal{M}^{\mathrm{cyc}} \subseteq \mathcal{M}^{\mathrm{cyc}},\ a^\star \mathcal{M}^{\mathrm{cyc}} a \subseteq \mathcal{M}^{\mathrm{cyc}}\ \forall a \in \mathbb{T},\ \mathrm{tr}(\mathcal{M}^{\mathrm{cyc}}) \subset \mathcal{M}^{\mathrm{cyc}}.$$

Given $S \subset \mathbb{T}$ let $\mathcal{M}^{\mathrm{cyc}}(S)$ be the cyclic quadratic module generated by $S$, i.e., the smallest cyclic quadratic module in $\mathbb{T}$ containing $S$. A cyclic quadratic module $\mathcal{M}^{\mathrm{cyc}}$ is called *archimedean* if for all $a \in \operatorname{Sym} \mathbb{T}$ there exists $N > 0$ such that $N - a \in \mathcal{M}^{\mathrm{cyc}}$. We start with a few preliminary results.

**Lemma 1.5.8** *Let $S \subset \mathbb{T}$.*

(1) *Elements of* $\mathcal{M}^{\mathrm{cyc}}(\varnothing)$ *are precisely sums of*

$$\mathrm{tr}(h_1h_1^\star)\cdots\mathrm{tr}(h_lh_l^\star)h_0h_0^\star$$

*for* $h_i \in \mathbb{T}$.

(2) *Elements of* $\mathcal{M}^{\mathrm{cyc}}(S)$ *are precisely sums of*

$$q_1, \quad h_1g_1h_1^\star, \quad \mathrm{tr}(h_2g_2h_2^\star)q_2$$

*for* $h_i \in \mathbb{T}$, $q_i \in \mathcal{M}^{\mathrm{cyc}}(\varnothing)$, $g_i \in S$.

(3) *Elements of* $\mathrm{tr}(\mathcal{M}^{\mathrm{cyc}}(S)) = \mathcal{M}^{\mathrm{cyc}}(S) \cap \mathrm{T}$ *are precisely sums of*

$$\mathrm{tr}(h_1h_1^\star)\cdots\mathrm{tr}(h_lh_l^\star)\,\mathrm{tr}(h_0gh_0^\star)$$

*for* $h_i \in \mathbb{T}$ *and* $g \in S$.

**Proposition 1.5.9** *A cyclic quadratic module* $\mathcal{M}^{\mathrm{cyc}}$ *is archimedean if and only if there exists* $N \in \mathbb{N}$ *such that* $N - \sum_{j=1}^n x_j^2 \in \mathcal{M}^{\mathrm{cyc}}$.

**Proposition 1.5.10** *Let* $(\mathcal{F}, \tau)$ *be a tracial pair and* $X = X^* \in \mathcal{F}$. *The following are equivalent:*

(i) $X \succeq 0$;

(ii) $\tau(XY) \geq 0$ *for all positive semidefinite contractions* $Y \in \mathcal{F}$;

(iii) $\tau(Xp(X)^2) \geq 0$ *for all* $p \in \mathbb{R}[t]$.

The following is the cyclic version of the Helton-McCullough theorem [126]. Note that while the constraints in Theorem 1.5.2 are arbitrary trace polynomials, the objective function needs to be a pure trace polynomial. A direct analog for non-pure trace objective polynomials fails, see Example 1.5.1 below.

**Theorem 1.5.2** *Let* $\mathcal{M}^{\mathrm{cyc}} \subseteq \mathrm{Sym}\,\mathbb{T}$ *be an archimedean cyclic quadratic module and* $a \in \mathrm{T}$. *The following are equivalent:*

(i) $a(\underline{X}) \geq 0$ *for all* $\underline{X} \in \mathcal{D}^{vN}_{\mathcal{M}^{\mathrm{cyc}}}$;

(ii) $a(\underline{X}) \geq 0$ *for all* $\underline{X} \in \mathcal{D}^{\mathrm{II}_1}_{\mathcal{M}^{\mathrm{cyc}}}$;

(iii) $a + \varepsilon \in \mathcal{M}^{\mathrm{cyc}}$ *for all* $\varepsilon > 0$.

For the reader unfamiliar with real algebraic geometry and noncommutative moment problems, we refer to the appendix of [R3] for a self-contained proof of Theorem 1.5.2 relying only on convex separation results and basic properties of von Neumann algebras.

Given a set of symmetric polynomials $S \subset \mathbb{R}\langle \underline{x} \rangle$ let $\mathcal{M}(S)$ denote the (free) quadratic module generated by $S$ [53, Section 1.4]. Hence $\mathcal{M}(S)$ is the smallest set that contains $S \cup \{1\}$, is closed under addition, and $f \in \mathcal{M}(S)$ implies $hfh^\star \in \mathcal{M}(S)$ for every $h \in \mathbb{R}\langle \underline{x} \rangle$.

**Lemma 1.5.11** *Let* $S_1 \subset \mathrm{T}$ *and* $S_2 \subset \mathbb{R}\langle \underline{x} \rangle$. *If* $g(0) \geq 0$ *for all* $g \in S_1$, *then*

$$\mathcal{M}^{\mathrm{cyc}}(S_1 \cup S_2) \cap \mathbb{R}\langle \underline{x} \rangle = \mathcal{M}(S_2).$$

**Example 1.5.1** *Let $n = 1$. Let $\mathcal{M}^{\text{cyc}}$ be the archimedean cyclic quadratic module in* $\operatorname{Sym}\mathbb{T}$ *generated by*

$$\{1 - x_1^2\} \cup \{\operatorname{tr}(x_1 p^2(x_1)) \mid p \in \mathbb{R}[t]\}.$$

*By Proposition 1.5.10, $X_1 \in \mathcal{D}^{\mathcal{F},\tau}_{\mathcal{M}^{\text{cyc}}}$ implies $X_1 \succeq 0$ for any tracial pair $(\mathcal{F}, \tau)$. On the other hand, if $\varepsilon \in [0, 1)$ then $x_1 + \varepsilon \notin \mathcal{M}(\{1 - x_1^2\})$ and therefore $x_1 + \varepsilon \notin \mathcal{M}^{\text{cyc}}$ by Lemma 1.5.11.*

To mitigate the absence of a non-pure analog of Theorem 1.5.2, we require the following technical lemma.

**Lemma 1.5.12** *Let $\varepsilon > 0$ and $n = \lceil 1/\varepsilon \rceil$. If $g_2 = \frac{\varepsilon}{2}(t-1)^{2n}$ and $g_1 = g_2 + t$, then*

(a) *$g_1$ is positive on $\mathbb{R}$, and thus a sum of (two) squares in $\mathbb{R}[t]$;*

(b) *$\frac{\varepsilon}{2} - g_2$ is nonnegative on $[0, 1]$, and thus an element of $\mathcal{M}(\{t, 1-t\})$.*

Although the tracial version of the Helton-McCullough Positivstellensatz [126] fails, we have the following positivity certificate for non-pure trace polynomials.

**Corollary 1.5.13** *Let $\mathcal{M}^{\text{cyc}} \subseteq \operatorname{Sym}\mathbb{T}$ be an archimedean cyclic quadratic module and $a \in \operatorname{Sym}\mathbb{T}$. The following are equivalent:*

(i) *$a(\underline{X}) \succeq 0$ for all $\underline{X} \in \mathcal{D}^{vN}_{\mathcal{M}^{\text{cyc}}}$;*

(ii) *$a(\underline{X}) \succeq 0$ for all $\underline{X} \in \mathcal{D}^{\text{II}_1}_{\mathcal{M}^{\text{cyc}}}$;*

(iii) *for every $\varepsilon > 0$, there exist sums of (two) squares $g_1, g_2 \in \mathbb{R}[t]$ such that*

$$a = g_1(a) - g_2(a), \qquad \varepsilon - \operatorname{tr}(g_2(a)) \in \mathcal{M}^{\text{cyc}}; \tag{1.66}$$

(iv) *for every $\varepsilon > 0$, there exist sums of (two) squares $g_1, g_2 \in \mathbb{R}[t]$ and $q \in \mathcal{M}^{\text{cyc}}$ such that*

$$\operatorname{tr}(ay) + \varepsilon = \operatorname{tr}(g_1(a)y + g_2(a)(1-y)) + q \tag{1.67}$$

*where $y$ is an auxiliary symmetric free variable. That is, $\operatorname{tr}(ay) + \varepsilon$ is in the cyclic quadratic module generated by $\mathcal{M}^{\text{cyc}}, y, 1-y$ (inside the free trace ring generated by $\underline{x}, y$).*

## Towards SDP hierarchies for trace optimization

Here, we apply Theorem 1.5.1 to optimization of pure trace objective functions subject to (pure) trace constraints and a norm boundedness condition. Doing so, we obtain later on a converging hierarchy of SDP relaxations. When flatness occurs in this hierarchy, one can extract a finite-dimensional minimizer. Finally, we apply Proposition 1.5.10 to handle the more general case of trace polynomials subject to trace constraints and a norm boundedness condition.

We define the set of *tracial words* (abbreviated as $\mathbb{T}$-words) by $\{\prod_i \operatorname{tr}(u_i)v \mid u_i, v \in \langle \underline{x} \rangle\}$, which is a subset of $\mathbb{T}$. The set of *pure trace words* (abbreviated as T-words) is the subset of $\mathbb{T}$-words belonging to T. For instance, $\operatorname{tr}(x_1)^2$ is a T-word and $\operatorname{tr}(x_1)x_1$ is a $\mathbb{T}$-word. For $u_i, v \in \langle \underline{x} \rangle$, we define the *tracial degree* of $\prod_i \operatorname{tr}(u_i)v$ as the sum of the degrees of the $u_i$ and the degree of $v$. The tracial degree of a trace polynomial $f \in \mathbb{T}$ is the length of the longest tracial word involved in $f$ up to cyclic equivalence. Let us denote by $\mathbf{W}_r^{\mathbb{T}}$ (resp. $\mathbf{W}_r^{\mathrm{T}}$) the vector of all $\mathbb{T}$-words (resp. T-words) w.r.t. to the lexicographic order. Finally, let $\mathbb{T}_r$ (resp. $\mathrm{T}_r$) denote the span of entries of $\mathbf{W}_r^{\mathbb{T}}$ (resp. $\mathbf{W}_r^{\mathrm{T}}$) in $\mathbb{T}$ (resp. T), and let $\sigma^{\mathbb{T}}(n, r)$ (resp. $\sigma^{\mathrm{T}}(n, r)$) the dimension of $\mathbb{T}_r$ (resp. $\mathrm{T}_r$), that is, the length of $\mathbf{W}_r^{\mathbb{T}}$ (resp. $\mathbf{W}_r^{\mathrm{T}}$).

We introduce the notion of trace Hankel and (pure) trace localizing matrices, which can be viewed as tracial analogs of the noncommutative localizing and Hankel matrices (see, e.g., [53, Lemma 1.44]). Given $g \in \mathbb{T}$, let us denote $r_g := \lceil \deg g/2 \rceil$. To $s$ and a linear functional $L : \mathrm{T}_{2r} \to \mathbb{R}$, one associates the following three matrices:

(a) the *tracial Hankel matrix* $\mathbf{M}^{\mathbb{T}}_r(L)$ is the symmetric matrix of size $\sigma^{\mathbb{T}}(n,r)$, indexed by $\mathbb{T}$-words $u,v \in \mathbb{T}_r$, with $(\mathbf{M}^{\mathbb{T}}_r(L))_{u,v} = L(\mathrm{tr}(u^\star v))$;

(b) if $g \in \mathrm{T}$, then the *pure trace localizing matrix* $\mathbf{M}^{\mathrm{T}}_{r-r_g}(g\,L)$ is the symmetric matrix of size $\sigma^{\mathrm{T}}(n,r-r_g)$, indexed by T-words $u,v \in \mathrm{T}_{r-r_g}$, with $(\mathbf{M}^{\mathrm{T}}_{r-r_g}(g\,L))_{u,v} = L(uvg)$*;*

(c) the *trace localizing matrix* $\mathbf{M}^{\mathbb{T}}_{r-r_g}(g\,L)$ is the symmetric matrix of size $\sigma^{\mathbb{T}}(n,r-r_g)$, indexed by $\mathbb{T}$-words $u,v \in \mathbb{T}_{r-r_g}$, with $(\mathbf{M}^{\mathbb{T}}_{r-r_g}(g\,L))_{u,v} = L(\mathrm{tr}(u^\star g v))$.

**Definition 1.5.14** *A matrix* $\mathbf{M}$ *indexed by* $\mathbb{T}$*-words of degree* $\leq r$ *satisfies the tracial Hankel condition if and only if*

$$\mathbf{M}_{u,v} = \mathbf{M}_{w,z} \textit{ whenever } \mathrm{tr}(u^\star v) = \mathrm{tr}(w^\star z)\,. \tag{1.68}$$

**Remark 1.5.1** *Linear functionals on* $\mathrm{T}_{2r}$ *and matrices from* $\mathbb{S}_{\sigma^{\mathbb{T}}(n,r)}$ *satisfying the tracial Hankel condition* (1.68) *are in bijective correspondence. To a linear functional* $L : \mathrm{T}_{2r} \to \mathbb{R}$*, one can assign the matrix* $\mathbf{M}^{\mathbb{T}}_r(L)$*, defined by* $(\mathbf{M}^{\mathbb{T}}_r(L))_{u,v} = L(\mathrm{tr}(u^\star v))$*, satisfying the tracial Hankel condition, and vice versa.*

One can relate the positivity of $L$ and the positive semidefiniteness of its tracial Hankel matrix $\mathbf{M}^{\mathbb{T}}_r(L)$. The proof of the following lemma is straightforward and analogous to its free counterpart [53, Lemma 1.44].

**Lemma 1.5.15** *Given a linear functional* $L : \mathrm{T}_{2r} \to \mathbb{R}$*, one has* $L(\mathrm{tr}(f^\star f)) \geq 0$ *for all* $f \in \mathbb{T}_r$*, if and only if,* $\mathbf{M}^{\mathbb{T}}_r(L) \succeq 0$*. Given* $g \in \mathrm{T}$*, one has* $L(a^2 g) \geq 0$ *for all* $a \in \mathrm{T}_{r-r_g}$*, if and only if,* $\mathbf{M}^{\mathrm{T}}_{r-r_g}(g\,L) \succeq 0$*. Given* $g \in \mathbb{T}$*, one has* $L(\mathrm{tr}(f^\star g f)) \geq 0$ *for all* $f \in \mathbb{T}_{r-r_g}$*, if and only if,* $\mathbf{M}^{\mathbb{T}}_{r-r_g}(g\,L) \succeq 0$*.*

## SDP hierarchy for pure trace polynomial optimization

For a finite $S \subseteq \mathrm{T}$, $N > 0$ and $r \in \mathbb{N}$ define

$$\mathcal{M}(S(N))_r := \left\{ \sum_{i=1}^{K} a_i^2 g_i \mid K \in \mathbb{N},\ a_i \in \mathrm{T},\ g_i \in S(N),\ \deg(a_i^2 g_i) \leq 2r \right\}. \tag{1.69}$$

Given $b \in \mathrm{T}$ and $p \in \mathbb{R}\langle \underline{x} \rangle$, note that $b^2\,\mathrm{tr}(pp^\star) = \mathrm{tr}((bp)(bp)^\star)$. Therefore, elements of $\mathcal{M}(S(N))_r$ correspond to sums of elements of the form

$$a_1^2 g\,, \quad a_2^2(N^k - \mathrm{tr}(x_j^{2k}))\,, \quad \mathrm{tr}(ff^\star)\,, \tag{1.70}$$

which are of degree at most $2r$, for $a_i \in \mathrm{T}, g \in S, 1 \leq j \leq n, k \in \mathbb{N}, f \in \mathbb{T}$.

Given a pure trace polynomial $a \in \mathrm{T}$, one can then use $\mathcal{M}(S(N))_r$ for $r = 1, 2, \ldots$ to design a hierarchy of semidefinite relaxations for minimizing $a \in \mathrm{T}$ over the von Neumann semialgebraic sets $\mathcal{D}^{\mathrm{vN}}_{S[N]}$ or $\mathcal{D}^{\mathrm{II}_1}_{S[N]}$.

Let us define $a_{\min}$ and $a^{\mathrm{II}_1}_{\min}$ as follows:

$$a_{\min} := \inf\{a(\underline{A}) \mid \underline{A} \in \mathcal{D}_{S[N]}\}\,, \tag{1.71}$$

$$a^{\mathrm{II}_1}_{\min} := \inf\{a(\underline{A}) \mid \underline{A} \in \mathcal{D}^{\mathrm{II}_1}_{S[N]}\} = \inf\{a(\underline{A}) \mid \underline{A} \in \mathcal{D}^{\mathrm{vN}}_{S[N]}\}\,. \tag{1.72}$$

Here the equality in (1.72) holds by Proposition 1.5.2. Since $\mathcal{D}_{S[N]}$ is a subset of $\mathcal{D}^{\mathrm{vN}}_{S[N]}$, one has $a^{\mathrm{II}_1}_{\min} \leq a_{\min}$. Let $r_{\min} := \max\{r_g : g \in \{a\} \cup S(N)\}$. Then, one can under-approximate $a^{\mathrm{II}_1}_{\min}$ via the following hierarchy of SDP programs, indexed by $r \geq r_{\min}$:

$$a^r_{\min} = \sup\{b \mid a - b \in \mathcal{M}(S(N))_r\}\,. \tag{1.73}$$

**Lemma 1.5.16** *The dual of* (1.73) *is the following SDP problem:*

$$
\begin{aligned}
\inf_{\substack{L:\mathrm{T}_{2r}\to\mathbb{R}\\ L\text{ linear}}} \quad & L(a) \\
\text{s.t.} \quad & (\mathbf{M}_r^{\mathbb{T}}(L))_{u,v} = (\mathbf{M}_r^{\mathbb{T}}(L))_{w,z}\,, \quad \textit{whenever } \operatorname{tr}(u^\star v) = \operatorname{tr}(w^\star z)\,, \\
& (\mathbf{M}_r^{\mathbb{T}}(L))_{1,1} = 1\,, \\
& \mathbf{M}_r^{\mathbb{T}}(L) \succeq 0\,, \\
& \mathbf{M}_{r-r_g}^{\mathrm{T}}(g\,L) \succeq 0\,, \quad \textit{for all } g \in S\,, \\
& \mathbf{M}_{r-k}^{\mathrm{T}}((N^k - \operatorname{tr}(x_j^{2k}))\,L) \succeq 0\,, \quad \textit{for all } j \in [n]\,, k \leq r\,.
\end{aligned}
\tag{1.74}
$$

Before proving that SDP (1.73) satisfies strong duality, we recall that an $\varepsilon$-neighborhood of 0 is the set $\mathcal{N}_\varepsilon$ defined for a given $\varepsilon > 0$ by:

$$
\mathcal{N}_\varepsilon := \bigcup_{k\in\mathbb{N}} \left\{ \underline{A} := (A_1, \ldots, A_n) \in \mathbb{S}_k^n : \varepsilon^2 - \sum_{i=1}^n A_i^2 \succeq 0 \right\} .
$$

**Lemma 1.5.17** *If $f \in \mathbb{T}$ vanishes on an $\varepsilon$-neighborhood of 0, then $f = 0$.*

**Theorem 1.5.3** *Let $S[N]$ be as in* (1.65) *and suppose that $\mathcal{D}_S$ contains an $\varepsilon$-neighborhood of 0. Then SDP* (1.73) *satisfies strong duality, i.e., there is no duality gap between SDP* (1.74) *and SDP* (1.73).

**Corollary 1.5.18** *The hierarchy of SDP programs* (1.73) *provides a sequence of lower bounds $(a^r_{\min})_{r\geq r_{\min}}$ monotonically converging to $a^{\mathrm{II}_1}_{\min}$.*

## Finite-dimensional GNS representations and minimizer extraction

In the commutative case, Curto and Fialkow provided sufficient conditions for linear functionals on the set of degree $2r$ polynomials to be represented by integration with respect to a nonnegative measure. The main sufficient condition to guarantee such a representation is flatness (see Definition 1.5.19) of the corresponding Hankel matrix. This notion was exploited in a noncommutative setting for the first time by McCullough [212] in his proof of the Helton-McCullough Sums of Squares theorem, cf. [212, Lemma 2.2] and relies on the GNS construction. In the pure noncommutative case [238] (see also [9, Chapter 21] and [53, Theorem 1.69]) provides a first noncommutative variant for the eigenvalue problem. See [54] for a similar construction for the trace problem.

The goal of this section is to derive an algorithm to extract minimizers of pure trace POP. The forthcoming statements can be seen as "pure trace" variants of the above mentioned results.

**Definition 1.5.19** *Suppose $L : \mathrm{T}_{2r+2\delta} \to \mathbb{R}$ is a tracial linear functional with restriction $\tilde{L} : \mathrm{T}_{2r} \to \mathbb{R}$. We associate to $L$ and $\tilde{L}$ the Hankel matrices $\mathbf{M}_{r+\delta}^{\mathbb{T}}(L)$ and $\mathbf{M}_r^{\mathbb{T}}(\tilde{L})$ respectively, and get the block form*

$$
\mathbf{M}_{r+\delta}^{\mathbb{T}}(L) = \begin{bmatrix} \mathbf{M}_r^{\mathbb{T}}(\tilde{L}) & B \\ B^T & C \end{bmatrix} .
$$

*We say that $L$ is $\delta$-*flat *or that $L$ is a $\delta$-*flat extension *of $\tilde{L}$, if $\mathbf{M}_{r+\delta}^{\mathbb{T}}(L)$ is flat over $\mathbf{M}_r^{\mathbb{T}}(\tilde{L})$, i.e., if* $\operatorname{rank} \mathbf{M}_{r+\delta}^{\mathbb{T}}(L) = \operatorname{rank} \mathbf{M}_r^{\mathbb{T}}(\tilde{L})$.

**Proposition 1.5.20** *Given $S \cup \{a\} \subseteq \mathrm{T}_{2r}$ let $S[N]$ be as in (1.65). Set $\delta := \max\{\lceil \deg s/2 \rceil : g \in S[N]\}$. Assume that $L$ is a $\delta$-flat extreme optimal solution of SDP (1.74). Then, one has*

$$a_{\min}^{d+\delta} = L(a) = a_{\min}^{\mathrm{II}_1}. \tag{1.75}$$

*Moreover, there are finitely many $n$-tuples $\underline{A}^{(j)}$ of symmetric matrices, and positive scalars $\lambda_j$ with $\sum_j \lambda_j = 1$, such that $a_{\min}^{\mathrm{II}_1} = a(\bigoplus_j \underline{A}^{(j)})$, where the tracial state is given by*

$$w\left(\bigoplus_j \underline{A}^{(j)}\right) \mapsto \sum_j \lambda_j \operatorname{tr}(w(\underline{A}^{(j)}))$$

*for $w \in \langle \underline{x} \rangle$.*

**Remark 1.5.2** *Proposition 1.5.20 guarantees that in a presence of a flat extension, there is an optimizer for $a_{\min}^{\mathrm{II}_1}$ arising from a finite-dimensional tracial pair $(\mathcal{F}, \tau)$; furthermore, the dimensions of $\underline{A}^{(j)}$ and the scalars $\lambda_j$ explicitly determine $\mathcal{F}$ and $\tau$, respectively. It is sensible to ask whether $a_{\min} = a_{\min}^{\mathrm{II}_1}$, that is, whether the optimum can be approximated arbitrarily well with a finite-dimensional factor, i.e., from $\mathcal{D}_{S[N]}$. If there exist sequences of positive rational numbers $(\lambda_j^{(m)})_m$ such that $\sum_j \lambda_j^{(m)} = 1$ for all $m \in \mathbb{N}$, $\lim_m \lambda_j^{(m)} = \lambda_j$ for all $j$, and $\bigoplus_j \underline{A}^{(j)} \in \mathcal{D}_{S[N]}^{vN}$ whenever the tracial state is given by*

$$w\left(\bigoplus_j \underline{A}^{(j)}\right) \mapsto \sum_j \lambda_j^{(m)} \operatorname{tr}(w(\underline{A}^{(j)})) \qquad \text{for } w \in \langle \underline{x} \rangle, \tag{1.76}$$

*then $a_{\min} = a_{\min}^{\mathrm{II}_1}$. Indeed, a finite-dimensional tracial pair with the rational-coefficient tracial state as in (1.76) embeds into a finite-dimensional factor. However, in general $a_{\min} \neq a_{\min}^{\mathrm{II}_1}$ even if $\mathcal{D}_S$ contains an $\varepsilon$-neighborhood of $0$ and $a_{\min}^{\mathrm{II}_1}$ admits a finite-dimensional optimizer; see the following example.*

**Example 1.5.2** *Fix $n = 1$, i.e., $\mathrm{T} = \mathbb{R}[\operatorname{tr}(x_1^i) \mid i \in \mathbb{N}]$. For $k \in \mathbb{N}$ let*

$$g_k := 1 + (\sqrt{2}+1)^2 - \left(\operatorname{tr}\left((x_1^2 - 2x_1)^2\right) + \left(\sqrt{2} - \operatorname{tr}(x_1)\right)^2\right) \operatorname{tr}(x_1^{2k}) \in \mathrm{T}.$$

*Let $X$ be a symmetric matrix. Then $X^2 \neq 2X$ or $\operatorname{tr}(X) \neq \sqrt{2}$. Furthermore, if $X$ is a contraction, then*

$$0 \preceq 2X - X^2 \preceq I, \quad |\sqrt{2} - \operatorname{tr}(X)| \leq \sqrt{2} + 1, \quad \operatorname{tr}(X^{2k}) \leq 1 \text{ for all } k \in \mathbb{N}.$$

*On the other hand, if $X$ is not a contraction, then there is $k \in \mathbb{N}$ such that*

$$\operatorname{tr}(X^{2k}) > \frac{1 + (\sqrt{2}+1)^2}{\operatorname{tr}\left((X^2 - 2X)^2\right) + \left(\sqrt{2} - \operatorname{tr}(X)\right)^2}.$$

*Let $S = \{g_k \mid k \in \mathbb{N}\}$ and $a = -\operatorname{tr}(x_1)$. Then $\mathcal{D}_S = \mathcal{D}_{1-x_1^2}$ by the above observations, and consequently $a_{\min} = -1$. On the other hand, consider the tracial pair $(\mathbb{R}^2, \tau)$ with $\tau(\xi_1, \xi_2) = \frac{1}{\sqrt{2}}\xi_1 + (1 - \frac{1}{\sqrt{2}})\xi_2$. Then $Y = (2, 0) \in \mathbb{R}^2$ satisfies $Y^2 = 2Y$ and $\tau(Y) = \sqrt{2}$, so $Y \in \mathcal{D}_S^{\mathrm{vN}}$. Therefore $a_{\min}^{\mathrm{II}_1} \leq a(Y) = -\sqrt{2}$. Eventually, let us prove that this finite-dimensional $Y$ is a minimizer for $a_{\min}^{\mathrm{II}_1}$. Take any operator $X$ in $\mathcal{D}_S$. If $X^2 \neq 2X$ or $\operatorname{tr}(X) \neq \sqrt{2}$, then $\operatorname{tr}(X) \leq 1$, as otherwise $\operatorname{tr}(X^{2k}) > 1$ for all $k$, which would contradict being in $\mathcal{D}_S$. Of course the alternative is that $X^2 = 2X$ and $\operatorname{tr}(X) = \sqrt{2}$. So this means that for all $X \in \mathcal{D}_S$, either $\operatorname{tr}(X) \leq 1$ or $\operatorname{tr}(X) = \sqrt{2}$, which proves that $a_{\min}^{\mathrm{II}_1} = -\sqrt{2}$.*

```
Require: an extreme $\delta$-flat linear $L : \mathrm{T}_{2r+2\delta} \to \mathbb{R}$ solution of (1.74).
Ensure: $(\underline{A}^{(1)}, \ldots, \underline{A}^{(k)})$ and $(\lambda_1, \ldots, \lambda_k)$.
Let us consider the set of $\mathbb{T}$-words $\{w_i\}$ of degree at most $\leqslant r$, such that $\mathscr{C}$, the matrix consisting
  of columns of $\mathbf{M}(L)$ indexed by the words $w_1, \ldots, w_r$, has full rank. Assume $w_1 = 1$.
Let $\mathbf{M}(\hat{L})$ be the principal submatrix of $\mathbf{M}(L)$ of columns and rows indexed by $w_1, \ldots, w_r$.
Let $C$ be the Cholesky factor of $\mathbf{M}(\hat{L})$, i.e., $C^T C = \mathbf{M}(\hat{L})$.
for $i \in [n]$ do
    Let $\mathscr{C}_i$ be the matrix consisting of columns of $\mathbf{M}(L)$ indexed by $x_i w_1, \ldots, x_i w_r$.
    Compute $\bar{A}_i$ as a solution of the system $\mathscr{C} \bar{A}_i = \mathscr{C}_i$.
    Let $A_i = C \bar{A}_i C^{-1}$.
end for
Compute $\mathbf{v} = Ce_1$.                                    ▷ $e_1 = (1, 0, \ldots, 0)$
Let $\mathcal{A}$ be the finite matrix algebra generated by $A_1, \ldots, A_n$. Compute an orthogonal matrix $Q$
  performing the simultaneous block-diagonalization of $A_1, \ldots, A_n$ by [219, Algorithm 4.1].   ▷
  $Q^T \mathcal{A} Q = \{\mathrm{Diag}(B^{(1)}, \ldots, B^{(k)}) \mid B^{(i)} \in \mathcal{A}_i\}$ where $\mathcal{A}_1, \ldots, \mathcal{A}_k$ are simple $\star$-algebras over $\mathbb{R}$
Compute $Q^T A_i Q = \mathrm{Diag}(A_i^{(1)}, \ldots, A_i^{(k)})$ for each $i \in [n]$, and $Q^T \mathbf{v} = ((\mathbf{v}^1)^T, \ldots, (\mathbf{v}^k)^T)^T$.
Compute $\lambda_j = \|\mathbf{v}^j\|$, and $\underline{A}^{(j)} = (A_1^{(j)}, \ldots, A_n^{(j)})$, for all $j \in [k]$.
```

Figure 1.10: `PureTraceGNS`.

The proof of Proposition 1.5.20, given in [R3], leads to the following procedure for minimizer extraction. The correctness of the procedure `PureTraceGNS` follows from the proof of Proposition 1.5.20.

**Corollary 1.5.21** *The procedure `PureTraceGNS` described in Figure 1.10 is sound and returns the n-tuples $\underline{A}^{(j)}$ and $\lambda_j$ from Proposition 1.5.20.*

**Remark 1.5.3** *Note that when flatness occurs, Proposition 1.5.20 guarantees convergence (actually stabilization) of our SDP hierarchy even if there is no ε-neighborhood of* 0 *in the feasible set. Moreover, while a flat extension is evasive from a numerical point of view, an "almost" flat extension, which is a much more viable output of an SDP solver, is likely sufficient [J5].*

## SDP hierarchy for trace polynomial optimization

Here we describe the reduction from the general trace setting to the pure trace setting.

Let $S \subset \operatorname{Sym} \mathbb{T}$ and $N > 0$. Denote

$$\widetilde{S} = \{\mathrm{tr}(fgf^\star) \mid g \in S,\ f \in \mathbb{T}\} \subset \mathrm{T}. \tag{1.77}$$

**Proposition 1.5.22** *Let $S \subset \operatorname{Sym} \mathbb{T}$, $N > 0$, and let $\widetilde{S}$ be as in (1.77). Then $\mathcal{D}^{\mathcal{F},\tau}_{\widetilde{S}[N]} = \mathcal{D}^{\mathcal{F},\tau}_{S[N]}$ for any tracial pair $(\mathcal{F}, \tau)$. Furthermore, the following are equivalent for $a \in \mathrm{T}$:*

(i) *$a(\underline{X}) \geq 0$ for all $\underline{X} \in \mathcal{D}^{vN}_{S[N]}$;*

(ii) *$a(\underline{X}) \geq 0$ for all $\underline{X} \in \mathcal{D}^{\mathrm{II}_1}_{S[N]}$;*

(iii) *$a + \varepsilon \in \mathcal{M}(\widetilde{S}(N))$ for all $\varepsilon > 0$.*

For all $r \in \mathbb{N}$, one has

$$\mathcal{M}(\widetilde{S}(N))_r = \left\{ \sum_{i=1}^{K} a_i^2 g_i \mid K \in \mathbb{N},\ a_i \in \mathrm{T},\ g_i \in \widetilde{S}(N),\ \deg(a_i^2 g_i) \le 2r \right\}.$$

Therefore, elements of $\mathcal{M}(\widetilde{S}(N))_r$ corresponds to sums of elements of the form

$$\mathrm{tr}(f_1\, g\, f_1^\star)\,, \quad a^2\big(N^k - \mathrm{tr}(x_j^{2k})\big)\,, \quad \mathrm{tr}(f_2 f_2^\star)\,, \tag{1.78}$$

which are of degree at most $2r$, for $f_i \in \mathbb{T}, a \in \mathrm{T}, g \in S, 1 \le j \le n, k \in \mathbb{N}$.

As before, given $a \in \mathrm{T}$, one can under-approximate $a_{\min}^{\mathrm{II}_1}$ via the following hierarchy of SDP programs, indexed by $r \ge r_{\min}$:

$$\widetilde{a}_{\min}^{r} = \sup\{b \mid a - b \in \mathcal{M}(\widetilde{S}(N))_r\}\,. \tag{1.79}$$

The dual of (1.79) is obtained by replacing the pure trace localizing matrix constraints in SDP (1.74) by trace localizing matrix constraints associated to each $g \in S$:

$$\begin{aligned} \inf_{\substack{L:\mathrm{T}_{2r}\to\mathbb{R} \\ L \text{ linear}}} \quad & L(a) \\ s.t. \quad & (\mathbf{M}_r^{\mathbb{T}}(L))_{u,v} = (\mathbf{M}_r^{\mathbb{T}}(L))_{w,z}\,, \quad \text{whenever } \mathrm{tr}(u^\star v) = \mathrm{tr}(w^\star z)\,, \\ & (\mathbf{M}_r^{\mathbb{T}}(L))_{1,1} = 1\,, \\ & \mathbf{M}_r^{\mathbb{T}}(L) \succeq 0\,, \\ & \mathbf{M}_{r-r_g}^{\mathbb{T}}(g\,L) \succeq 0\,, \quad \text{for all } g \in S\,, \\ & \mathbf{M}_{r-k}^{\mathrm{T}}((N^k - \mathrm{tr}(x_j^{2k}))\,L) \succeq 0\,, \quad \text{for all } j \in [n]\,, k \le d\,. \end{aligned} \tag{1.80}$$

As in Theorem 1.5.3, one can prove that if $\mathcal{D}_S$ contains an $\varepsilon$-neighborhood of 0, then there is no duality gap between SDP (1.80) and SDP (1.79). In addition, the hierarchy of SDP programs (1.79) provides a sequence of lower bounds monotonically converging to $a_{\min}^{\mathrm{II}_1}$.

Finally, the next result provides an alternative characterization of (not necessarily pure) trace polynomials positive on tracial semialgebraic sets (cf. Corollary 1.5.13).

**Proposition 1.5.23** *Let $S \subset \mathrm{Sym}\,\mathbb{T}$, $N > 0$, and let $\widetilde{S}$ be as in* (1.77)*. For $a \in \mathrm{Sym}\,\mathbb{T}$, the following are equivalent:*

(i) *$a(\underline{X}) \succeq 0$ for all $\underline{X} \in \mathcal{D}_{S[N]}^{vN}$;*

(ii) *$a(\underline{X}) \succeq 0$ for all $\underline{X} \in \mathcal{D}_{S[N]}^{\mathrm{II}_1}$;*

(iii) *for every $\varepsilon > 0$, there exist sums of (two) squares $g_1, g_2 \in \mathbb{R}[t]$ such that*

$$a = g_1(a) - g_2(a), \qquad \varepsilon - \mathrm{tr}(g_2(a)) \in \mathcal{M}(\widetilde{S}(N)); \tag{1.81}$$

(iv) *for every $\varepsilon > 0$, there exist sums of (two) squares $g_1, g_2 \in \mathbb{R}[t]$ and $q \in \mathcal{M}(\widetilde{S}(N))$ such that*

$$\mathrm{tr}(ay) + \varepsilon = \mathrm{tr}(g_1(a)y + g_2(a)(1-y)) + q \tag{1.82}$$

*where $y$ is an auxiliary symmetric free variable.*

Note that Proposition 1.5.23 allows one to certify that a given trace polynomial is positive semidefinite on a tracial semialgebraic set. Constructing a hierarchy of SDP programs converging to the minimal eigenvalue of trace polynomials is postponed for future work. As Example 1.5.1 indicates, it cannot be simply derived from our scheme for the pure trace polynomial objective function; namely, the norm of an operator cannot be uniformly estimated with traces in a dimension-free way.

## A toy example

Consider the optimization problem

$$\begin{aligned} &\inf && \tau(X_1X_2X_3)+\tau(X_1X_2)\tau(X_3) \\ &\text{s.t.} && X_j^2 = X_j^* = X_j \quad \text{for } j=1,2,3. \end{aligned} \tag{1.83}$$

over triples $(X_1, X_2, X_3)$ of operators in tracial pairs $(\mathcal{F}, \tau)$. Note that if $\mathcal{F}$ is a commutative von Neumann algebra with a tracial state $\tau$ and $X_1, X_2, X_3 \in \mathcal{F}$ are projections, then $\tau(X_1X_2X_3), \tau(X_1X_2), \tau(X_3) \geq 0$. Hence if (1.83) were restricted only to commutative von Neumann algebras, the solution would be 0. On the other hand, the projections

$$X_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad X_2 = \begin{pmatrix} \frac{1}{16} & \frac{\sqrt{15}}{16} \\ \frac{\sqrt{15}}{16} & \frac{15}{16} \end{pmatrix}, \quad X_3 = \begin{pmatrix} \frac{3}{8} & -\frac{\sqrt{15}}{8} \\ -\frac{\sqrt{15}}{8} & \frac{5}{8} \end{pmatrix}$$

give

$$\operatorname{tr}(X_1X_2X_3) + \operatorname{tr}(X_1X_2)\operatorname{tr}(X_3) = -\frac{1}{32}.$$

Below we show that $-\frac{1}{32}$ is actually the solution of (1.83).

Let $n=3$, $a = \operatorname{tr}(x_1x_2x_3) + \operatorname{tr}(x_1x_2)\operatorname{tr}(x_3)$ and $S = \{x_j^2 - x_j, x_j - x_j^2 : j = 1,2,3\}$. The solution of (1.83) equals $\lim_{n\to\infty} \check{a}^r_{\min}$, where $\check{a}^r_{\min}$ is the solution of (1.74) for $r \geq 2$. In this particular example, the constraints can be used to vastly simplify (1.74). Namely, it suffices to consider only tracial words without consecutive repetitions of $x_j$; furthermore, the last two lines in (1.74) are then superfluous. To state this concretely, let us introduce some auxiliary notation.

A $\mathbb{T}$-word is *reduced* if no proper powers of $x_1, x_2, x_3$ appear in it. To each $\mathbb{T}$-word $w$ we can assign the reduced $\mathbb{T}$-word $\rho(w)$ by repeatedly replacing $x_j^2$ with $x_j$. Let $\mathbf{W}_r^\rho$ be the vector of all reduced $\mathbb{T}$-words of tracial degree at most $r$, and let $R_r$ be the span of entries of $\mathbf{W}_r^\rho$. Given a linear functional $L : R_{2r} \to \mathbb{R}$, the *reduced tracial Hankel matrix* $\mathbf{M}_r^\rho(L)$ is indexed by $\mathbf{W}_r^\rho$ and $(\mathbf{M}_r^\rho(L))_{u,v} = L(\operatorname{tr}(\rho(u^*v)))$. Then $\check{a}^r_{\min}$ is the solution of the SDP

$$\begin{aligned} &\inf_{\substack{L:R_{2r}\to\mathbb{R} \\ L \text{ linear}}} && L(a) \\ &\text{s.t.} && (\mathbf{M}_r^\rho(L))_{u,v} = (\mathbf{M}_r^\rho(L))_{w,z}\,, \quad \text{whenever } \operatorname{tr}(u^\star v) = \operatorname{tr}(w^\star z)\,, \\ &&& (\mathbf{M}_r^\rho(L))_{1,1} = 1\,, \\ &&& \mathbf{M}_r^\rho(L) \succeq 0 \end{aligned} \tag{1.84}$$

We start with $r=2$. The matrix $\mathbf{M}_2^\rho(L)$ is indexed by reduced tracial words

$$\begin{aligned} &1, x_1, x_2, x_3, \operatorname{tr}(x_1), \operatorname{tr}(x_2), \operatorname{tr}(x_3), \\ &x_1x_2, x_2x_1, x_1x_3, x_3x_1, x_2x_3, x_3x_2, \operatorname{tr}(x_1x_2), \operatorname{tr}(x_1x_3), \operatorname{tr}(x_2x_3), \\ &\operatorname{tr}(x_1)x_1, \operatorname{tr}(x_1)x_2, \operatorname{tr}(x_1)x_3, \operatorname{tr}(x_2)x_1, \operatorname{tr}(x_2)x_2, \operatorname{tr}(x_2)x_3, \operatorname{tr}(x_3)x_1, \operatorname{tr}(x_3)x_2, \operatorname{tr}(x_3)x_3, \\ &\operatorname{tr}(x_1)^2, \operatorname{tr}(x_2)^2, \operatorname{tr}(x_3)^2, \operatorname{tr}(x_1)\operatorname{tr}(x_2), \operatorname{tr}(x_1)\operatorname{tr}(x_3), \operatorname{tr}(x_2)\operatorname{tr}(x_3). \end{aligned}$$

The SDP (1.84) minimizes over $31 \times 31$ positive semidefinite matrices subject to 881 linear equations in their entries. By solving it we get $\check{a}^2_{\min} = -0.0467$.

In the next step we have $r=3$, and $\mathbf{M}_3^\rho(L)$ is a $108 \times 108$ matrix with 11270 linear relations. Now the solution of (1.84) is $\check{a}^3_{\min} = -0.0312$, which up to floating point precision agrees with $-\frac{1}{32}$. Since $\check{a}^3_{\min}$ is a lower bound for the solution of (1.83) and is attained by the $2 \times 2$ projections above, we conclude that $-\frac{1}{32}$ is the solution of (1.83). While the SDPs themselves were solved using `SeDuMi`, the sparse input matrices were construed using Mathematica.

CHAPTER 2

# Certified polynomial optimization

## Contents

Chapter 1 was mostly dedicated to the modeling of various problems with moment-SOS hierarchies. We already illustrated the ability of such hierarchies to solve concrete instances with numerical, thus "inexact", SDP solvers. We will now focus on certified or "exact" optimization. In general, certified algorithms provide a way to ensure the safety of several systems in engineering sciences, program analysis as well as cyber-physical critical components. Since these systems often involve nonlinear functions, such as polynomials, it is highly desirable to design certified polynomial optimization schemes and to be able to interpret the behaviors of numerical solvers implementing these schemes.

- In Section 2.1, we interpret some wrong results (due to numerical inaccuracies) already observed when solving SDP relaxations for polynomial optimization on a double precision floating point SDP solver. It turns out that this behavior can be explained and justified satisfactorily by a relatively simple paradigm. In such a situation, the SDP solver (and not the user) performs some "robust optimization" without being told to do so. A mathematical rationale behind this "autonomous" behavior is described.

- Then, we describe, analyze and compare both from the theoretical and practical points of view, several algorithms computing weighted sums of squares decomposition for univariate and multivariate polynomials with rational coefficients, respectively in Section 2.2 and Section 2.3. In the univariate case, the first algorithm, developed by Schweighofer [263], relies on real root isolation, quadratic approximations of positive polynomials and square-free decomposition but its complexity was not analyzed. We provide bit complexity estimates, both on runtime and output size of this algorithm. They are exponential in the degree of the input univariate polynomial and linear in the maximum bitsize of its complexity. This analysis is obtained using quantifier elimination and root isolation bounds. The second univariate algorithm, due to Chevillard, Harrison, Joldes and Lauter [61], relies on complex root isolation and square-free decomposition and has been introduced for certifying positiveness of polynomials in the context of computer arithmetics. Again, its complexity was not analyzed. We provide bit complexity estimates, both on runtime and output size of this algorithm, which are polynomial in the degree of the input polynomial and linear in the maximum bitsize of its complexity. This analysis is obtained using Vieta's formula and root isolation bounds. We report on our implementations of both algorithms. While the second algorithm is, as expected from the complexity result, more efficient on most of examples, we exhibit families of non-negative polynomials for which the first algorithm is better. Then, we provide a hybrid numeric-symbolic algorithm computing exact rational SOS decompositions for polynomials lying in the interior of the SOS cone. It computes an approximate SOS decomposition for a perturbation of the input polynomial with an arbitrary-precision SDP solver. An exact SOS decomposition is obtained thanks to the perturbation terms. We prove that bit complexity estimates on output size and runtime are both polynomial in the degree of the input polynomial and simply exponential in the number of variables. Next, we apply this algorithm to compute exact Reznick and Putinar's representations for positive definite forms and positive polynomials over basic compact semialgebraic sets, respectively. We also compare the implementation of our algorithms with existing methods in computer algebra including cylindrical algebraic decomposition and critical point method.

- In Section 2.4, we rely on recently developed alternative methods to obtain nonnegativity certificates in a potentially cheaper way for sparse input polynomials with rational coefficients. We start to provide two hybrid numeric-symbolic optimization algorithms, computing exact sum of nonnegative circuits (SONC) and sum of arithmetic-geometric-mean-exponentials (SAGE) decompositions. Moreover, we provide a hybrid numeric-symbolic decision algorithm for polynomials lying in the interior of the SAGE cone. Each framework, inspired by previous contributions of Parrilo and Peyrl [237], is a rounding-projection procedure. For a polynomial lying in the interior of the SAGE cone, we prove that the decision algorithm terminates within a number of arithmetic operations, which is polynomial in the degree and number of terms of the input, and singly exponential in the number of variables. We also provide experimental comparisons regarding the implementation of the two optimization algorithms.

These contributions are in collaboration with researchers working in polynomial optimization: J.-B. Lasserre, as well as experts at the intersection of real algebraic geometry and computer algebra: M. Schweighofer (Professor, University of Konstanz) M. Safey El Din (Professor, LIP6/Sorbonne Université), T. de Wolff (Assistant Professor, TU Braunschweig) and his former PhD student H. Seidler (TU Berlin).

## 2.1 Two-player games between polynomial optimizers and SDP solvers

Wrong results (due to numerical inaccuracies) in some output results from SDP solvers have been observed in quite different applications, and notably in recent applications of the moment-SOS hierarchy for solving POP, see e.g., [298, 297]. In fact this particular application has even become a source of illustrating examples for potential pathological behavior of SDP solvers [233]. An intuitive mathematical rationale for the wrong results has been already provided informally in [176] and [223], but does not yield a satisfactory picture for the whole process.

An immediate and irrefutable negative conclusion is that double precision floating point SDP solvers are not robust and cannot be trusted as they sometimes provide wrong results in these so-called "pathological" cases. The present section is an attempt to provide a different and more positive viewpoint around the interpretation of such inaccuracies in SDP solvers, at least when applying the moment-SOS hierarchy of semidefinite relaxations in polynomial optimization as described in [180].

We claim that in such a situation, in fact the floating point SDP solver, and not the user, is precisely doing some robust optimization, *without being told to do so*. It solves a "max − min" problem in a two-player zero-sum game where the solver is the leader who maximizes (over some ball of radius $\varepsilon > 0$) in the parameter space of the criterion, and the user is a "follower" who minimizes over the original decision variables. In traditional robust optimization, one solves the "min − max" problem where the user (now the leader) minimizes to find a “robust decision variable”, whereas the SDP solver (now the follower) maximizes in the same ball of the parameter space. In this convex relaxation case, both min − max and max − min problems give the same solution. So it is fair to say that *the solver* is doing what the optimizer should have done in robust optimization.

As an active (and even leader) player of this game, the floating point SDP solver can also play with its two parameters which are (a) the threshold level for eigenvalues to declare a matrix positive semidefinite, and (b) the tolerance level at which to declare a linear equality constraint to be satisfied. Indeed, the result of the “max − min" game strongly depends on the absolute value of both levels, as well as on their relative values.

Of course and so far, the rationale behind this viewpoint which provides a more positive view of inaccurate results from semidefinite solvers, is proper to the context of semidefinite relaxations for polynomial optimization. Indeed in such a context we can exploit a mathematical rationale to explain and support this view. An interesting issue is to validate this viewpoint to a larger class of SDP and perhaps the canonical form of SDP:

$$\min_{\mathbf{G}} \{ \langle \mathbf{F}_0, \mathbf{G} \rangle : \langle \mathbf{G}_\alpha, \mathbf{G} \rangle = c_\alpha ; \mathbf{G} \succeq 0 \},$$

in which case the SDP solver would solve the robust optimization problem

$$\max_{\tilde{\mathbf{c}} \in \mathbf{B}_\infty(\mathbf{c}, \varepsilon)} \min_{\mathbf{G}} \{ \langle \mathbf{F}_0, \mathbf{G} \rangle : \langle \mathbf{F}_\alpha, \mathbf{G} \rangle = \tilde{c}_\alpha ; \mathbf{G} \succeq 0 \},$$

where $\langle \cdot \rangle$ stands for the matrix trace. This point of view is briefly analyzed and discussed later.

## Two examples of surprising phenomenons

Let us consider the general POP

$$\mathbf{P}: \quad f_{\min} = \min_{\mathbf{x}} \{ f(\mathbf{x}) : \mathbf{x} \in \mathbf{X} \}, \tag{2.1}$$

where $f$ is a polynomial and $\mathbf{X}$ is a basic closed semialgebraic set as in (1.1).

One can approximate $f_{\min}$ with the hierarchy of SDP relaxations [180], for which efficient modern softwares are available. These numerical solvers all rely on interior-point methods, and are implemented either in double precision arithmetics, e.g., SeDuMi [279], SDPA [310], MOSEK [7], or with arbitrary precision arithmetics, e.g., SDPA-GMP [221]. When relying on such numerical frameworks, the input data considered by solvers might differ from the ones given by the user. Thus the input data, consisting of the cost vector and matrices, are subject to uncertainties. In [96] the authors study SDP whose input data depend on some unknown but bounded perturbation parameters. For the reader interested in robust optimization in general, we refer to [34].

In general, when applied for solving $\mathbf{P}$, the moment-SOS hierarchy [180] is quite efficient, modulo its scalability (indeed for large size problems one has to exploit sparsity often encountered in the description of $\mathbf{P}$). However, in some cases, some quite surprising phenomena have been observed and provided additional support to the pessimistic and irrefutable conclusion that: *Results returned by double precision floating point SDP solvers cannot be trusted as they are sometimes completely wrong.*

Let us briefly describe two such phenomena, already analyzed and commented in [298, 223].

**Case 1:** When $\mathbf{X} = \mathbb{R}^n$ (unconstrained optimization) then the moment-SOS hierarchy collapses to the single SDP $f^k_{\min} = \max_b \{ b : f - b \in \Sigma[\mathbf{x}]_k \}$ (with $2k$ being the degree of $f$). Equivalently, one solves the SDP:

$$f^k_{\min} = \max_{\mathbf{G} \succeq 0, b} \{ b : f_\alpha - b \, 1_{\alpha=0} = \langle \mathbf{G}, \mathbf{B}_\alpha \rangle, \quad \alpha \in \mathbb{N}^n_{2k} \} \tag{2.2}$$

for some appropriate real symmetric matrices $(\mathbf{B}_\alpha)_{\alpha\in\mathbb{N}^n_{2k}}$; see, e.g., [180].

Only two cases can happen: if $f - f_{\min} \in \Sigma[\mathbf{x}]_k$ then $f^k_{\min} = f_{\min}$ and $f^k_{\min} < f_{\min}$ otherwise (with possibly $f^k_{\min} = -\infty$). Solving $f^r_{\min} = \max_b \{ b : f - b \in \Sigma[\mathbf{x}]_r \}$ for $r > k$ is useless as it would yield $f^r_{\min} = f^k_{\min}$ because if $f - f_{\min}$ is SOS, then it has to be in $\Sigma[\mathbf{x}]_k \subset \Sigma[\mathbf{x}]_r$ anyway.

The Motzkin-like polynomial $\mathbf{x} \mapsto f(\mathbf{x}) = x^2y^2(x^2+y^2-1) + 1/27$ is nonnegative (with $k=3$ and $f_{\min} = 0$) and has 4 global minimizers, but the polynomial $\mathbf{x} \mapsto f(\mathbf{x}) - f_{\min} (= f)$ is *not* an SOS and $f^3_{\min} = -\infty$, which also implies $f^r_{\min} = -\infty$ for all $r$. However, as already observed in [131], by solving (2.2) with $r = 8$ and a double precision floating point SDP solver, we obtain $f^8_{\min} \approx -10^{-4}$. In addition, one may extract 4 global minimizers close the global minimizers of $f$ up to four digits of precision! The same occurs with $r > 8$ and the higher is $r$ the better is the result. So undoubtly the SDP solver is returning a wrong solution as $f - f^r_{\min}$ *cannot* be an SOS, no matter the value of $f^r_{\min}$.

In this case, a rationale for this behavior is that $\tilde{f} = f + \varepsilon(1 + x^{16} + y^{16})$ is an SOS for small $\varepsilon > 0$, provided that $\varepsilon$ is not too small (in [176] it is shown that every nonnegative polynomial can be approximated as closely as desired by a sequence of polynomials that are SOS). After inspection of the returned optimal solution, the equality constraints

$$f_\alpha - b\,1_{\alpha=0} = \langle \mathbf{G}, \mathbf{B}_\alpha \rangle, \quad \alpha \in \mathbb{N}^n_{2r}, \tag{2.3}$$

when solving $\mathbf{D}^r$ in (2.2), are not satisfied accurately and the result can be interpreted as if the SDP solver *has replaced* $f$ with the perturbated criterion $\tilde{f} = f + \boldsymbol{\varepsilon}$, with $\boldsymbol{\varepsilon}(\mathbf{x}) = \sum_\alpha \varepsilon_\alpha\, \mathbf{G}^\alpha \in \mathbb{R}[\mathbf{x}]_{2r}$, so that

$$\underbrace{f_\alpha + \varepsilon_\alpha}_{\tilde{f}_\alpha} - b\,1_{\alpha=0} = \langle \mathbf{G}, \mathbf{B}_\alpha \rangle, \quad \alpha \in \mathbb{N}^n_{2r},$$

and in fact it has done so. A similar "mathematical paradox" has also been investigated in a noncommutative context [223]. As previously mentioned in Section 1.5, noncommutative polynomials can also be analyzed thanks to a specific variant of the moment-SOS hierarchy (see [53] for a recent survey). As in the above commutative case, it is explained in [223] how numerical inaccuracies allow to obtain converging lower bounds for positive Weyl polynomials that do not admit SOS decompositions.

**Case 2:** Another surprising phenomenon occurred when minimizing a high-degree univariate polynomial $f$ with a global minimizer at $x = 100$ and a local minimizer at $x = 1$ with value $f(1) > f_{\min}$ but very close to $f_{\min} = f(100)$. The double precision floating point SDP solver returns a single minimizer $\tilde{x} \approx 1$ with value very close to $f_{\min}$, providing another irrefutable proof that the double precision floating point SDP solver has returned a wrong solution. It turns out that again the result can be interpreted as if the SDP solver *has replaced* $f$ with a perturbated criterion $\tilde{f}$, as in Case 1.

When solving (2.2) in Case 1, one has voluntarily embedded $f \in \mathbb{R}[\mathbf{x}]_6$ into $\mathbb{R}[\mathbf{x}]_{2r}$ (with $r > 3$) to obtain a perturbation $\tilde{f} \in \mathbb{R}[\mathbf{x}]_{2r}$ whose minimizers are close enough to those of $f$. Of course the precision is in accordance with the solver parameters involved in controlling the semidefiniteness of the Gram matrix $\mathbf{G}$ and the accuracy of the linear equations (2.3). Indeed, if one tunes these parameters to a much stronger threshold, then the solver returns a more accurate answer with a much higher precision.

In both contexts, we can interpret what the SDP solver does as perturbing the coefficients of the input polynomial data. One approach to get rid of numerical uncertainties consists of solving SDP problems in an exact way [134], while using symbolic computation algorithms. However, such exact algorithms only scale up to moderate size instances. For situations when one has to rely on more efficient, yet inexact numerical algorithms, there is a need to understand the behavior of the associated numerical solvers. In [298], the authors investigate strange behaviors of double-precision SDP solvers for semidefinite relaxations in polynomial optimization. They compute the optimal values of the SDP relaxations of a simple one-dimensional POP. The sequence of SDP values practically converges to the optimal value of the initial problem while they should converge to a strict lower bound of this value. One possible remedy, used in [298], is to rely on an arbitrary-precision SDP solver, such as SDPA-GMP [221] in order to make this paradoxal phenomenon disappear. Relying on such arbitrary-precision solvers comes together with a more expensive cost but paves a way towards exact certification of nonnegativity. In Section 2.3 (see also [C8]), we present a hybrid numeric-symbolic algorithm computing exact SOS certificates for a polynomial lying in the interior of the SOS cone. This algorithm uses SDP solvers to compute an approximate SOS decomposition after additional perturbation of the coefficients of the input polynomial. The idea is to benefit from the perturbation terms added by the *user* to compensate the numerical uncertainties added by the *solver*. The present section focuses on analyzing specifically how the solver modifies the input and perturbates the polynomials of the initial optimization problem.

## A “noise" model

Let $\mathbb{S}_t$ be the set of real symmetric matrices of size $t = \binom{n+r}{n}$. Given a finite sequence of matrices $(\mathbf{F}_\alpha)_{\alpha \in \mathbb{N}^n_{2r}} \subset \mathbb{S}_t$, a (primal) cost vector $\mathbf{c} = (c_\alpha)_{\alpha \in \mathbb{N}^n_{2r}}$, we recall the standard form of *primal* SDP solved by numerical solvers such as SDPA [310]:

$$\begin{aligned} \min_{\mathbf{y}} \quad & \sum_{\alpha \in \mathbb{N}^n_{2r}} c_\alpha \, y_\alpha \\ \text{s.t.} \quad & \sum_{0 \neq \alpha \in \mathbb{N}^n_{2r}} \mathbf{F}_\alpha \, y_\alpha \succeq \mathbf{F}_0 \,, \end{aligned} \tag{2.4}$$

whose *dual* is the following SDP optimization problem:

$$\begin{aligned} \max_{\mathbf{G}} \quad & \langle \mathbf{F}_0, \mathbf{G} \rangle \\ \text{s.t.} \quad & \langle \mathbf{F}_\alpha, \mathbf{G} \rangle = c_\alpha \,, \quad \alpha \in \mathbb{N}^n_{2r} \,, \quad \alpha \neq 0 \,, \\ & \mathbf{G} \succeq 0 \,, \quad \mathbf{G} \in \mathbb{S}_t \,. \end{aligned} \tag{2.5}$$

We are interested in the numerical analysis of the moment-SOS hierarchy [180] to solve

$$\mathbf{P} : \quad \min_{\mathbf{x} \in \mathbf{X}} f(\mathbf{x}) \,,$$

where $f \in \mathbb{R}[\mathbf{x}]_{2r}$ and $\mathbf{X}$ is a basic compact semialgebraic set as in (1.1). Given $\alpha, \beta \in \mathbb{N}^n$, let $1_{\alpha=\beta}$ stands for the function which returns 1 if $\alpha = \beta$ and 0 otherwise. Let $t_j := \binom{n+r-r_j}{r-r_j}$. At step $r$ of the hierarchy, one solves the following SDP primal program.

$$\mathbf{P}^r : \inf_{\mathbf{y}} \{ L_{\mathbf{y}}(f) : \, y_0 = 1; \quad \mathbf{M}_{r-r_j}(g_j \, \mathbf{y}) \succeq 0, \quad j = 0, \ldots, m \} \,, \tag{2.6}$$

whose dual is the SDP:

$$\sup_{\mathbf{G}_j, b} \left\{ b : \begin{array}{l} f_\alpha - b \, 1_{\alpha=0} = \sum_{j=0}^{m} \langle \mathbf{C}^j_\alpha, \mathbf{G}_j \rangle \,, \quad \alpha \in \mathbb{N}^n_{2r} \,, \\ \mathbf{G}_j \succeq 0 \,, \quad \mathbf{G}_j \in \mathbb{S}_{t_j} \,, \quad j = 0, \ldots, m \end{array} \right\} , \tag{2.7}$$

where we have written $\mathbf{M}_{r-r_j}(g_j \, \mathbf{y}) = \sum_{\alpha \in \mathbb{N}^n_{2r}} \mathbf{C}^j_\alpha \, y_\alpha$; the matrix $\mathbf{C}^j_\alpha$ has rows and columns indexed by $\mathbb{N}^n_{r-r_j}$ with $(\beta, \gamma)$ entry equal to $\sum_{\beta+\gamma+\delta=\alpha} g_{j,\delta}$. In particular for $m = 0$, one has $g_0 = 1$ and the matrix $\mathbf{B}_\alpha := \mathbf{C}^0_\alpha$ has $(\beta, \gamma)$ entry equal to $1_{\beta+\gamma=\alpha}$.

Then the dual SDP (2.7) can be rewritten as

$$\begin{aligned}\sup_b \{b : f - b \in \mathcal{M}(\mathbf{X})_r\} = \sup_{b,\sigma_j} \{ b : \quad & f - b = \sum_{j=0}^m \sigma_j g_j \,, \\ & \deg(\sigma_j g_j) \,\leq 2r \,, \quad \sigma_j \in \Sigma[\mathbf{x}] \, \}.\end{aligned} \tag{2.8}$$

In the sequel, we suppose that $\mathbf{X}$ involves the constraint $N - \|\mathbf{x}\|_2^2 \geq 0$, so that Assumption 1.1.1 holds and we have strong duality between (2.6) and (2.8) by [148].

In floating point computation, the numerical SDP solver treats all (ideally) equality constraints as the following inequality constraints

$$\sum_{j=0}^m \langle \mathbf{C}_\alpha^j, \mathbf{G}_j \rangle + b 1_{\alpha=0} - f_\alpha = 0 \,, \quad \alpha \in \mathbb{N}_{2r}^n \,, \tag{2.9}$$

of (2.7) with the following inequality constraints

$$\left| \sum_{j=0}^m \langle \mathbf{C}_\alpha^j, \mathbf{G}_j \rangle + b 1_{\alpha=0} - f_\alpha \right| \leq \varepsilon \,, \quad \alpha \in \mathbb{N}_{2r}^n \,, \tag{2.10}$$

for some a priori fixed tolerance $\varepsilon > 0$ (for instance $\varepsilon = 10^{-8}$). Similarly, we assume that for each $j = 0, \ldots, m$, the SDP constraint $\mathbf{G}_j \succeq 0$ of (2.7) is relaxed to $\mathbf{G}_j \succeq -\eta \, \mathrm{I}$ for some prescribed *individual semidefiniteness tolerance* $\eta > 0$. This latter relaxation of $\succeq 0$ to $\succeq -\eta \, \mathrm{I}$ is used here as an idealized situation for modeling purpose; in practice it seems to be more complicated, as explained later in the numerical section.

That is, all iterates $(\mathbf{G}_{j,k})_{k \in \mathbb{N}}$ of the implemented minimization algorithm satisfy (2.10) and $\mathbf{G}_{j,k} \succeq -\eta \, \mathrm{I}$ instead of the idealized (2.9) and $\mathbf{G}_{j,k} \succeq 0$.

Therefore we interpret the SDP solver behavior by considering the following "noise" model which is the $(\varepsilon, \eta)$-perturbed version of SDP (2.7):

$$\begin{aligned}\sup_{\mathbf{G}_j, b} \{ b : \quad & -\varepsilon \leq \sum_{j=0}^m \langle \mathbf{C}_\alpha^j, \mathbf{G}_j \rangle + b 1_{\alpha=0} - f_\alpha \leq \varepsilon \,, \quad \alpha \in \mathbb{N}_{2r}^n \,, \\ & \mathbf{G}_j \succeq -\eta \, \mathrm{I} \,, \quad \mathbf{G}_j \in \mathbb{S}_{t_j} \,, \quad j = 0, \ldots, m \} ,\end{aligned} \tag{2.11}$$

now assuming exact computations. For any real symmetric matrix $\mathbf{M}$, denote by $\|\mathbf{M}\|_*$ its *nuclear* norm and recall that if $\mathbf{M} \succeq 0$ then $\|\mathbf{M}\|_* = \langle \mathrm{I}, \mathbf{M} \rangle$.

**Proposition 2.1.1** *The dual of Problem (2.11) is the convex optimization problem*

$$\begin{aligned} \inf_{\mathbf{y}} \quad & \{ L_{\mathbf{y}}(f) + \eta \sum_{j=0}^m \|\mathbf{M}_{r-r_j}(g_j \, \mathbf{y})\|_* + \varepsilon \|\mathbf{y}\|_1 : \\ \text{s.t.} \quad & y_0 = 1; \quad \mathbf{M}_{r-r_j}(g_j \, \mathbf{y}) \succeq 0, \quad j = 0, \ldots, m \, \} \end{aligned} \tag{2.12}$$

*which is an SDP.*

**Remark 2.1.1** *Notice that the criterion of (2.12) consists of the original criterion $L_{\mathbf{y}}(f)$ perturbated with a sparsity-inducing norm $\varepsilon \, \|\mathbf{y}\|_1$ for the variable $\mathbf{y}$ and a low-rank-inducing norm $\eta \, \sum_j \|\mathbf{M}_{r-r_j}(g_j \, \mathbf{y})\|_*$ for the localizing matrices. Considering this low-rank-inducing term can be seen as the convexification of a more realistic penalization with a logarithmic barrier function used in interior-point methods for SDP, namely $-\eta \log \det \left( \sum_{j=0}^m \mathbf{M}_{r-r_j}(g_j \, \mathbf{y}) \right)$. One could also consider to replace each SDP constraint $\mathbf{M}_{r-r_j}(g_j \, \mathbf{y}) \succeq 0$ with $\mathbf{M}_{r-r_j}(g_j \, \mathbf{y}) \succeq \varepsilon_3 \, \mathrm{I}$, in the primal moment problem (2.6). This corresponds to add $-\varepsilon_3 \|\mathbf{G}\|_*$ in the related perturbation of the dual SOS problem (2.7). One can in turn interpret this term as a convexification of the more standard logarithmic barrier penalization term* $\log \det \mathbf{G}$. *Even though interior-point algorithms could practically perform such logarithmic barrier penalizations, we do not have a simple interpretation for the related noise model.*

We now distinguish among two particular cases.

## Priority to trace equalities

With $\varepsilon = 0$ and individual semidefiniteness-tolerance $\eta$, Problem (2.12) becomes

$$\begin{aligned} \inf_{\mathbf{y}} \quad & \{ L_{\mathbf{y}}(f) + \eta \sum_{j=0}^{m} \|\mathbf{M}_{r-r_j}(g_j\, \mathbf{y})\|_* \\ \text{s.t.} \quad & y_0 = 1; \quad \mathbf{M}_{r-r_j}(g_j\, \mathbf{y}) \succeq 0, \quad j = 0, \dots, m \}. \end{aligned} \tag{2.13}$$

Given $\eta > 0$, $j \in \mathbb{N}$, let us define:

$$\mathbf{B}_\infty^r(f, \mathbf{X}, \eta) := \{ f + \theta \sum_{j=0}^{m} g_j(\mathbf{x}) \sum_{\beta \in \mathbb{N}^n_{r-r_j}} \mathbf{x}^{2\beta} : |\theta| \leq \eta \}, \tag{2.14}$$

$$\mathbf{B}_\infty(f, \mathbf{X}, \eta) := \bigcup_{j \in \mathbb{N}} \mathbf{B}_\infty^r(f, \mathbf{X}, \eta) .$$

Recall that SDP (2.13) is the dual of SDP (2.11) with $\varepsilon = 0$, that is,

$$\begin{aligned} \sup_{\mathbf{G}_j, b} \{ b : \quad & f_\alpha - b 1_{\alpha = 0} = \sum_{j=0}^{m} \langle \mathbf{C}_\alpha^j, \mathbf{G}_j \rangle , \quad \alpha \in \mathbb{N}^n_{2r} , \\ & \mathbf{G}_j \succeq -\eta \ \mathrm{I} , \quad \mathbf{G}_j \in \mathbb{S}_{t_j} , \quad j = 0, \dots, m \}, \end{aligned} \tag{2.15}$$

Fix $r \in \mathbb{N}$ and consider the following robust POP

$$\mathbf{P}_\eta^{\max} : \quad \max_{\tilde{f} \in \mathbf{B}_\infty(f, \mathbf{X}, \eta)} \{ \min_{\mathbf{x} \in \mathbf{X}} \{ \tilde{f}(\mathbf{x}) \} \} . \tag{2.16}$$

If in (2.16), we restrict ourselves to $\mathbf{B}_\infty^r(f, \mathbf{X}, \eta)$ and we replace the inner minimization by its step-$r$ relaxation, we obtain

$$\mathbf{P}_\eta^{\max, r} : \quad \max_{\tilde{f} \in \mathbf{B}_\infty^r(f, \mathbf{X}, \eta)} \left\{ \inf_{\mathbf{y}} \{ L_{\mathbf{y}}(\tilde{f}) : y_0 = 1; \ \mathbf{M}_{r-r_j}(g_j\, \mathbf{y}) \succeq 0, \ j = 0, \dots, m \} \right\} .$$

Observe that Problem $\mathbf{P}_\eta^{\max, r}$ is a strenghtening of Problem $\mathbf{P}_\eta^{\max}$, that is, the optimal value of the former is smaller than the optimal value of the latter.

**Proposition 2.1.2** *Under Assumption 1.1.1, there is no duality gap between primal SDP (2.13) and dual SDP (2.15). In addition, Problem $\mathbf{P}_\eta^{\max, r}$ is equivalent to SDP (2.13). Therefore, solving primal SDP (2.13) (resp. dual SDP (2.15)) can be interpreted as solving* exactly*, i.e., with no semidefiniteness-tolerance, the step-r strenghtening $\mathbf{P}_\eta^{\max, r}$ associated with Problem $\mathbf{P}_\eta^{\max}$.*

In the unconstrained case, i.e., when $m = 0$, solving $\mathbf{P}_\eta^{\max, r}$ boils down to minimize the perturbed polynomial $f_{\eta, r}(\mathbf{x}) := f(\mathbf{x}) + \eta \sum_{|\beta| \leq r} \mathbf{x}^{2\beta}$, that is the sum of $f$ and all monomial squares of degree up to $2r$ with coefficient magnitude $\eta$. As a direct consequence from [176], the next result shows that for given nonnegative polynomial $f$ and perturbation $\eta > 0$, the polynomial $f_{\eta, r}$ is SOS for large enough $r$.

**Corollary 2.1.3** *Let assume that $f \in \mathbb{R}[\mathbf{x}]$ is nonnegative over $\mathbb{R}^n$ and let us fix $\eta > 0$. Then $f_{\eta, r} \in \Sigma[\mathbf{x}]$, for large enough $r$.*

## Priority to semidefiniteness inequalities

Problem (2.12) with $\eta = 0$ and individual trace equality perturbation $\varepsilon$ becomes

$$\begin{aligned} \inf_{\mathbf{y}} \quad & \{ L_{\mathbf{y}}(f) + \varepsilon \|\mathbf{y}\|_1 : \\ \text{s.t.} \quad & y_0 = 1; \quad \mathbf{M}_{r-r_j}(g_j\, \mathbf{y}) \succeq 0, \quad j = 0, \dots, m \}. \end{aligned} \tag{2.17}$$

Given $\varepsilon > 0$, $r \in \mathbb{N}$, let us define

$$\mathbf{B}^r_\infty(f,\varepsilon) := \{\, \tilde{f} \in \mathbb{R}[\mathbf{x}]_{2r} : \|f - \tilde{f}\|_\infty \leq \varepsilon \,\}\,, \quad \mathbf{B}_\infty(f,\varepsilon) := \bigcup_{r\in\mathbb{N}} \mathbf{B}^r_\infty(f,\varepsilon)\,. \tag{2.18}$$

Recall that (2.17) is the dual of (2.11) with $\eta = 0$, that is,

$$\begin{aligned} \sup_{\tilde{f},b} \quad & \{\, b : \tilde{f} - b \in \mathcal{M}(\mathbf{X})_r; \quad |f_\alpha - \tilde{f}_\alpha| \leq \varepsilon\,, \quad \alpha \in \mathbb{N}^n_{2r}\,, \\ & b \in \mathbb{R}\,, \quad \tilde{f} \in \mathbb{R}[\mathbf{x}]_{2r} \,\}. \end{aligned} \tag{2.19}$$

Fix $r \in \mathbb{N}$ and consider the following robust POP:

$$\mathbf{P}^{\max}_\varepsilon : \quad \max_{\tilde{f} \in \mathbf{B}_\infty(f,\varepsilon)} \{\, \min_{\mathbf{x}\in\mathbf{X}} \{\tilde{f}(\mathbf{x})\} \,\}. \tag{2.20}$$

If in (2.20), we restrict ourselves to $\mathbf{B}^r_\infty(f,\varepsilon)$ in the outer maximization problem and we replace the inner minimization by its step-$r$ relaxation, we obtain

$$\begin{aligned} \mathbf{P}^{\max,r}_\varepsilon \quad &: \quad \max_{\tilde{f} \in \mathbf{B}^r_\infty(f,\varepsilon)} \{\, \sup_b \{\, b : \tilde{f} - b \in \mathcal{M}(\mathbf{X})_r \} \,\} \\ &= \quad \max_{\tilde{f} \in \mathbf{B}^r_\infty(f,\varepsilon)} \{\, \inf_{\mathbf{y}} \{\, L_{\mathbf{y}}(\tilde{f}) : y_0 = 1;\ \mathbf{M}_j(g_j\,\mathbf{y}) \succeq 0,\ j = 0,\ldots,m \} \,\} \end{aligned} \tag{2.21}$$

Here, we rely again on Assumption 1.1.1 to ensure strong duality and obtain (2.21). Problem $\mathbf{P}^{\max,r}_\varepsilon$ is a strengthening of $\mathbf{P}^{\max}_\varepsilon$ and whose dual is exactly (2.17), that is:

**Proposition 2.1.4** *Under Assumption 1.1.1, solving* (2.17) *(equivalently* (2.19)*) can be interprated as solving* exactly*, i.e.,with no trace-equality tolerance, the step-r reinforcement $\mathbf{P}^{\max,r}_\varepsilon$ associated with $\mathbf{P}^{\max}_\varepsilon$.*

## A two-player game interpretation

If we now assume that one can perform computations exactly, we can interpret the whole process in $\mathbf{P}^{\max,r}_\eta$ (resp. $\mathbf{P}^{\max,r}_\varepsilon$) as a two-player zero-sum game in which:

- Player 1 (the solver) chooses a polynomial $\tilde{f} \in \mathbf{B}^r_\infty(f,\mathbf{X},\eta)$ (resp. $\tilde{f} \in \mathbf{B}^r_\infty(f,\varepsilon)$).
- Player 2 (the optimizer) then selects a minimizer $\mathbf{y}^{\text{opt}}(\tilde{f})$ in the inner minimization of (2.21), e.g., with an exact interior point method.

As a result, Player 1 (the leader) obtains an optimal polynomial $\tilde{f}^{\text{opt}} \in \mathbf{B}^r_\infty(f,\mathbf{X},\eta)$ (resp. $\tilde{f}^{\text{opt}} \in \mathbf{B}^r_\infty(f,\varepsilon)$) and Player 2 (the follower) obtains an associated minimizer $\mathbf{y}^{\text{opt}}(\tilde{f}^{\text{opt}})$.
The polynomial $\tilde{f}^{\text{opt}}$ is the *worst* polynomial in $\mathbf{B}^r_\infty(f,\mathbf{X},\eta)$ (resp. $\mathbf{B}^r_\infty(f,\varepsilon)$) for the step-$r$ semidefinite relaxation associated with the optimization problem $\min_{\mathbf{x}}\{\, \tilde{f}(\mathbf{x}) : \mathbf{x} \in \mathbf{X}\}$. This max $-$ min problem is then equivalent to the single min-problem (2.13) (resp. (2.17)) which is a convex relaxation and whose convex criterion is not linear as it contains the sum of $\ell_\infty$-norm terms $\sum_{j=0}^m \|\mathbf{M}_{r-r_j}(g_j\,\mathbf{y})\|_*$ (resp. the $\ell_1$-norm term $\|\mathbf{y}\|_1$). Notice that in this scenario the optimizer (Player 2) is *not* active; initially he wanted to solve the convex relaxation associated with $f$. It is Player 1 (the adversary uncertainty in the solver) who in fact *gives* the exact algorithm his own choice of the function $\tilde{f} \in \mathbf{B}^r_\infty(f,\mathbf{X},\eta)$ (resp. $\tilde{f} \in \mathbf{B}^r_\infty(f,\varepsilon)$). But in fact, as we are in the convex case, the following result (a generalization of Von Neumann's minimax theorem, namely the following Sion's minimax theorem [269]) implies that this max $-$ min game is also equivalent to the min $-$ max game:

**Theorem 2.1.5** *Let* $\mathbf{B}$ *be a compact convex subset of a linear topological space and* $\mathbf{Y}$ *be a convex subset of a linear topological space. If* $h$ *is a real-valued function on* $\mathbf{B} \times \mathbf{Y}$ *with* $h(\mathbf{b}, \cdot)$ *lower semi-continuous and quasi-convex on* $\mathbf{Y}$*, for all* $\mathbf{b} \in \mathbf{B}$ *and* $h(\cdot, \mathbf{y})$ *upper semi-continuous and quasi-concave on* $\mathbf{B}$*, for all* $\mathbf{y} \in \mathbf{Y}$*, then*

$$\max_{\mathbf{b}\in\mathbf{B}} \inf_{\mathbf{y}\in\mathbf{Y}} h(\mathbf{b}, \mathbf{y}) = \inf_{\mathbf{y}\in\mathbf{Y}} \max_{\mathbf{b}\in\mathbf{B}} h(\mathbf{b}, \mathbf{y}) .$$

Indeed, $\mathbf{P}_\eta^{\max,r}$ is equivalent to

$$\inf_{\mathbf{y}} \max_{\tilde{f}\in\mathbf{B}_\infty^r(f,\mathbf{X},\eta)} \{ L_{\mathbf{y}}(\tilde{f}) : y_0 = 1; \mathbf{M}_j(g_j\, \mathbf{y}) \succeq 0, j = 0, \ldots, m \} ,$$

and $\mathbf{P}_\varepsilon^{\max,r}$ is equivalent to

$$\inf_{\mathbf{y}} \max_{\tilde{f}\in\mathbf{B}_\infty^r(f,\varepsilon)} \{ L_{\mathbf{y}}(\tilde{f}) : y_0 = 1; \mathbf{M}_j(g_j\, \mathbf{y}) \succeq 0, j = 0, \ldots, m \} ,$$

So now in this scenario (which assumes exact computations):

- Player 1 (the robust optimizer) chooses a feasible moment sequence $\mathbf{y}$ with $\mathbf{y}_0 = 1$ and $\mathbf{M}_{r-r_j}(g_j\, \mathbf{y}) \succeq 0, j = 0, \ldots, m$.
- When priority is given to trace equalities, Player 2 (the solver) then selects $\tilde{f}(\mathbf{y}) = \arg\max\{L_{\mathbf{y}}(\tilde{f}) : \tilde{f} \in \mathbf{B}_\infty^r(f, \mathbf{X}, \eta)\}$ to obtain the value $L_{\mathbf{y}}(f) + \eta \sum_{j=0}^m \|\mathbf{M}_{r-r_j}(g_j\, \mathbf{y})\|_*$.
  When priority is given to semidefinitess inequalities, Player 2 selects $\tilde{f}(\mathbf{y}) = \arg\max\{L_{\mathbf{y}}(\tilde{f}) : \tilde{f} \in \mathbf{B}_\infty^r(f, \varepsilon)\}$ to obtain the value $L_{\mathbf{y}}(f) + \varepsilon\|\mathbf{y}\|_1$, that is $\tilde{f}(\mathbf{y})_\alpha = f_\alpha + \text{sign}(y_\alpha)\, \varepsilon$, $\alpha \in \mathbb{N}_{2r}^n$.

Here the optimizer (now Player 1) is "active" as *he* decides to compute a "robust" optimal relaxation $\mathbf{y}$ assuming uncertainty in the function $f$ in the criterion $L_{\mathbf{y}}(f)$.

Since both scenarii are equivalent it is fair to say that the SDP solver is indeed solving the robust convex relaxation that the optimizer whould have given to a solver with exact arithmetic (if he had wanted to solve robust relaxations)

## Relating to robust optimization

Suppose that there is no computation errror but we want to solve a robust version of the optimization problem $\min\{f(\mathbf{x}) : \mathbf{x} \in \mathbf{X}\}$ because there is some uncertainty in the coefficients of the *nominal* polynomial $f \in \mathbb{R}[\mathbf{x}]_d$. So assume that $f \in \mathbb{R}[\mathbf{x}]_d$ can be considered as potentially of degree at most $2r$ (after perturbation).

When priority is given to trace equalities, the robust optimization problem reads:

$$\mathbf{P}_\eta^{\min,r} : \quad \min_{\mathbf{x}\in\mathbf{X}} \{ \max_{\tilde{f}\in\mathbf{B}_\infty^r(f,\mathbf{X},\eta)} \{\tilde{f}(\mathbf{x})\} \}. \tag{2.22}$$

Straightforward calculation reduces (2.22) to:

$$\mathbf{P}_\eta^{\min,r} : \quad \min_{\mathbf{x}\in\mathbf{X}} [ f(\mathbf{x}) + \eta \sum_{\beta\in\mathbb{N}_{r-r_j}^n} \mathbf{x}^{2\beta} g_j(\mathbf{x}) ] . \tag{2.23}$$

which is a POP.

**Theorem 2.1.1** *Suppose that Assumption 1.1.1 holds. Assume that after solving SDP (2.13), one obtains* $\mathbf{y}^{\text{opt}}$ *such that* $\mathbf{M}_r(\mathbf{y}^{\text{opt}})$ *is a rank-one matrix. Then* $\mathbf{P}_\eta^{\min,r}$ *is equivalent to* $\mathbf{P}_\eta^{\max,r}$.

When priority is given to semidefiniteness inequalities, the robust optimization problem reads:

$$\mathbf{P}_{\varepsilon}^{\min,r}: \quad \min_{\mathbf{x}\in\mathbf{X}} \{ \max_{\tilde{f}\in\mathbf{B}_{\infty}^{r}(f,\varepsilon)} \{\tilde{f}(\mathbf{x})\} \}. \tag{2.24}$$

It is easy to see that (2.24) reduces to

$$\mathbf{P}_{\varepsilon}^{\min,r}: \quad \min_{\mathbf{x}\in\mathbf{X}} [\, f(\mathbf{x}) + \varepsilon \sum_{\alpha\in\mathbb{N}_{2r}^{n}} |\mathbf{x}^{\alpha}|\,] . \tag{2.25}$$

which is *not* a POP (but is still a semialgebraic optimization problem). As for Theorem 2.1.1, one proves the following result:

**Theorem 2.1.2** *Suppose that Assumption 1.1.1 holds. Assume that after solving SDP (2.17), one obtains $\mathbf{y}^{\text{opt}}$ such that $\mathbf{M}_r(\mathbf{y}^{\text{opt}})$ is a rank-one matrix. Then $\mathbf{P}_{\varepsilon}^{\min,r}$ is equivalent to $\mathbf{P}_{\varepsilon}^{\max,r}$.*

Notice an important conceptual difference between the two approaches. In the latter one, i.e., when considering $\mathbf{P}_{\eta}^{\min}$ (resp. $\mathbf{P}_{\varepsilon}^{\min}$), the user is active. Indeed the user decides to choose some optimal $\hat{f} \in \mathbf{B}_{\infty}^{r}(f,\mathbf{X},\eta)$ (resp. $\mathbf{B}_{\infty}^{r}(f,\varepsilon)$). In the former one, i.e., when considering $\mathbf{P}_{\eta}^{\max}$ (resp. $\mathbf{P}_{\varepsilon}^{\max}$), the user is passive, as indeed he imposes $f$ but the solver decides to choose some optimal $f_{\min} \in \mathbf{B}_{\infty}^{r}(f,\mathbf{X},\eta)$ (resp. $\mathbf{B}_{\infty}^{r}(f,\varepsilon)$).
If after solving SDP (2.13) (resp. SDP (2.17)), one obtains $\mathbf{y}^{\text{opt}}$ where $\mathbf{M}_j(\mathbf{y}^{\text{opt}})$ is rank-one (which is to be expected), one obtains the same solution: in other words, we can interpret what the solver does as performing robust polynomial optimization.

In the sequel, we show how this interpretation relates with a more general robust SDP framework, when priority is given to semidefinitess inequalities.

## Link with robust semidefinite programming

Let $\mathbf{c} = (c_j) \in \mathbb{R}^n$, $\mathbf{F}_j$ be a real symmetric matrix, $j = 0,1,\ldots,n$, and let $\mathbf{F}(\mathbf{y}) := \sum_{j=1}^{n} \mathbf{F}_j\, y_j - \mathbf{F}_0$. Consider the canonical SDP:

$$\mathbf{P}: \quad \inf_{\mathbf{y}} \{ \mathbf{c}^T\mathbf{y} : \mathbf{F}(\mathbf{y}) \succeq 0 \} \tag{2.26}$$

with dual

$$\mathbf{P}^*: \quad \sup_{\mathbf{G}\succeq 0} \{ \langle \mathbf{F}_0, \mathbf{G} \rangle : \langle \mathbf{F}_j, \mathbf{G} \rangle = c_j, \quad j \in [n] \}. \tag{2.27}$$

Given $\varepsilon > 0$ fixed, let $\mathbf{B}_{\infty}(\mathbf{c},\varepsilon) := \{ \tilde{\mathbf{c}} : \|\tilde{\mathbf{c}} - \mathbf{c}\|_{\infty} \leq \varepsilon \}$ and consider the max-min problem associated with $\mathbf{P}$:

$$\max_{\tilde{\mathbf{c}}\in\mathbf{B}_{\infty}(\mathbf{c},\varepsilon)} \inf_{\mathbf{y}} \{ \tilde{\mathbf{c}}^T\, \mathbf{y} : \mathbf{F}(\mathbf{y}) \succeq 0 \}, \tag{2.28}$$

whose dual is

$$\sup_{\mathbf{X}\succeq 0} \{ \langle \mathbf{F}_0, \mathbf{X} \rangle : \quad | \langle \mathbf{F}_j, \mathbf{X} \rangle - c_j | \leq \varepsilon, \quad j \in [n] \}. \tag{2.29}$$

As before, there is a simple two-player game interpretation of (2.28). Player 1 (the leader) searches for the “best" cost function $\tilde{\mathbf{c}} \in \mathbf{B}_{\infty}(\mathbf{c},\varepsilon)$ which is “robust" against the *worst* decision $\mathbf{y}$ made by Player 2 (the follower, the decision maker), once Player 1’s choice $\tilde{\mathbf{c}}$ is known.

**Proposition 2.1.6** *Assume that there exists $\hat{\mathbf{y}}$ such that $\mathbf{F}(\hat{\mathbf{y}}) \succ 0$. Then solving the max-min problem (2.28) is equivalent to solving :*

$$\inf_{\mathbf{y}} \{ \mathbf{c}^T\, \mathbf{y} + \varepsilon\, \|\mathbf{y}\|_1 : \mathbf{F}(\mathbf{y}) \succeq 0 \}. \tag{2.30}$$

So again, with an appropriate value of $\varepsilon$ related the the numerical precision of SDP solvers, (2.29) can be considered as a fair model of treating inaccuracies by relaxing the equality constraints of (2.27) up to some tolerance level $\varepsilon$. That is, instead of solving exactly (2.27) with nominal criterion $\mathbf{c}$, Player 1 (the SDP solver) is considering a related robust version where it solves (exactly) (2.27) but now with some optimal choice of a new cost vector $\tilde{\mathbf{c}} \in \mathbf{B}_\infty(\mathbf{c}, \varepsilon)$. But this is a robustness point of view from the solver (*not* from the decision maker) and the resulting robust solution is some optimal cost vector $\tilde{\mathbf{c}}^* \in \mathbf{B}_\infty(\mathbf{c}, \varepsilon)$.

In the particular case of SDP relaxations for polynomial optimization, we retrieve (2.17) as an instance of (2.30) and (2.19) as an instance of (2.29).

## Robust SDP

On the other hand, the objective function $\tilde{\mathbf{c}}^T\mathbf{y}$ is bilinear in $(\tilde{\mathbf{c}}, \mathbf{y})$, the set $\mathbf{B}^r_\infty(\mathbf{c}, \varepsilon)$ is convex and compact, and the set $\mathbf{Y} := \{\mathbf{y} : \mathbf{F}(\mathbf{y}) \succeq 0\}$ is convex. Hence by Theorem 2.1.5, (2.28) is equivalent to solving the min-max problem:

$$\inf_{\mathbf{y}} \quad \{ \max_{\tilde{\mathbf{c}} \in \mathbf{B}_\infty(\mathbf{c},\varepsilon)} \{ \tilde{\mathbf{c}}^T\mathbf{y} \} : \quad \mathbf{F}(\mathbf{y}) \succeq 0 \}, \tag{2.31}$$

which is a “robust" version of (2.26) from the point of view of the decision maker when there is uncertainty in the cost vector. That is, the cost vector $\tilde{\mathbf{c}}$ is not known exactly and belongs to the uncertainty set $\mathbf{B}_\infty(\mathbf{c}, \varepsilon)$. The decision maker has to make a robust decision $\mathbf{y}^*$ with is the best against all possible values of the cost function $\tilde{\mathbf{c}} \in \mathbf{B}_\infty(\mathbf{c}, \varepsilon)$. This well-known latter point of view is that of *robust optimization* in presence of uncertainty for the cost vector; see, e.g., [96].

So if the latter robustness point of view (of the decision maker) is well-known, what is perhaps less known (but not so surprising) is that it can be interpreted in terms of a robustness point of view from an inexact “solver" when treating equality constraints with inaccuracies in a problem with nominal criterion. Given problem (2.26) with nominal criterion $\mathbf{c}$, and without being asked to do so, the solver behaves *as if* it is solving *exactly* the robust version (2.31) (from the decision maker viewpoint), whereas the decision maker is willing to solve (2.26) exactly. In other words, Sion's minimax theorem validates the informal (and not surprising) statement that the treatment of inaccuracies by the SDP solver can be viewed as a robust treatment of uncertainties in the cost vector.

However, in the case of SDP relaxations for polynomial optimization, this behavior is indeed more surprising and even spectacular. Indeed, some unconstrained optimization instances such as minimizing Motzkin-like polynomials (i.e., when $f - f_{\min}$ is not SOS), cannot be theoretically handled by SDP relaxations (assuming that one relies on exact SDP solvers). Yet, double floating point SDP solvers solve them in a practical manner, provided that higher-order relaxations are allowed so that a polynomial of degree $d$ can be (and indeed is!) treated as a higher degree polynomial (but with zero coefficients for monomials of degree higher than $d$).

In general, similar phenomena can occur while relying on general floating point algorithms. We presume that they could also appear when handling POP with alternative convex programming relaxations relying on interior-point algorithms, for instance linear/geometric programming.

## Examples

All experimental results are obtained by computing the solutions of the primal-dual SDP relaxations (2.6)-(2.7) of Problem $\mathbf{P}$. These SDP relaxations are implemented in the `RealCertify` [C9] library, available within MAPLE, and interfaced with the SDP solvers SDPA [310] and SDPA-GMP [221].

For the two upcoming examples, we rely on the procedure described in [131] to extract the approximate global minimizer(s) of some given objective polynomial functions. We compare the results obtained with (1) the SDPA solver implemented in double floating point precision, which corresponds to $\varepsilon = 10^{-7}$ and (2) the arbitrary-precision SDPA-GMP solver, with $\varepsilon = 10^{-30}$. The value of our robust-noise model parameter $\varepsilon$ roughly matches with the one of the parameter `epsilonStar` of SDPA.

We also noticed that decreasing the value of the SDPA parameter `lambdaStar` seems to boil down to increasing the value of our robust-noise model parameter $\eta$. An expected justification is that `lambdaStar` is used to determine a starting point $\mathbf{X}^0$ for the interior-point method, i.e., such that $\mathbf{X}^0 = \texttt{lambdaStar} \times \mathrm{I}$ (the default value of `lambdaStar` is equal to $10^2$ in SDPA and is equal to $10^4$ in SDPA-GMP). A similar behavior occurs when decreasing the value of the parameter `betaBar`, which controls the search direction of the interior-point method when the matrix $\mathbf{X}$ is not positive semidefinite.

However, the correlation between the values of `lambdaStar` (resp. `betaBar`) and $\eta$ appears to be nontrivial. Thus, our robust-noise model would be theoretically valid if one could impose the value of a parameter $\eta$, ensuring that $\mathbf{X} \succeq -\eta\ \mathrm{I}$ when the interior-point method terminates. From the best of our knowledge, this feature happens to be unavailable in modern SDP solvers. For that reason, our experimental comparisons are performed by changing the value of `epsilonStar` in the parameter file of the SDP solver.

First, we consider the Motkzin polynomial $f = \frac{1}{27} + x_1^2 x_2^2 (x_1^2 + x_2^2 - 1)$. This polynomial is nonnegative but is not SOS. The minimum $f_{\min}$ of $f$ is 0 and $f$ has four global minimizers with coordinates $x_1 = \pm\frac{\sqrt{3}}{3}$ and $x_2 = \pm\frac{\sqrt{3}}{3}$. As noticed in [131, Section 4], one can retrieve these global minimizers by solving the primal-dual SDP relaxations (2.6)-(2.7) of Problem $\mathbf{P}$ at relaxation order $r = 8$:

(1) With $\varepsilon = 10^{-7}$, we obtain an approximate lower bound of $-1.81 \cdot 10^{-4} \leq f_{\min}$, as well as the four global minimizers of $f$ with the extraction procedure. The dual SDP (2.7) allows to retrieve the approximate SOS decomposition $f(\mathbf{x}) = \sigma(\mathbf{x}) + \Delta(\mathbf{x})$, where $\sigma$ is an SOS polynomial and the corresponding polynomial remainder $\Delta$ has coefficients of approximately equal magnitude, and which is less than $10^{-8}$.

(2) With $\varepsilon = 10^{-30}$, we obtain an approximate lower bound of $-1.83 \cdot 10^{1} \leq f_{\min}$ and the extraction procedure fails. The corresponding polynomial remainder has coefficients of magnitude less than $10^{-31}$.

We notice that the support of $\Delta$ contains only terms of even degrees, i.e., terms of the form $\mathbf{x}^{2\beta}$, with $|\beta| \leq 8$. Hence we consider a perturbation $\tilde{f}_\gamma$ of $f$ defined by $\tilde{f}_\gamma(\mathbf{x}) = f(\mathbf{x}) + \gamma \sum_{|\beta| \leq r} \mathbf{x}^{2\beta}$, with $\gamma = 10^{-8}$. By solving the SDP relaxation (with $r = 8$) associated to $\tilde{f}_\gamma$, with $\varepsilon = 10^{-30}$, we retrieve again the four global minimizers of $f$.

Then, we consider the following univariate optimization problem:

$$f_{\min} = \min_{x \in \mathbb{R}} f(x) \,,$$

with $f(x) = (x - 100)^2 \left((x-1)^2 + \frac{\gamma}{99^2}\right)$ and $\gamma \geq 0$.
Note that the minimum of $f$ is $f_{\min} = 0 = f(100)$ and $f(1) = \gamma$.

We first examine the case where $\gamma = 0$. In this case, $f$ has two global minimizers 1 and 100. At relaxation order $r$, with $2 \leq r \leq 5$, we retrieve the following results (rounded to four significant digits):

(1) With $\varepsilon = 10^{-7}$, we obtain $\hat{x}^{(1)} = 0.9999 \simeq 1$, corresponding to the smallest global minimizer of $f$.

(2) With $\varepsilon = 10^{-30}$, we obtain $\hat{x} = 50.5000 = \frac{1+100}{2}$, corresponding to the average of the two global minimizers of $f$.

We also used the `realroot` procedure, available within Maple, to compute the local minimizers of the following function on $[0, \infty)$:

$$\tilde{f}_{\varepsilon,r}(x) = f(x) + \varepsilon \sum_{|\alpha| \leq 2r} |x^{\alpha}| = f(x) + \varepsilon \sum_{|\alpha| \leq 2r} x^{\alpha} , \tag{2.32}$$

(1) With $\varepsilon = 10^{-7}$, we obtain $\tilde{x}^{(1)} = 0.9961 \simeq \hat{x}^{(1)}$.

(2) With $\varepsilon = 10^{-30}$, we obtain $\tilde{x}^{(1)} = 0.9961 \simeq \hat{x}^{(1)}$ and $\tilde{x}^{(2)} = 99.9960 \simeq 100$, the largest global minimizer of $f$. The corresponding values of $\tilde{f}_{\varepsilon,r}$ are 0.1496 and 0.1495, respectively.

These experiments confirm our explanations that the solver computes the solution of SDP relaxations associated to the perturbed function $\tilde{f}_{\varepsilon,r}$ from (2.32). With double floating point precision (1), this perturbed function has a single minimizer, retrieved by the extraction procedure. With higher precision (2), this perturbed function has two local minimizers, whose average is retrieved by the extraction procedure.

Next, we examine the case where $\gamma = 10^{-3}$. In this case, $f$ has a single global minimizer, equal to 100 and another local minimizer At relaxation order $r$, with $2 \leq r \leq 5$, we retrieve the following results (rounded to four significant digits):

(1) With $\varepsilon = 10^{-7}$, we obtain $\hat{x}^{(1)} = 0.9999 \simeq 1$, corresponding to the smallest global minimizer of $f$ when $\gamma = 0$.

(2) With $\varepsilon = 10^{-30}$, we obtain $\hat{x}^{(2)} = 99.1593 \simeq 100$, corresponding to the single global minimizer of $f$.

We also compute the local minimizers of $\tilde{f}_{\varepsilon,r}$ with `realroot`:

(1) With $\varepsilon = 10^{-7}$, we obtain $\tilde{x}^{(1)} = 1.0039 \simeq \hat{x}^{(1)}$.

(2) With $\varepsilon = 10^{-30}$, we obtain $\tilde{x}^{(1)} = 1.0039 \simeq \hat{x}^{(1)}$ and $\tilde{x}^{(2)} = 99.9961 \simeq 100$, the single global minimizer of $f$. The corresponding values of $\tilde{f}_{\varepsilon,r}$ are 0.1505 and 0.1495, respectively. This confirms that $\tilde{x}^{(2)}$ is the single global minimizer of $\tilde{f}_{\varepsilon,r}$, approximately extracted, as $\hat{x}^{(2)}$.

Here again, our robust-noise model, relying on the perturbed polynomial function $\tilde{f}_{\varepsilon,r}$, fits with the above experimental observations. This perturbed function has a single global minimizer, whose value depends on the parameter $\varepsilon$, and which can be approximately retrieved by the extraction procedure.

## 2.2 Exact SOS certificates: the univariate case

Despite the fact that "inexact" SDP solvers provide approximate nonnegativity certificates, we can derive several algorithms to obtain "exact" ones. From now on in this chapter, we focus on this other two-player game.

The outlined results from this section have been published in [J17]. We begin this section by recalling the following classical result for nonnegative real-valued univariate polynomials (see e.g., [243, Section 8.1]):

**Theorem 2.2.1** *Let $f \in \mathbb{R}[x]$ be a nonnegative univariate polynomial, i.e., $f(x) \geq 0$ for all $x \in \mathbb{R}$. Then $f$ can be written as the sum of two polynomial squares in $\mathbb{R}[x]$.*

Given a subfield $K$ of $\mathbb{R}$ and a nonnegative univariate polynomial $f \in K[x]$, we consider the problem of proving the existence of, and computing, weighted sum of squares decompositions of $f$ with coefficients also lying in $K$, i.e., $a_1, \ldots, a_l \in K^{\geq 0}$ and $g_1, \ldots, g_l \in K[x]$ such that $f = \sum_{i=1}^{l} a_i g_i^2$.

Beyond the theoretical interest of this question, finding certificates of nonnegative polynomials is mandatory in many application fields. Among them, one can mention the stability proofs of critical control systems often relying on Lyapunov functions ([248]), the certified evaluation of mathematical functions in the context of computer arithmetic (see for instance [61]), the formal verification of real inequalities ([J2]) within proof assistants such as COQ ([288]) or HOL-LIGHT ([120]); in these situations the univariate case is already an important one. In particular, formal proofs of polynomial nonnegativity can be handled with weighted sum of squares certificates. These certificates are obtained with tools available outside of the proof assistants and eventually verified inside. Because of the limited computing power available inside such proof assistants, it is crucial to devise algorithms that produce certificates, whose checking is computationally reasonably simple. In particular, we would like to ensure that such algorithms output weighted sum of squares certificates of moderate bitsize and ultimately with a computational complexity being polynomial with respect to the input.

**Related Works** Decomposing nonnegative univariate polynomials into weighted SOS has a long story; very early quantitative aspects like the number of needed squares have been studied. For the case $K = \mathbb{Q}$, Landau showed in [169] that for every nonnegative polynomial in $\mathbb{Q}[x]$, there exists a decomposition involving a weighted sum of (at most) eight polynomial squares in $\mathbb{Q}[x]$. In [240], Pourchet improves this result by showing the existence of a decomposition involving only a weighted sum of (at most) five squares. This is done using approximation and valuation theory; extracting an algorithm from these tools is not the subject of study of this section.

More recently, the use of SDP for computing weighted SOS certificates of nonnegativity for polynomials has become very popular since [180]. Given a polynomial $f$ of degree $d$, this method consists in finding a real symmetric matrix $\mathbf{G}$ with nonnegative eigenvalues (a positive semidefinite matrix) such that $f(x) = v(x)^T \mathbf{G} v(x)$, where $v$ is the vector of monomials of degree less than $d/2$. Hence, this leads to the problem of solving a so-called linear matrix inequality (LMI), and one can rely on SDP to find the coefficients of $\mathbf{G}$. This task can be delegated to an SDP solver (e.g., SEDUMI, SDPA, SDPT3). An important technical issue arises from the fact that such SDP solvers are most of the time implemented with floating-point double precision. More accurate solvers are available (e.g., SDPA-GMP [221]). However, these solvers always compute numerical approximations of the algebraic solution to the SDP under consideration. Hence, they are not sufficient to provide algebraic certificates of posivity with rational coefficients. Hence, a process is needed to replace the computed numerical approximations of a sum of squares certificate by an exact, weighted sum of squares certificate with all weights and coefficients rational. This issue was tackled in [236, 151]. The certification scheme described in [J12] allows one to obtain lower bounds of nonnegative polynomials over compact sets. However, despite their efficiency, there is no guarantee that these methods will output a rational solution to an LMI when it exists (and especially when it is far from the computed numerical solution).

A more systematic treatment of this problem has been brought by the symbolic computation community. LMI can be solved as a decision problem over the reals with polynomial constraints using the Cylindrical Algebraic Decomposition algorithm [67] or more efficient critical point methods (see e.g., [30] for complexity estimates, and see [141, 100] for practical algorithms). But using such general algorithms is overkill, and, dedicated algorithms have been designed for computing exact algebraic solutions to LMI [134, 136]. Computing rational solutions can also be considered, thanks to convexity properties [260]. In particular, the algorithm in [110] can be used to compute weighted sum of squares certificates with rational coefficients for a nonnegative univariate polynomial of degree $d$ with coefficients of bitsize bounded by $\tau$ using at most $\tau^{\mathcal{O}(1)} 2^{\mathcal{O}(d^3)}$ boolean operations (see [110, Theorem 1.1]). In [46], the authors derive positivity certificates of polynomials positive over $[-1, 1]$ in the Bernstein basis. This certificate allows one in turn to produce a Positivstellensatz identity of total bitsize bounded by $\mathcal{O}(d^4 \log d + d^4 \tau)$, thus polynomial in $d$ and $\tau$ (see [46, Theorem 8]). To the best of our knowledge, there is no available implementation of this method.

For the case where $K$ is an arbitrary subfield of $\mathbb{R}$, Schweighofer gives in [263] a new proof of the existence of a decomposition involving a sum of (at most) $d$ polynomial squares in $K[x]$. This existence proof comes together with a recursive algorithm to compute such decompositions. At each recursive step, the algorithm performs real root isolation and quadratic approximations of positive polynomials. Later on, a second algorithm is derived in [61, Section 5.2], where the authors show the existence of a decomposition involving a sum of (at most) $d + 3$ polynomial squares in $K[x]$. Note that this second algorithm was presented earlier in [144, Section 7] (albeit with less detail and without a pointer to the code). This algorithm is based on approximating complex roots of perturbed positive polynomials.

Neither of these latter algorithms were analyzed, despite the fact that they were implemented and used. An outcome of our work is a bit complexity analysis for both of them, showing that they have better complexities than the algorithm in [110], the second algorithm being polynomial in $d$ and $\tau$.

## Nichtnegativstellensätze with quadratic approximations

We start with the case of degree $d = 2$ polynomials.

**Lemma 2.2.2** *Let $K$ be an ordered field. Let $g = ax^2 + bx + c \in K[x]$ with $a, b, c \in K$ and $a \neq 0$. Then $g$ can be rewritten as $g = a\left(x + \frac{b}{2a}\right)^2 + \left(c - \frac{b^2}{4a}\right)$. Moreover, when $g$ is nonnegative over $K$, one has $a > 0$ and $c - \frac{b^2}{4a} \geq 0$.*

Given a field $K$ and $g \in K[x]$, one says that $g$ is a *square-free* polynomial when there is no prime element $p \in K[x]$ such that $p^2$ divides $g$. Now let $f \in K[x] \setminus \{0\}$. A decomposition of $f$ of the form $f = a g_1^1 g_2^2 \ldots g_d^d$ with $a \in K$ and normalized pairwise coprime square-free polynomials $g_1, g_2, \ldots, g_d$ is called a *square-free decomposition* of $f$ in $K[x]$.

**Lemma 2.2.3** *[217, § 6.3.1] & [160, Lemma 9.26] Let $K$ be a field of characteristic 0 and $L$ a field extension of $K$. The square-free decomposition in $L[x]$ of any polynomial $f \in K[x] \setminus \{0\}$ is the same as the square-free decomposition of $f$ in $K[x]$. Any polynomial $f \in K[x] \setminus \{0\}$ which is a square-free polynomial in $K[x]$ is also square-free in $L[x]$.*

Let $f \in K[x]$ be a square-free polynomial that is nonnegative over $\mathbb{R}$. Then $f$ is positive over $\mathbb{R}$; otherwise $f$ would have at least one real root, implying that $f$ would be neither a square-free polynomial in $\mathbb{R}[x]$ nor a square-free polynomial in $K[x]$, according to Lemma 2.2.3. We want to find a polynomial $g \in K[x]$ that fulfills the following conditions:

(i) $\deg g \leq 2$,

(ii) $g$ is nonnegative over $\mathbb{R}$,

(iii) $f - g$ is nonnegative over $\mathbb{R}$,

(iv) $f - g$ has a root $t \in K$.

Assume that Property (i) holds. Then the existence of a weighted sum of squares decomposition in $K[x]$ for $g$ is ensured from Property (ii). Property (iii) implies that $h = f - g$ has only nonnegative values over $\mathbb{R}$. The aim of Property (iv) is to ensure the existence of a root $t \in K$ of $h$, which is stronger than the existence of a real root. Note that the case where the degree of $h = f - g$ is less than the degree of $f$ occurs only when $\deg f = 2$. In this latter case, we can rely on Lemma 2.2.2 to prove the existence of a weighted sum of squares decomposition.

Now, we investigate the properties of a polynomial $g \in K[x]$ that fulfills conditions (i)-(iv). Using Property (i) and Taylor Decomposition, we obtain $g(x) = g(t) + g'(t)(x-t) + c(x-t)^2$. By Property (iv), one has $g(t) = f(t)$. In addition, Property (iii) yields $f(x) - g(x) \geq 0 = f(t) - g(t)$, for all $x \in K$, which implies that $(f-g)'(t) = 0$ and $g'(t) = f'(t)$. By Property (ii), the quadratic polynomial $g(x+t) = f(t) + f'(t)x + cx^2$ has at most one real root. This implies that the discriminant of $g(x+t)$, namely $f'(t)^2 - 4cf(t)$, cannot be positive; thus one has $c \geq \frac{f'(t)^2}{4f(t)}$ (since $f(t) > 0$).

Finally, given a polynomial $g$ satisfying (i)-(iii) and (iv), one necessarily has $g = f_{t,c}$ with $\frac{f'(t)^2}{4f(t)} \leq c \in K$, and $f_{t,c} = f(t) + f'(t)(x-t) + c(x-t)^2$.

In this case, one also has that the polynomial $g = f_{t,c'}$, with $c' = \frac{f'(t)^2}{4f(t)}$, fulfills (i)-(iii) and (iv). Indeed, (i) and (iv) trivially hold. Let us prove that (ii) holds: when $\deg f_{t,c'} = 0$, $g = f(t) \geq 0$, and when $\deg f_{t,c'} = 2$, $g$ has a single root $t - \frac{f'(t)}{2c'}$, and the minimum of $g$ is $g\left(t - \frac{f'(t)}{2c'}\right) = 0$. The inequalities $f_{t,c'} \leq f_{t,c} \leq f$ over $\mathbb{R}$ yield (iii).

Therefore, given $f \in K[x]$ with $f$ positive over $\mathbb{R}$, we are looking for $t \in K$ such that the inequality $f \geq f_t$ holds over $\mathbb{R}$, with

$$f_t := f(t) + f'(t)(x-t) + \frac{f'(t)^2}{4f(t)}(x-t)^2 \in K[x] \,.$$

The main problem is to ensure that $t$ lies in $K$. If we choose $t$ to be a global minimizer of $f$, then $f_t$ would be the constant polynomial $\min\{f(x) \mid x \in \mathbb{R}\}$. The idea is then to find $t$ in the neighborhood of a global minimizer of $f$. The following lemma shows that the inequality $f_t \leq f$ can always be satisfied for $t$ in some neighborhood of a local minimizer of $f$.

**Lemma 2.2.4** *Let $f \in \mathbb{R}[x]$ and assume that $f$ is positive over $\mathbb{R}$. Let $a$ be a local minimizer of $f$. For all $t \in \mathbb{R}$ with $f(t) \neq 0$, let us define the polynomial $f_t$:*

$$f_t := f(t) + f'(t)(x-t) + \frac{f'(t)^2}{4f(t)}(x-t)^2 \in \mathbb{R}[x] \,.$$

*Then there exists a neighborhood $U \subset \mathbb{R}$ of $a$ such that the inequality $f_t(x) \leq f(x)$ holds for all $(x,t) \in U \times U$.*

Lemma 2.2.4 states the existence of a neighborhood $U$ of a local minimizer of $f$ such that the inequality $f_t(x) \leq f(x)$ holds for all $(x,t) \in U \times U$. Now, we show that with such a neighborhood $U$ of the smallest global minimizer $a$ of $f$, there exists $\varepsilon > 0$ such that the inequality $f_t(x) \leq f(x)$ holds for all $t \in (a - \varepsilon, a)$, and for all $x \in \mathbb{R}$.

**Proposition 2.2.5** *Let $f \in \mathbb{R}[x]$ with $\deg f > 0$. Assume that $f$ is positive over $\mathbb{R}$. Let $a$ be the smallest global minimizer of $f$. Then there exists a positive $\varepsilon \in \mathbb{R}$ such that for all $t \in \mathbb{R}$ with $a - \varepsilon < t < a$, the quadratic polynomial $f_t$, defined by*

$$\begin{aligned} f_t &:= f(t) + f'(t)(x-t) + \frac{f'(t)^2}{4f(t)}(x-t)^2 \\ &= \frac{f'(t)^2}{4f(t)}\left[\frac{2f(t)}{f'(t)} + (x-t)\right]^2 \in \mathbb{R}[x] \,, \end{aligned} \tag{2.33}$$

```
**Require:** nonnegative polynomial $f \in K[x]$ of degree $d \geq 2$, with $K$ a subfield of $\mathbb{R}$.
**Ensure:** pair of lists of polynomials $(\texttt{h\_list}, \texttt{q\_list})$ with coefficients in $K$.
 1: h_list ← [ ], q_list ← [ ].
 2: while $\deg f > 2$ do
 3:     $(g, h) := \texttt{sqrfree}(f)$                                          ▷ $f = gh^2$
 4:     if $\deg h > 0$ then h_list ← h_list $\cup \{h\}$, q_list ← q_list $\cup \{0\}$, $f \leftarrow g$
 5:     else
 6:         $f_t := \texttt{parab}(f)$
 7:         $(g, h) := \texttt{sqrfree}(f - f_t)$
 8:         h_list ← h_list $\cup \{h\}$, q_list ← q_list $\cup \{f_t\}$, $f \leftarrow g$
 9:     end if
10: end while
11: h_list ← h_list $\cup \{0\}$, q_list ← q_list $\cup \{f\}$
12: return h_list, q_list
```

Figure 2.1: `univsos1`: algorithm to compute SOS decompositions of nonnegative univariate polynomials.

*satisfies $f_t \leq f$ over $\mathbb{R}$.*

**Proposition 2.2.6** *Let $K$ be a subfield of $\mathbb{R}$ and $f \in K[x]$ with $\deg f = d \geq 1$. Then $f$ is nonnegative on $\mathbb{R}$ if and only if $f$ is a weighted sum of $d$ polynomial squares in $K[x]$, i.e., there exist $a_1, \dots, a_d \in K^{\geq 0}$ and $g_1, \dots, g_d \in K[x]$ such that $f = \sum_{i=1}^{d} a_i g_i^2$. (In fact, for $d \geq 4$, $d - 1$ squares suffice.)*

## Algorithm `univsos1`

The smallest global minimizer $a$ of $f$ is a real root of $f' \in K[x]$. Therefore, by using root isolation techniques [29, Chap. 10], one can isolate all the real roots of $f'$ in non-overlapping intervals with endpoints in $K$. Such techniques rely on applying successive bisections, so that one can arbitrarily reduce the width of every interval and sort them w.r.t. their left endpoints. Eventually, we apply this procedure to find a sequence of elements in $K$ converging from below to the smallest global minimizer of $f$ in order to find a suitable $t$. We denote by $\texttt{parab}(f)$ the corresponding procedure performing root isolation and returning the polynomial $f_t := \frac{f'(t)^2}{4f(t)}(x - t)^2 + f'(t)(x - t) + f(t)$ such that $t \in K$ and $f \geq f_t$ over $\mathbb{R}$.

Algorithm `univsos1`, depicted in Figure 2.1, takes as input a polynomial $f \in K[x]$ of even degree $d \geq 2$. The steps performed by this algorithm correspond to what is described in the proof of Proposition 2.2.6 and rely on two auxiliary procedures. The first one is the procedure `parab` (see Step 6). The second one is denoted by `sqrfree` and performs square-free decomposition: for a given polynomial $f \in K[x]$, $\texttt{sqrfree}(f)$ returns two polynomials $g$ and $h$ in $K[x]$ such that $f = gh^2$ and $g$ is square-free. When $f$ is square-free, the procedure returns $g = f$ and $h = 1$ (in this case $\deg h = 0$). As in the proof of Proposition 2.2.6, this square-free decomposition procedure is performed either on the input polynomial $f$ (Step 3) or on the nonnegative polynomial $(f - f_t)$ (Step 7). The output of Algorithm `univsos1` is a pair of lists of polynomials in $K[x]$, allowing one to retrieve an SOS decomposition of $f$. By Proposition 2.2.6 the length of all output lists, denoted by $l$, is bounded by $d/2$. If we write $h_l, \dots, h_1$ for the polynomials belonging to `h_list`, and $q_l, \dots, q_1$ the positive definite quadratic polynomials belonging to `q_list`, one obtains the following Horner-like decomposition: $f = h_l^2(h_{l-1}^2(h_{l-2}^2(\dots) + q_{l-2}) + q_{l-1}) + q_l$. Since each positive definite quadratic polynomial $q_i$ is a weighted SOS polynomial, this yields a weighted SOS decomposition for $f$.

**Example 2.2.1** *Let us consider the polynomial $f := \frac{1}{16}x^6 + x^4 - \frac{1}{9}x^3 - \frac{11}{10}x^2 + \frac{2}{15}x + 2 \in \mathbb{Q}[x]$.*

*We describe the different steps performed by Algorithm* `univsos`*1:*

- *The polynomial $f$ is square-free, and the algorithm starts by providing the value $t = -1$ as an approximation of the smallest minimizer of $f$. With $f(t) = \frac{1397}{720}$ and $f'(t) = \frac{-19}{8}$, one obtains $f_{-1} = \frac{720}{1397}(-\frac{19}{16}x + \frac{271}{360})^2$.*

- *Next, after obtaining the square-free decomposition $f(x) - f_{-1} = (x+1)^2 g$, the same procedure is applied on $g$. One obtains the value $t = 1$ as an approximation of the smallest minimizer of $g$ and $g_1 = \frac{502920}{237293}(-\frac{1}{18}x + \frac{88411}{167640})^2$.*

- *Eventually, one obtains the square-free decomposition $g(x) - g_1 = (x-1)^2 h$ with $h = \frac{1}{16}(x - \frac{19108973}{17085096})$.*

*Overall, Algorithm* `univsos`*1 provides the lists* `h_list` $= [1, x+1, 1, x-1, 0]$ *and* `q_list` $= [\frac{720}{1397}(-\frac{19}{16}x + \frac{271}{360})^2, 0, \frac{502920}{237293}(-\frac{1}{18}x + \frac{88411}{167640})^2, 0, \frac{1}{16}(x - \frac{19108973}{17085096})]$, *yielding the following weighted SOS decomposition:*

$$\begin{aligned} f =& (x+1)^2 \left[ (x-1)^2 \left( \frac{1}{16} \left( x - \frac{19108973}{17085096} \right)^2 \right) + \frac{502920}{237293} \left( -\frac{1}{18}x + \frac{88411}{167640} \right)^2 \right] \\ &+ \frac{720}{1397} \left( -\frac{19}{16}x + \frac{271}{360} \right)^2 . \end{aligned}$$

In the sequel, we analyze the complexity of Algorithm `univsos1` in the particular case $K = \mathbb{Q}$. We provide bounds on the bitsize of related SOS decompositions as well as bounds on the arithmetic cost required for computation and verification.

## Bit complexity analysis

For complexity estimates, we use the bit complexity model. For an integer $b \in \mathbb{Z} \backslash \{0\}$, we denote by $\tau(b) := \lfloor \log_2(|b|) \rfloor + 1$ the bitsize of $b$, with the convention $\tau(0) := 1$. We write a given polynomial $f \in \mathbb{Z}[x]$ of degree $d \in \mathbb{N}$ as $f = \sum_{i=0}^d b_i x^i$, with $b_0, \ldots, b_d \in \mathbb{Z}$. In this case, we define $\|f\|_\infty := \max_{0 \le i \le d} |b_i|$ and, using a slight abuse of notation, we denote $\tau(\|f\|_\infty)$ by $\tau(f)$. Observe that when $f$ has degree $d$, the bitsize necessary to encode $f$ is bounded by $d\tau(f)$ (when storing the coefficients of $f$). The derivative of $f$ is $f' = \sum_{i=1}^d i b_i x^{i-1}$. For a rational number $q = \frac{b}{c}$, with $b \in \mathbb{Z}, c \in \mathbb{Z} \backslash \{0\}$ and $\gcd(b,c) = 1$, we denote $\max\{\tau(b), \tau(c)\}$ by $\tau(q)$. For two mappings $g, h : \mathbb{N}^l \to \mathbb{R}^{>0}$, the expression "$g(v) = \mathcal{O}(h(v))$" means that there exists an integer $b \in \mathbb{N}$ and $N \in \mathbb{N}$ such that when all coordinates of $v$ are greater than or equaled to $N$, $g(v) \le b h(v)$. The expression "$g(v) = \tilde{\mathcal{O}}(h(v))$" means that there exists an integer $c \in \mathbb{N}$ such that for all $v \in \mathbb{N}^l$, $g(v) = O(h(v)(\log(h(v))^c)$.

**Lemma 2.2.7** *Let $f \in \mathbb{Z}[x]$ be a positive polynomial over $\mathbb{R}$, with* $\deg f = d$ *and $\tau$ an upper bound on the bitsize of the coefficients of $f$. When applying Algorithm* `univsos1` *to $f$, the sub-procedure* `parab` *outputs a polynomial $f_t$ such that $\tau(t) = \mathcal{O}(d^2\tau)$.*

**Lemma 2.2.8** *Let $f \in \mathbb{Z}[x]$ be a positive polynomial over $\mathbb{R}$, with* $\deg f = d$ *and $\tau$ an upper bound on the bitsize of the coefficients of $f$. Let $t$ and $f_t$ be as in Lemma 2.2.7. Let us write $t = \frac{t_1}{t_2}$, with $t_1 \in \mathbb{Z}, t_2 \in \mathbb{Z} \backslash \{0\}$, $t_1$ and $t_2$ being coprime. Let $\hat{f}(x) := t_2^{2d} f(t) f(x)$ and $\hat{f}_t(x) := t_2^{2d} f(t) f_t(x)$. The polynomial $f_t$ has coefficients of bitsize bounded by $\mathcal{O}(d^3\tau)$. Moreover, there exists $g \in \mathbb{Z}[x]$ such that $\hat{f} - \hat{f}_t = (x-t)^2 g$ and $\tau(g) = \mathcal{O}(d^3\tau)$.*

**Theorem 2.2.1** *Let $f \in \mathbb{Z}[x]$ be a positive polynomial over $\mathbb{R}$, with $\deg f = d = 2k$ and $\tau$ an upper bound on the bitsize of the coefficients of $f$. Then the maximum bitsize of the coefficients involved in the SOS decomposition of $f$ obtained with Algorithm* `univsos1` *is bounded from above by $\mathcal{O}\left((k!)^3\tau\right) = \mathcal{O}\left(\left(\left(\frac{d}{2}\right)^{\frac{3d}{2}}\right)\tau\right)$.*

**Theorem 2.2.2** *Let $f \in \mathbb{Z}[x]$ be a positive polynomial over $\mathbb{R}$, with $\deg f = d = 2k$ and $\tau$ an upper bound on the bitsize of the coefficients of $f$. Then, on input $f$, Algorithm* `univsos1` *runs in*

$$\widetilde{\mathcal{O}}\left(k^3 \cdot (k!)^3\tau\right) = \widetilde{\mathcal{O}}\left(\left(\frac{d}{2}\right)^{\frac{3d}{2}}\tau\right)$$

*boolean operations.*

For a given polynomial $f$ of degree $2k$, one can check the correctness of the SOS decomposition obtained with Algorithm `univsos1` by evaluating this SOS polynomial at $2k+1$ distinct points and compare the results with the ones obtained while evaluating $f$ at the same points.

**Theorem 2.2.3** *Let $f \in \mathbb{Z}[x]$ be a positive polynomial over $\mathbb{R}$, with $\deg f = d = 2k$ and $\tau$ an upper bound on the bitsize of the coefficients of $f$. Then one can check the correctness of the SOS decomposition of $f$ obtained with Algorithm* `univsos1` *within*

$$\widetilde{\mathcal{O}}\left(k \cdot (k!)^3\tau\right) = \widetilde{\mathcal{O}}\left(\left(\left(\frac{d}{2}\right)^{\frac{3d}{2}}\right)\tau\right)$$

*boolean operations.*

**Remark 2.2.1** *Let $f_k = f \in \mathbb{Z}[x]$. Under the strong assumption that each polynomial $f_k, \ldots, f_1$ involved in Algorithm* `univsos1` *has at least one integer global minimizer, Algorithm* `univsos1` *has polynomial complexity. Indeed, in this case, $q_i = f_i(t_i)$, $\tau(t_i) = \mathcal{O}\left(\tau(f_i)\right)$ and $\tau(f_{i-1}) = \mathcal{O}\left(2(i-1) + \tau(f_i)\right)$, for all $i = 2, \ldots, k$. Hence, the maximal bitsize of the coefficients involved in the SOS decomposition of $f$ is bounded from above by $\mathcal{O}\left(k^2 + \tau\right)$, and this decomposition can be computed using an expected number of $\widetilde{\mathcal{O}}\left(k^4 + k^3\tau\right)$ boolean operations.*

## Nichtnegativstellensätze with perturbed polynomials

Here, we recall the algorithm given in [61, Section 5.2]. The description of this algorithm, denoted by `univsos2`, is given in Figure 2.2.

## Algorithm `univsos2`

Given a subfield $K$ of $\mathbb{R}$ and a nonnegative polynomial $f = \sum_{i=0}^{d} f_i\, x^i \in K[x]$ of degree $d = 2k$, one first obtains the square-free decomposition of $f$, yielding $f = p\, h^2$ with $p > 0$ on $\mathbb{R}$ (see

```
Require: nonnegative polynomial $f \in K[x]$ of degree $d \geq 2$, with $K$ a subfield of $\mathbb{R}$, $\varepsilon \in K$ such
    that $0 < \varepsilon < f_d$, precision $\delta \in \mathbb{N}$ for complex root isolation
Ensure: list c_list of numbers in $K$ and list s_list of polynomials in $K[x]$
 1: $(p, h) \leftarrow$ sqrfree$(f)$                                          ▷ $f = p\,h^2$
 2: $d' := \deg p, k := d'/2$
 3: $p_\varepsilon \leftarrow p - \varepsilon \sum_{i=0}^{k} x^{2i}$
 4: while has_real_roots$(p_\varepsilon)$ do
 5:     $\varepsilon \leftarrow \frac{\varepsilon}{2}, p_\varepsilon \leftarrow p - \varepsilon \sum_{i=0}^{k} x^{2i}$
 6: end while
 7: $\varepsilon \leftarrow \frac{\varepsilon}{2}$
 8: $(s_1, s_2) \leftarrow$ sum_two_squares$(p_\varepsilon, \delta)$
 9: $C \leftarrow f_d, u \leftarrow p_\varepsilon - C s_1^2 - C s_2^2, u_{-1} \leftarrow 0, u_{2k+1} \leftarrow 0$     ▷ $u = \sum_{i=0}^{2k-1} u_i x^i$
10: while $\varepsilon < \max_{0 \leq i \leq k} \{ \frac{|u_{2i+1}|}{4} - u_{2i} + |u_{2i-1}| \}$ do
11:     $\delta \leftarrow 2\delta$, $(s_1, s_2) \leftarrow$ sum_two_squares$(p_\varepsilon, \delta)$, $u \leftarrow p_\varepsilon - C s_1^2 - C s_2^2$
12: end while
13: c_list $\leftarrow [C, C]$, s_list $\leftarrow [h\,s_1, h\,s_2]$
14: for $i = 0$ to $k - 1$ do
15:     c_list $\leftarrow$ c_list $\cup \{|u_{2i+1}|\}$, s_list $\leftarrow$ s_list $\cup \{h\,(x^{i+1} + \frac{\operatorname{sgn}(u_{2i+1})}{2} x^i)\}$
16:     c_list $\leftarrow$ c_list $\cup \{\varepsilon - \frac{|u_{2i+1}|}{4} + u_{2i} - |u_{2i-1}|\}$, s_list $\leftarrow$ s_list $\cup \{h\,x^i\}$
17: end for
18: return c_list $\cup \{\varepsilon + u_d - |u_{d-1}|\}$, s_list $\cup \{h\,x^k\}$
```

Figure 2.2: univsos2: algorithm to compute SOS decompositions of nonnegative univariate polynomials.

Step 1 of Figure 2.2).Then the idea is to find a positive number $\varepsilon > 0$ in $K$ such that the perturbed polynomial $p_\varepsilon(x) := p(x) - \varepsilon \sum_{i=0}^{k} x^{2i}$ is also positive on $\mathbb{R}$ (see [61, Section 5.2.2] for more details). This number is computed thanks to the loop going from Step 4 to Step 6, and relies on the auxiliary procedure has_real_roots, which checks whether the polynomial $p_\varepsilon$ has real roots using root isolation techniques. As mentioned in [61, Section 5.2.2], the number $\varepsilon$ is divided by 2 again to allow a margin of safety (Step 7).

Note that one can always ensure that the leading coefficient $C := p_d$ of $p$ is the same as the leading coefficient $f_d$ of the input polynomial $f$.

We obtain an approximate weighted rational sum of two polynomial squares decomposition of the polynomial $p_\varepsilon$ with the auxiliary procedure sum_two_squares (Step 8), relying on an arbitrary precision complex root finder. Recalling Theorem 2.2.1, this implies that the polynomial $p$ can be approximated as closely as desired by a weighted sum of two polynomial squares in $\mathbb{Q}[x]$, that is $C s_1^2 + C s_2^2$.

Thus there exists a remainder polynomial $u := p_\varepsilon - C s_1^2 - C s_2^2$ with coefficients of arbitrarily small magnitude (as mentioned in [61, Section 5.2.3]). The magnitude of the coefficients converges to 0 as the precision $\delta$ of the complex root finder goes to infinity. The precision is increased thanks to the loop going from Step 10 to Step 12 until a condition between the coefficients of $u$ and $\varepsilon$ becomes true, ensuring that $\varepsilon \sum_{i=0}^{k} x^{2i} + u(x)$ also admits a weighted SOS decomposition. For more details, see [61, Section 5.2.4].

The reason why Algorithm univsos2 terminates is the following: at first, one can always find a sufficiently small perturbation $\varepsilon$ such that the perturbed polynomial $p_\varepsilon$ remains positive. Next, one can always find sufficiently precise approximations of the complex roots of $p_\varepsilon$ ensuring that the error between the initial polynomial $p$ and the approximate SOS decomposition is compensated, thanks to the perturbation term.

The outputs of Algorithm `univsos2` are a list of numbers in $K$ and a list of polynomials in $K[x]$, allowing one to retrieve a weighted SOS decomposition of $f$. The size $l$ of both lists is equal to $2k+3 = d'+3 \leq d+3$. If we write $c_l, \dots, c_1$ for the numbers belonging to `c_list` and $s_l, \dots, s_1$ for the polynomials belonging to `s_list`, one obtains the following SOS decomposition: $f = c_l s_l^2 + \dots + c_1 s_1^2$.

**Example 2.2.2** *Let us consider the same polynomial* $f := \frac{1}{16}x^6 + x^4 - \frac{1}{9}x^3 - \frac{11}{10}x^2 + \frac{2}{15}x + 2 \in \mathbb{Q}[x]$ *as in Example 2.2.1. We describe the different steps performed by Algorithm* `univsos2`*:*

- *The polynomial $f$ is square-free, so we obtain $p = f$ (Step 1). After performing the loop from Step 3 to Step 4, Algorithm* `univsos2` *provides the value $\varepsilon = \frac{1}{32}$ at Step 7 as well as the polynomial $p_\varepsilon := p - \frac{1}{32}(1 + x^2 + x^4 + x^6)$, which has no real root.*
- *Next, after increasing three times the precision in the loop going from Step 6 to Step 17, the result of the approximate root computation yields $s_1 = x^3 - \frac{69}{8}x$ and $s_2 = 7x^2 - \frac{1}{4}x - \frac{63}{8}$.*

*Applying Algorithm* `univsos2`*, we obtain the following two lists of size $6+3=9$:*

$$\mathtt{c\_list} = \left[\frac{1}{32}, \frac{1}{32}, \frac{913}{15360}, \frac{731}{92160}, \frac{7}{1152}, \frac{1}{32}, \frac{79}{7680}, \frac{1}{576}, 0\right],$$
$$\mathtt{s\_list} = \left[x^3 - \frac{69}{8}x, 7x^2 - \frac{1}{4}x - \frac{63}{8}, 1, x, x^2, x^3, x + \frac{1}{2}, x(x - \frac{1}{2}), x^2(x + \frac{1}{2})\right],$$

*yielding the following weighted SOS decomposition:*

$$f = \frac{1}{32}\left(x^3 - \frac{69}{8}x\right)^2 + \frac{1}{32}\left(7x^2 - \frac{1}{4}x - \frac{63}{8}\right)^2 + \frac{913}{15360} + \frac{731}{92160}x^2$$
$$+ \frac{7}{1152}x^4 + \frac{1}{32}x^6 + \frac{79}{7680}\left(x + \frac{1}{2}\right)^2 + \frac{1}{576}x^2\left(x - \frac{1}{2}\right)^2.$$

## Bit complexity analysis

First, we need the following auxiliary result:

**Lemma 2.2.9** *Let $p \in \mathbb{Z}[x]$ be a positive polynomial over $\mathbb{R}$, with $\deg p = d$ and $\tau$ an upper bound on the bitsize of the coefficients of $p$. Then, one has*

$$\inf_{x \in \mathbb{R}} p(x) > (d 2^\tau)^{-d+2} 2^{-d \log_2 d - d\tau}.$$

**Lemma 2.2.10** *Let $p \in \mathbb{Z}[x]$ be a positive polynomial over $\mathbb{R}$, with $\deg p = d = 2k$ and let $\tau$ be an upper bound on the bitsize of the coefficients of $p$. Then there exists a positive integer $N = \mathcal{O}(d \log_2 d + d\tau)$ such that for all $N' \geq N$ the following holds. For $\varepsilon(N') := \frac{1}{2^{N'}}$, the polynomial $p_{\varepsilon(N')} := p - \varepsilon(N') \sum_{i=0}^k x^{2i}$ is positive over $\mathbb{R}$.*

**Theorem 2.2.4** *Let $f \in \mathbb{Z}[x]$ be a positive polynomial over $\mathbb{R}$, with $\deg f = d$ and $\tau$ an upper bound on the bitsize of the coefficients of $f$. Then the maximal bitsize of the weights and coefficients involved in the weighted SOS decomposition of $f$ obtained with Algorithm* `univsos2` *is bounded from above by $\mathcal{O}(d^3 + d^2\tau)$.*

**Theorem 2.2.5** *Let $f \in \mathbb{Z}[x]$ be a positive polynomial over $\mathbb{R}$, with $\deg f = d = 2k$ and $\tau$ an upper bound on the bitsize of the coefficients of $f$. Then, on input $f$, Algorithm* `univsos2` *runs in $\widetilde{\mathcal{O}}(d^4 + d^3\tau)$ boolean operations.*

We state now the complexity result for checking the SOS certificates output by Algorithm `univsos2`. As for the output of Algorithm `univsos1`, this is done through evaluation of the output at $d + 1$ distinct values where $d$ is the degree of the output.

**Theorem 2.2.6** *Let $f \in \mathbb{Z}[x]$ be a positive polynomial over $\mathbb{R}$, with $\deg f = d = 2k$ and $\tau$ an upper bound on the bitsize of the coefficients of $f$. Then one can check the correctness of the weighted SOS decomposition of $f$ obtained with Algorithm* `univsos2` *using $\widetilde{\mathcal{O}}(d^4 + d^3\tau)$ bit operations.*

## Practical experiments

Now we present experimental results obtained by applying Algorithm `univsos1` and Algorithm `univsos2`. Both algorithms have been implemented in a tool, called `RealCertify` [C9], written in Maple. The interested reader can find more details about installation and benchmark execution on the dedicated webpage.[1] The two algorithms are integrated into the RAGlib Maple package[2]. SOS decomposition (resp. verification) times are provided after averaging over five (resp., one thousand) runs.

As mentioned in [61, Section 6], the SOS decomposition performed by Algorithm `univsos2` has been implemented using the PARI/GP software tool[3] and is freely available (see [61]). To ensure fair comparison, we have rewritten this algorithm in Maple. To compute approximate complex roots of univariate polynomials, we rely on the PARI/GP procedure `polroots` through an interface with our Maple library. We also tried to use the Maple procedure `fsolve`, but the `polroots` routine from Pari/GP yielded significantly better performance for the polynomials involved in our examples.

The nine polynomial benchmarks presented in Table 2.1 allow to approximate some given mathematical functions, considered in [61, Section 6]. Computation and verification of SOS certificates are a mandatory step required to validate the supremum norm of the difference between such functions and their respective approximation polynomials on given closed intervals. This boils down to certifying two inequalities of the form $\forall x \in [b, c], p(x) \geq 0$, with $p \in \mathbb{Q}[X], b, c \in \mathbb{Q}$ and $\deg p = d$. As explained in [61, Section 5.2.5], this latter problem can be addressed by computing a weighted SOS decomposition of the polynomial $q(y) := (1 + y^2)^d \, p\left(\frac{b+cy^2}{1+y^2}\right)$, with either Algorithm `univsos1` or Algorithm `univsos2`. For each benchmark, we indicate in Table 2.1 the degree $d$ and the bitsize $\tau$ of the input polynomial, the bitsize $\tau_1$ of the weighted SOS decomposition provided by Algorithm `univsos1` as well as the corresponding computation (resp. verification) time $t_1$ (resp. $t_1'$) in milliseconds. Similarly, we display $\tau_2, t_2, t_2'$ for Algorithm `univsos2`. The table results show that for all other eight benchmarks, Algorithm `univsos2` yields better certification and verification performance, together with more concise SOS certificates. This observation confirms

[1] https://gricad-gitlab.univ-grenoble-alpes.fr/magronv/RealCertify

[2] http://www-polsys.lip6.fr/~safey/RAGLib/

[3] http://pari.math.u-bordeaux.fr

what we could expect after comparing the theoretical complexity results.

Table 2.1: Comparison results of output size and performance between Algorithm `univsos1` and Algorithm `univsos2` for nonnegative polynomial benchmarks from [61].

| Id | $d$ | $\tau$ (bits) | `univsos1` | | | `univsos2` | | |
|---|---|---|---|---|---|---|---|---|
| | | | $\tau_1$ (bits) | $t_1$ (ms) | $t'_1$ (ms) | $\tau_2$ (bits) | $t_2$ (ms) | $t'_2$ (ms) |
| # 1 | 13 | 22 682 | 3 403 218 | 2 723 | 0.40 | 51 992 | 824 | 0.14 |
| # 3 | 32 | 269 958 | 11 613 480 | 13 109 | 1.18 | 580 335 | 2 640 | 0.68 |
| # 4 | 22 | 47 019 | 1 009 507 | 4 063 | 1.45 | 106 797 | 1 776 | 0.31 |
| # 5 | 34 | 117 307 | 8 205 372 | 102 207 | 20.1 | 265 330 | 5 204 | 0.60 |
| # 6 | 17 | 26 438 | 525 858 | 1 513 | 0.74 | 59 926 | 1 029 | 0.21 |
| # 7 | 43 | 67 399 | 62 680 827 | 217 424 | 48.1 | 152 277 | 11 190 | 0.87 |
| # 8 | 22 | 27 581 | 546 056 | 1 979 | 0.77 | 63 630 | 1 860 | 0.38 |
| # 9 | 20 | 30 414 | 992 076 | 964 | 0.44 | 68 664 | 1 605 | 0.25 |
| # 10 | 25 | 42 749 | 3 146 982 | 1 100 | 0.38 | 98 926 | 2 753 | 0.39 |

Table 2.2: Comparison results of output size and performance between Algorithm `univsos1` and Algorithm `univsos2` for nonnegative polynomial benchmarks from [296].

| Id | $d$ | $\tau$ (bits) | `univsos1` | | | `univsos2` | | |
|---|---|---|---|---|---|---|---|---|
| | | | $\tau_1$ (bits) | $t_1$ (ms) | $t'_1$ (ms) | $\tau_2$ (bits) | $t_2$ (ms) | $t'_2$ (ms) |
| # A | 40 | 290 265 | 579 515 | 1 184 | 2.27 | 294 745 | 7 553 | 1.14 |
| # B | | 290 369 | 579 720 | 1 008 | 2.25 | 294 803 | 7 543 | 0.99 |
| # C | | 282 964 | 539 693 | 428 | 1.01 | 589 939 | 9 080 | 6.21 |
| # D | | 289 630 | 552 702 | 500 | 1.14 | 596 604 | 8 902 | 0.62 |
| # E | | 279 304 | 19 389 110 | 17 024 | 1.26 | 604 918 | 20 161 | 0.69 |

The five benchmarks from Table 2.2 are related to problems arising in verification of digital filters against frequency specifications (see [296, Section III B)]). As for the problems from Table 2.1, computation and verification of SOS certificates are mandatory to show the nonnegativity of a polynomial, which allows one in turn to validate the bounds of a rational function. By contrast with the comparison results from Table 2.1, Algorithm `univsos1` is faster for all examples. In addition, Algorithm `univsos1` produces output certificates of smaller size, compared to Algorithm `univsos2`, on the two benchmarks # C and # D. For all three other benchmarks, Algorithm `univsos2` provides more concise certificates. The slower performance of Algorithm `univsos2` is due to the time spent to obtain accurate approximations of the polynomial roots.

The comparison results available in Table 2.3 are obtained for power sums of increasing degrees. For a given natural number $d = 2k$ with $10 \leq d \leq 500$, we consider the polynomial $P_d := 1 + x + \cdots + x^d$. The roots of this polynomial are the $(d+1)$-st roots of unity, thus yielding the following SOS decomposition with real coefficients: $P_d := \prod_{j=1}^{k}((x - \cos\theta_j)^2 + \sin^2\theta_j)$, with $\theta_j := \frac{2j\pi}{d+1}$, for each $j \in [k]$. By contrast with the benchmarks from Table 2.1, Table 2.3 shows that Algorithm `univsos1` produces output certificates of much smaller size compared to Algorithm `univsos2`, with a bitsize ratio lying between 6 and 38 for values of $d$ between 10 and 200. This is due to the fact that Algorithm `univsos1` outputs a value of $t$ equal to 0 at each step. The execution performance of Algorithm `univsos1` is also much better in this case. The lack of efficiency of Algorithm `univsos2` is due to the computational bottleneck occurring when obtaining an accurate approximation of the relatively close roots $\cos\theta_j \pm i \sin\theta_j$, $j \in [k]$. For $d \geq 300$, the execution of Algorithm `univsos2` did not succeed after two hours of computation, as indicated by the symbol $-$ in the corresponding line.

Table 2.3: Comparison results of output size and performance between Algorithm `univsos1` and Algorithm `univsos2` for nonnegative power sums of increasing degrees.

| $d$ | `univsos1` | | | `univsos2` | | |
|---|---|---|---|---|---|---|
| | $\tau_1$ (bits) | $t_1$ (ms) | $t_1'$ (ms) | $\tau_2$ (bits) | $t_2$ (ms) | $t_2'$ (ms) |
| 10 | 84 | 7 | 0.03 | 567 | 264 | 0.03 |
| 20 | 195 | 10 | 0.05 | 1 598 | 485 | 0.06 |
| 40 | 467 | 26 | 0.09 | 6 034 | 2 622 | 0.18 |
| 60 | 754 | 45 | 0.14 | 12 326 | 6 320 | 0.32 |
| 80 | 1 083 | 105 | 0.18 | 21 230 | 12 153 | 0.47 |
| 100 | 1 411 | 109 | 0.26 | 31 823 | 19 466 | 0.69 |
| 200 | 3 211 | 444 | 0.48 | 120 831 | 171 217 | 2.08 |
| 300 | 5 149 | 1 218 | 0.74 | $-$ | $-$ | $-$ |
| 400 | 7 203 | 2 402 | 0.95 | | | |
| 500 | 9 251 | 4 292 | 1.19 | | | |
| 1000 | 20 483 | 30 738 | 2.56 | | | |

Further experiments are summarized in Table 2.4 for modified Wilkinson polynomials $W_d$ of increasing degrees $d = 2k$ with $10 \leq d \leq 600$ and $W_d := 1 + \prod_{j=1}^{k}(x-j)^2$. The roots $j \in [k]$ of $W_d - 1$ are relatively close (i.e.,the difference between two consecutive roots is small by comparison with the size of the coefficients), which yields again significantly slower performance of Algorithm `univsos2`. As observed in the case of power sums, timeout behaviors occur for $d \geq 60$. In addition, the bitsize of the SOS decompositions returned by Algorithm `univsos1` are much smaller. This is a consequence of the fact that in this case, $a = 1$ is the smallest global minimizer of $W_d$. Hence the algorithm always terminates at the first iteration by returning the trivial quadratic approximation $f_t = f_a = 1$ together with the square-free decomposition of $W_d - f_t = \prod_{j=1}^{k}(x-j)^2$.

Table 2.4: Comparison results of output size and performance between Algorithm `univsos1` and Algorithm `univsos2` for modified Wilkinson polynomials of increasing degrees.

| $d$ | $\tau$ (bits) | `univsos1` | | | `univsos2` | | |
|---|---|---|---|---|---|---|---|
| | | $\tau_1$ (bits) | $t_1$ (ms) | $t_1'$ (ms) | $\tau_2$ (bits) | $t_2$ (ms) | $t_2'$ (ms) |
| 10 | 140 | 47 | 17 | 0.01 | 2 373 | 751 | 0.03 |
| 20 | 737 | 198 | 31 | 0.01 | 12 652 | 3 569 | 0.08 |
| 40 | 3 692 | 939 | 35 | 0.01 | 65 404 | 47 022 | 0.17 |
| 60 | 9 313 | 2 344 | 101 | 0.01 | $-$ | $-$ | $-$ |
| 80 | 17 833 | 4 480 | 216 | 0.01 | | | |
| 100 | 29 443 | 7 384 | 441 | 0.01 | | | |
| 200 | 137 420 | 34 389 | 3 249 | 0.01 | | | |
| 300 | 335 245 | 83 859 | 11 440 | 0.01 | | | |
| 400 | 628 968 | 157 303 | 34 707 | 0.02 | | | |
| 500 | 1 022 771 | 255 767 | 73 522 | 0.02 | | | |
| 600 | 1 519 908 | 380 065 | 149 700 | 0.04 | | | |

Finally, we consider experimentation performed on modified Mignotte polynomials defined by

$M_{d,m} := x^d + 2(101x-1)^m$ and $N_d := (x^d + 2(101X-1)^2)(x^d + 2((101 + \frac{1}{101})x - 1)^2)$, for even integers $d$ and $m \geq 2$. The corresponding results are displayed in Table 2.5 for $M_{d,m}$ with $m = 2$ and $10 \leq d \leq 10000$, $m = d-2$ and $10 \leq d \leq 100$ as well as for $N_d$ with $10 \leq d \leq 100$. Note that similar benchmarks are used in [278] to anayze the efficiency of (real) root isolation techniques for polynomials with close roots. As for modified Wilkinson polynomials, Algorithm `univsos2` can only handle instances of small size, due to the limited scalability of the `polroots` procedure. In this case, Algorithm `univsos1` computes the approximation $t = \frac{1}{100}$ of the unique global minimizer of $M_{d,2}$. Thus, Algorithm `univsos1` always outputs weighted SOS decompositions of polynomials $M_{d,2}$ within a single iteration by first computing the quadratic polynomial $f_t = 2(101x-1)^2$ and the trivial square-free decomposition $W_d - f_t = x^d$. In the absence of such minimizers, Algorithm `univsos1` can only handle instances of polynomials $M_{d,d-2}$ and $N_d$ with $d \leq 100$.

Table 2.5: Comparison results of output size and performance between Algorithm `univsos1` and Algorithm `univsos2` for modified Mignotte polynomials of increasing degrees.

| Id | $d$ | $\tau$ (bits) | `univsos1` | | | `univsos2` | | |
|---|---|---|---|---|---|---|---|---|
| | | | $\tau_1$ (bits) | $t_1$ (ms) | $t'_1$ (ms) | $\tau_2$ (bits) | $t_2$ (ms) | $t'_2$ (ms) |
| $M_{d,2}$ | 10 | 27 | 23 | 2 | 0.01 | 4 958 | 1 659 | 0.04 |
| | $10^2$ | | | 3 | | – | – | – |
| | $10^3$ | | | 85 | | | | |
| | $10^4$ | | | 3 041 | | | | |
| $M_{d,d-2}$ | 10 | 288 | 25 010 | 21 | 0.03 | 6 079 | 2 347 | 0.04 |
| | 20 | 1 364 | 182 544 | 138 | 0.04 | 26 186 | 10 922 | 0.06 |
| | 40 | 5 936 | 1 365 585 | 1 189 | 0.13 | – | – | – |
| | 60 | 13 746 | 4 502 551 | 4 966 | 0.33 | | | |
| | 100 | 39 065 | 20 384 472 | 38 716 | 1.66 | | | |
| $N_d$ | 10 | 212 | 25 567 | 27 | 0.04 | – | – | – |
| | 20 | | 189 336 | 87 | 0.05 | | | |
| | 40 | | 5 027 377 | 1 704 | 0.17 | | | |
| | 60 | | 16 551 235 | 8 075 | 0.84 | | | |
| | 100 | | 147 717 572 | 155 458 | 11.1 | | | |

## 2.3 Exact SOS certificates: the multivariate case

Here we extend the previous framework from Section 2.2 to the multivariate case. The outlined results from this section have been published in [C8, J16, C9]. Namely, with $\mathbf{x} = (x_1, \ldots, x_n)$, we consider the problem of deciding the nonnegativity of $f \in \mathbb{Q}[\mathbf{x}]$ either over $\mathbb{R}^n$ or over a compact semialgebraic set $\mathbf{X}$ as in (1.1), defined by some constraints $g_1 \geq 0, \ldots, g_m \geq 0$, with $g_j \in \mathbb{Q}[\mathbf{x}]$. Further, $d$ denotes the maximum of the total degrees of these polynomials.

### Classical approaches

The Cylindrical Algebraic Decomposition algorithm due to [67] and [307] allows one to solve it in time doubly exponential in $n$ (and polynomial in $d$). This has been significantly improved, through the so-called critical point method, starting from [105] which culminates with [28] to establish that this decision problem can be solved in time $((m+1)d)^{O(n)}$. These latter ones have been developed to obtain practically fast implementations which reflect the complexity gain (see, e.g., [24, 20, 258, 257, 22, 109, 21, 101, 102]). These algorithms are "root finding" ones: they are designed to compute at least one point in each connected component of the set defined by $f < 0$. This is done by solving polynomial systems defining critical points of some well-chosen polynomial maps restricted to $f = -\varepsilon$ for $\varepsilon$ small enough. Hence the complexity of these algorithms depends on the difficulty of solving these polynomial systems (which can be exponential in $n$ as the Bézout bound on the number of their solutions is). Moreover, when $f$ is nonnegative, they return an empty list without a *certificate* that can be checked *a posteriori*. This section focuses on the computation of such certificates under some favorable situations. Since not all positive polynomials are SOS, what to do when SOS certificates do not exist? Also, given inputs with rational coefficients, can we obtain certificates with rational coefficients?

For these questions, we inherit from previous contributions in the univariate case by [61, J17], stated in Section 2.2, as well as in the multivariate case by [237, 152]. Note that [152, 151] allow us to compute SOS with rational coefficients on some degenerate examples. Moreover, [151] allows to compute decompositions into SOS of rational fractions. Diophantine aspects are considered in [259, 111]. When an SOS decomposition exists with coefficients in a totally real Galois field, [140] and [246] provide bounds on the total number of squares. In the multivariate unconstrained case, Parillo and Peyrl designed a rounding-projection algorithm in [237] to compute a weighted rational SOS decompositon of a given polynomial $f$ in the interior of the SOS cone. The algorithm computes an approximate Gram matrix of $f$, and rounds it to a rational matrix. With sufficient precision digits, the algorithm performs an orthogonal projection to recover an exact Gram matrix of $f$. The SOS decomposition is then obtained with an exact $\mathbf{LDL}^T$ procedure. This approach was significantly extended in [152] to handle rational functions and in [108] to derive certificates of impossibility for Hilbert-Artin representations of a given degree. In a recent work by [170], the author derives an algorithm based on facial reduction techniques to obtain exact rational decompositions for some sub-classes of nonnegative polynomials lying in the border of the SOS cone. Among such degenerate sub-classes, he considers polynomials that can be written as SOS of polynomials with coefficients in an algebraic extension of $\mathbb{Q}$ of odd degree.

## Exact SOS representations

Let $\Sigma[\mathbf{x}]$ be the convex cone of SOS in $\mathbb{R}[\mathbf{x}]$ and $\mathring{\Sigma}[\mathbf{x}]$ be the interior of $\Sigma[\mathbf{x}]$. We will be interested in those polynomials which lie in $\mathbb{Z}[\mathbf{x}] \cap \Sigma[\mathbf{x}]$. For instance, the polynomial

$$f = 4x_1^4 + 4x_1^3x_2 - 7x_1^2x_2^2 - 2x_1x_2^3 + 10x_2^4 = (2x_1x_2 + x_2^2)^2 + (2x_1^2 + x_1x_2 - 3x_2^2)^2$$

lies in $\mathbb{Z}[\mathbf{x}] \cap \Sigma[\mathbf{x}]$.

The *Newton polytope* or *cage* $\mathcal{C}(f)$ is the convex hull of the vectors of exponents of monomials that occur in $f \in \mathbb{R}[\mathbf{x}]$. For the above example, $\mathcal{C}(f) = \{(4,0),(3,1),(2,2),(1,3),(0,4)\}$. For a given Newton polytope $P$, let $\Sigma_P[\mathbf{x}]$ be the convex cone of SOS whose Newton polytope is contained in $P$. Since the Newton polytope $P$ is often clear from the context, we suppress the index $P$.

With $f \in \mathbb{R}[\mathbf{x}]$ of degree $d = 2k$, we consider the following formulation, which is the same as SDP (2.2):

$$\max_{\mathbf{G} \succeq 0, b} \{ b : f_\gamma - b\, 1_{\gamma=0} = \langle \mathbf{G}, \mathbf{B}_\gamma \rangle, \quad \gamma \in \mathbb{N}^n_{2k} \} \tag{2.34}$$

where $\langle \mathbf{G}, \mathbf{B} \rangle$ stands for the trace of $\mathbf{GB}$. Recall that $\mathbf{B}_\gamma$ has rows (resp. columns) indexed by $\mathbb{N}^n_k$ with $(\alpha, \beta)$ entry equal to 1 if $\alpha + \beta = \gamma$ and 0 otherwise.

**Theorem 2.3.1** *[180, Theorem 3.2] Let $f \in \mathbb{R}[\mathbf{x}]$ of degree $d = 2k$ and global infimum $f_{\min} := \inf_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x})$. Assume that SDP (2.34) has a feasible solution $\mathbf{G} = \sum_{i=1}^{l} \lambda_i \mathbf{q}_i\, \mathbf{q}_i^T$, with the $\mathbf{q}_i$ being the eigenvectors of $\mathbf{G}$ corresponding to the nonnegative eigenvalues $\lambda_i$, for all $i \in [l]$. Then $f - f_{\min} = \sum_{i=1}^{l} \lambda_i q_i^2$.*

For the sake of efficiency, one reduces the size of the matrix $\mathbf{G}$ by indexing its rows and columns by half of $\mathcal{C}\,(f)$:

**Theorem 2.3.2** *[16, Theorem 1] Let $f \in \Sigma[\mathbf{x}]$ with $f = \sum_{i=1}^{l} s_i^2$, $P := \mathcal{C}\,(f)$ and $\mathscr{B} := P/2 \cap \mathbb{N}^n$. Then for all $i \in [l]$, $\mathcal{C}\,(s_i) \subseteq \mathscr{B}$.*

Given $f \in \mathbb{R}[\mathbf{x}]$, Theorem 2.3.1 states that one can theoretically certify that $f$ lies in $\Sigma[\mathbf{x}]$ by solving SDP (2.34). However, available SDP solvers are typically implemented in finite-precision and require the existence of a strictly feasible solution $\mathbf{G} \succ 0$ to converge. This is equivalent for $f$ to lie in $\mathring{\Sigma}[\mathbf{x}]$ as stated in [64, Proposition 5.5]:

**Theorem 2.3.1** *Let $f \in \mathbb{Z}[\mathbf{x}]$ with $P := \mathcal{C}\,(f)$, $\mathscr{B} := P/2 \cap \mathbb{N}^n$ and $v_k$ be the vector of all monomials with support in $\mathscr{B}$. Then $f \in \mathring{\Sigma}[\mathbf{x}]$ if and only if there exists a positive definite matrix $\mathbf{G}$ such that $f = v_k^T \mathbf{G} v_k$.*

In the sequel, we state and analyze a hybrid numeric-symbolic algorithm, called `intsos`, computing weighted SOS decompositions of polynomials in $\mathbb{Z}[\mathbf{x}] \cap \mathring{\Sigma}[\mathbf{x}]$. This algorithm relies on perturbations of such polynomials. We first establish the following preliminary result.

**Proposition 2.3.3** *Let $f \in \mathbb{Z}[\mathbf{x}] \cap \mathring{\Sigma}[\mathbf{x}]$ of degree $d = 2k$, with $\tau = \tau(f)$, $P = \mathcal{C}(f)$ and $\mathscr{B} := P/2 \cap \mathbb{N}^n$. Then, there exists $N \in \mathbb{N} - \{0\}$ such that for $\varepsilon := \frac{1}{2^N}$, $f - \varepsilon \sum_{\alpha \in \mathscr{B}} \mathbf{x}^{2\alpha} \in \mathring{\Sigma}[\mathbf{x}]$, with $N \leq \tau(\varepsilon) \leq \mathcal{O}\,(\tau \cdot (4d+2)^{3n+3})$.*

The following can be found in [78, Lemma 2.1] and [78, Theorem 3.2].

**Proposition 2.3.4** *Let $\mathbf{G} \succ 0$ be a matrix with rational entries indexed on $\mathbb{N}_r^n$. Let $\mathbf{L}$ be the factor of $\mathbf{G}$ computed using Cholesky's decomposition with finite precision $\delta_c$. Then $\mathbf{L}\mathbf{L}^T = \mathbf{G} + \mathbf{F}$ where*

$$|\mathbf{F}_{\alpha,\beta}| \leq \frac{(r+1)2^{-\delta_c}|\mathbf{G}_{\alpha,\alpha}\,\mathbf{G}_{\beta,\beta}|^{\frac{1}{2}}}{1-(r+1)2^{-\delta_c}}\,. \tag{2.35}$$

*In addition, if the smallest eigenvalue $\tilde{\lambda}$ of $\mathbf{G}$ satisfies the inequality*

$$2^{-\delta_c} < \frac{\tilde{\lambda}}{r^2 + r + (r-1)\tilde{\lambda}}\,, \tag{2.36}$$

*Cholesky's decomposition returns a rational nonsingular factor $\mathbf{L}$.*

## Algorithm `intsos`

We present our algorithm `intsos` computing exact weighted rational SOS decompositions for polynomials in $\mathbb{Z}[\mathbf{x}] \cap \mathring{\Sigma}[\mathbf{x}]$.

Given $f \in \mathbb{Z}[\mathbf{x}]$ of degree $d = 2k$, one first computes its Newton polytope $P := \mathcal{C}\,(f)$ (see line 1) and $\mathscr{B} := P/2 \cap \mathbb{N}^n$ using standard algorithms such as quickhull by [55]. The loop going from line 3 to line 4 finds a positive $\varepsilon \in \mathbb{Q}$ such that the perturbed polynomial $f_\varepsilon := f - \varepsilon \sum_{\alpha \in \mathscr{B}} \mathbf{x}^{2\alpha}$ is also in $\mathring{\Sigma}[\mathbf{x}]$. This is done thanks to any external *oracle* deciding the nonnegativity of a polynomial. Even if this oracle is able to *decide* nonnegativity, we would like to emphasize that our algorithm outputs an SOS certificate in order to *certify* the nonnegativity of the input. In practice, we often choose the value of $\varepsilon$ while relying on a heuristic technique rather than this external oracle, for the sake of efficiency.

```
Require: $f \in \mathbb{Z}[\mathbf{x}]$, positive $\varepsilon \in \mathbb{Q}$, precision parameters $\delta, R \in \mathbb{N}$ for the SDP solver, precision
    $\delta_c \in \mathbb{N}$ for the Cholesky's decomposition
Ensure: list c_list of numbers in $\mathbb{Q}$ and list s_list of polynomials in $\mathbb{Q}[\mathbf{x}]$
 1: $P := \mathcal{C}(f), \mathscr{B} := P/2 \cap \mathbb{N}^n$
 2: $t := \sum_{\alpha \in \mathscr{B}} \mathbf{x}^{2\alpha}$, $f_\varepsilon \leftarrow f - \varepsilon t$
 3: while $f_\varepsilon \notin \mathring{\Sigma}[\mathbf{x}]$ do $\varepsilon \leftarrow \frac{\varepsilon}{2}$, $f_\varepsilon \leftarrow f - \varepsilon t$
 4: end while
 5: ok := false
 6: while not ok do
 7:     $(\tilde{\mathbf{G}}, \tilde{\lambda}) \leftarrow$ sdp$(f_\varepsilon, \delta, R)$
 8:     $(s_1, \dots, s_l) \leftarrow$ cholesky$(\tilde{\mathbf{G}}, \tilde{\lambda}, \delta_c)$          $\triangleright f_\varepsilon \simeq \sum_{i=1}^{l} s_i^2$
 9:     $u \leftarrow f_\varepsilon - \sum_{i=1}^{l} s_i^2$
10:     c_list $\leftarrow [1, \dots, 1]$, s_list $\leftarrow [s_1, \dots, s_l]$
11:     for $\alpha \in \mathscr{B}$ do $\varepsilon_\alpha := \varepsilon$
12:     end for
13:     c_list, s_list, $(\varepsilon_\alpha) \leftarrow$ absorb$(u, \mathscr{B}, (\varepsilon_\alpha)$, c_list, s_list$)$
14:     if $\min_{\alpha \in \mathscr{B}} \{\varepsilon_\alpha\} \geq 0$ then ok := true
15:     else $\delta \leftarrow 2\delta$, $R \leftarrow 2R$, $\delta_c \leftarrow 2\delta_c$
16:     end if
17: end while
18: for $\alpha \in \mathscr{B}$ do c_list $\leftarrow$ c_list $\cup \{\varepsilon_\alpha\}$, s_list $\leftarrow$ s_list $\cup \{\mathbf{x}^\alpha\}$
19: end for
20: return c_list, s_list
```

Figure 2.3: intsos

```
Require: $u \in \mathbb{Q}[\mathbf{x}]$, multi-index set $\mathscr{B}$, lists $(\varepsilon_\alpha)$ and c_list of numbers in $\mathbb{Q}$, list s_list of poly-
    nomials in $\mathbb{Q}[\mathbf{x}]$
Ensure: lists $(\varepsilon_\alpha)$ and c_list of numbers in $\mathbb{Q}$, list s_list of polynomials in $\mathbb{Q}[\mathbf{x}]$
for $\gamma \in \mathrm{supp}(u)$ do
    if $\gamma \in (2\mathbb{N})^n$ then $\alpha := \frac{\gamma}{2}$, $\varepsilon_\alpha := \varepsilon_\alpha + u_\gamma$
    else
        Find $\alpha, \beta \in \mathscr{B}$ such that $\gamma = \alpha + \beta$
        $\varepsilon_\alpha := \varepsilon_\alpha - \frac{|u_\gamma|}{2}$, $\varepsilon_\beta := \varepsilon_\beta - \frac{|u_\gamma|}{2}$
        c_list $\leftarrow$ c_list $\cup \{\frac{|u_\gamma|}{2}\}$
        s_list $\leftarrow$ s_list $\cup \{\mathbf{x}^\alpha + \mathrm{sgn}(u_\gamma)\mathbf{x}^\beta\}$
    end if
end for
```

Figure 2.4: absorb

If $f \in \mathbb{Z}[\mathbf{x}] \cap \mathring{\Sigma}[\mathbf{x}]$, then the set $\{e \in \mathbb{R}^{>0} : \forall \mathbf{x} \in \mathbb{R}^n, f(\mathbf{x}) - e\sum_{\alpha \in \mathscr{B}} \mathbf{x}^{2\alpha} \geq 0\}$ is non empty (see the proof of Proposition 2.3.3). If the oracle asserts that $\mathbf{x} \mapsto f(\mathbf{x}) - e\sum_{\alpha \in \mathscr{B}} \mathbf{x}^{2\alpha}$ is nonnegative on $\mathbb{R}^n$, then $e$ belongs to this set and it is enough to select $\varepsilon = e/2$ to ensure that $f_\varepsilon := f - \varepsilon\sum_{\alpha \in \mathscr{B}} \mathbf{x}^{2\alpha} \in \mathring{\Sigma}[\mathbf{x}]$.

Next, we enter in the loop starting from line 6. Given $f_\varepsilon \in \mathbb{Z}[\mathbf{x}]$, positive integers $\delta$ and $R$, the sdp function calls an SDP solver and tries to compute a rational approximation $\tilde{\mathbf{G}}$ of the Gram matrix associated to $f_\varepsilon$ together with a rational approximation $\tilde{\lambda}$ of its smallest eigenvalue.

In order to analyse the complexity of the procedure (see Remark 2.3.1), we assume that sdp relies on the ellipsoid algorithm by [206].

**Remark 2.3.1** *In [159], the authors analyze the complexity of the short step, primal interior point method, used in SDP solvers. Within fixed accuracy, they obtain a polynomial complexity, as for the ellipsoid method, but the exact value of the exponents is not provided.*

*Also, in practice, we use an arbitrary-precision SDP solver implemented with an interior-point method.*

SDP problems are solved with this latter algorithm in polynomial-time within a given accuracy $\delta$ and a radius bound $R$ on the Frobenius norm of $\tilde{\mathbf{G}}$. The first step consists of solving SDP (2.34) by computing an approximate Gram matrix $\tilde{\mathbf{G}} \succeq 2^{-\delta} I$ such that

$$|\langle \tilde{\mathbf{G}}, \mathbf{B}_\gamma \rangle - (f_\varepsilon)_\gamma| = |\sum_{\alpha+\beta=\gamma} \tilde{\mathbf{G}}_{\alpha,\beta} - (f_\varepsilon)_\gamma| \leq 2^{-\delta}$$

and $\sqrt{\langle \tilde{\mathbf{G}}, \tilde{\mathbf{G}} \rangle} \leq R$. We pick large enough integers $\delta$ and $R$ to obtain $\tilde{\mathbf{G}} \succ 0$ and $\tilde{\lambda} > 0$ when $f_\varepsilon \in \mathring{\Sigma}[\mathbf{x}]$.

The `cholesky` function computes the approximate Cholesky's decomposition $\mathbf{L}\mathbf{L}^T$ of $\tilde{\mathbf{G}}$ with precision $\delta_c$. In order to guarantee that $\mathbf{L}$ will be a rational nonsingular matrix, a preliminary step consists of verifying that the inequality (2.36) holds, which happens when $\delta_c$ is large enough. Otherwise, `cholesky` selects the smallest $\delta_c$ such as (2.36) holds. Let $v_k$ be the size $l$ vector of all monomials $\mathbf{x}^\alpha$ with $\alpha$ belonging to $\mathscr{B}$. The output is a list of rational polynomials $[s_1, \dots, s_l]$ such that for all $i \in [l]$, $s_i$ is the inner product of the $i$-th row of $\mathbf{L}$ by $v_k$. By Theorem 2.3.1, one would have $f_\varepsilon = \sum_{i=1}^l s_i^2$ with $s_i \in \mathbb{R}[\mathbf{x}]$ after using exact SDP and Cholesky's decomposition. Here, we have to consider the remainder $u = f - \varepsilon \sum_{\alpha \in \mathscr{B}} \mathbf{x}^{2\alpha} - \sum_{i=1}^l s_i^2$, with $s_i \in \mathbb{Q}[\mathbf{x}]$.

After these steps, the algorithm starts to perform symbolic computation with the `absorb` subroutine at line 13. The loop from `absorb` is designed to obtain an exact weigthed SOS decomposition of $\varepsilon t + u = \varepsilon \sum_{\alpha \in \mathscr{B}} \mathbf{x}^{2\alpha} + \sum_\gamma u_\gamma \mathbf{x}^\gamma$, yielding in turn an exact decomposition of $f$. Each term $u_\gamma \mathbf{x}^\gamma$ can be written either $u_\gamma \mathbf{x}^{2\alpha}$ or $u_\gamma \mathbf{x}^{\alpha+\beta}$, for $\alpha, \beta \in \mathscr{B}$. In the former case (line 2), one has

$$\varepsilon \mathbf{x}^{2\alpha} + u_\gamma \mathbf{x}^{2\alpha} = (\varepsilon + u_\gamma) \mathbf{x}^{2\alpha} .$$

In the latter case (line 4), one has

$$\varepsilon(\mathbf{x}^{2\alpha} + \mathbf{x}^{2\beta}) + u_\gamma \mathbf{x}^{\alpha+\beta} = |u_\gamma|/2(\mathbf{x}^\alpha + \operatorname{sgn}(u_\gamma) \mathbf{x}^\beta)^2 + (\varepsilon - |u_\gamma|/2)(\mathbf{x}^{2\alpha} + \mathbf{x}^{2\beta}) .$$

If the positivity test of line 14 fails, then the coefficients of $u$ are too large and one cannot ensure that $\varepsilon t + u$ is SOS. So we repeat the same procedure after increasing the precision of the SDP solver and Cholesky's decomposition.

In prior work [J17], outlined in Section 2.2, we formalized and analyzed the so-called `univsos2` algorithm, initially provided in [61]. Given a univariate polynomial $f > 0$ of degree $d = 2k$, this algorithm computes weighted SOS decompositions of $f$. With $t := \sum_{i=0}^k \mathbf{x}^{2i}$, the first numeric step of `univsos2` is to find $\varepsilon$ such that the perturbed polynomial $f_\varepsilon := f - \varepsilon t > 0$ and to compute its complex roots, yielding an approximate SOS decomposition $s_1^2 + s_2^2$. The second symbolic step is very similar to the loop from line 1 to line 9 in `intsos`: one considers the remainder polynomial $u := f_\varepsilon - s_1^2 - s_2^2$ and tries to computes an exact SOS decomposition of $\varepsilon t + u$. This succeeds for large enough precision of the root isolation procedure. Therefore, `intsos` can be seen as an extension of `univsos2` in the multivariate case by replacing the numeric step of root isolation by SDP and keeping the same symbolic step.

**Example 2.3.1** *We apply Algorithm* `intsos` *on*

$$f = 4x_1^4 + 4x_1^3 x_2 - 7x_1^2 x_2^2 - 2x_1 x_2^3 + 10x_2^4 ,$$

*with $\varepsilon = 1$, $\delta = R = 60$ and $\delta_c = 10$. Then*

$$\mathscr{B} := \mathcal{C}(f)/2 \cap \mathbb{N}^n = \{(2,0),(1,1),(0,2)\}$$

*(line 1). The loop from line 3 to line 4 ends and we get $f - \varepsilon t = f - (x_1^4 + x_1^2x_2^2 + x_2^4) \in \mathring{\Sigma}[\mathbf{x}]$. The* `sdp` *(line 7) and* `cholesky` *(line 8) procedures yield*

$$s_1 = 2x_1^2 + x_1x_2 - \frac{8}{3}x_2^2, \quad s_2 = \frac{4}{3}x_1x_2 + \frac{3}{2}x_2^2 \quad \text{and} \quad s_3 = \frac{2}{7}x_2^2.$$

*The remainder polynomial is $u = f - \varepsilon t - s_1^2 - s_2^2 - s_3^2 = -x_1^4 - \frac{1}{9}x_1^2x_2^2 - \frac{2}{3}x_1x_2^3 - \frac{781}{1764}x_2^4$.*

*At the end of the loop from line 1 to line 9, we obtain $\varepsilon_{(2,0)} = (\varepsilon - x_1^4 = 0$, which is the coefficient of $x_1^4$ in $\varepsilon t + u$. Then,*

$$\varepsilon(x_1^2x_2^2 + x_2^4) - \frac{2}{3}x_1x_2^3 = \frac{1}{3}(x_1x_2 - x_2^2)^2 + (\varepsilon - \frac{1}{3})(x_1^2x_2^2 + x_2^4).$$

*In the polynomial $\varepsilon t + u$, the coefficient of $x_1^2x_2^2$ is $\varepsilon_{(1,1)} = \varepsilon - \frac{1}{3} - \frac{1}{9} = \frac{5}{9}$ and the coefficient of $x_4^4$ is $\varepsilon_{(0,2)} = \varepsilon - \frac{1}{3} - \frac{781}{1764} = \frac{395}{1764}$.*

*Eventually, we obtain the weighted rational SOS decomposition:*

$$\begin{aligned} 4x_1^4 + 4x_1^3x_2 - 7x_1^2x_2^2 - 2x_1x_2^3 + 10x_2^4 = & \frac{1}{3}(x_1x_2 - x_2^2)^2 + \frac{5}{9}(x_1x_2)^2 + \frac{395}{1764}x_2^4 \\ & + (2x_1^2 + x_1x_2 - \frac{8}{3}x_2^2)^2 + (\frac{4}{3}x_1x_2 + \frac{3}{2}x_2^2)^2 + (\frac{2}{7}x_2^2)^2) . \end{aligned}$$

## Bit complexity analysis

Let $f \in \mathbb{Z}[\mathbf{x}] \cap \mathring{\Sigma}[\mathbf{x}]$ of degree $d = 2k$, $\tau := \tau(f)$ and $\mathscr{B} := \mathcal{C}(f)/2 \cap \mathbb{N}^n$.

**Proposition 2.3.5** *Let $\mathbf{G}$ be a positive definite Gram matrix associated to $f$ and take $0 < \varepsilon \in \mathbb{Q}$ as in Proposition 2.3.3 so that $f_\varepsilon = f - \varepsilon \sum_{\alpha \in \mathscr{B}} \mathbf{x}^{2\alpha} \in \mathring{\Sigma}[\mathbf{x}]$. Then, there exist positive integers $\delta$, $R$ such that $\mathbf{G} - \varepsilon \mathrm{I}$ is a Gram matrix associated to $f_\varepsilon$, satisfies $\mathbf{G} - \varepsilon \mathrm{I} \succeq 2^{-\delta} I$ and $\sqrt{\langle \mathbf{G} - \varepsilon \mathrm{I}, \mathbf{G} - \varepsilon \mathrm{I} \rangle} \leq R$. Also, the maximal bitsizes of $\delta$ and $R$ are upper bounded by $\mathcal{O}(\tau \cdot (4d+2)^{3n+3})$ and $\mathcal{O}(\tau \cdot (4d+2)^{4n+3})$, respectively.*

**Proposition 2.3.6** *Let $f$ be as above. When applying Algorithm* `intsos` *to $f$, the procedure always terminates and outputs a weighted SOS decomposition of $f$ with rational coefficients. The maximum bitsize of the coefficients involved in this SOS decomposition is upper bounded by $\mathcal{O}(\tau \cdot (4d+2)^{4n+3})$.*

**Theorem 2.3.2** *For $f$ as above, there exist $\varepsilon, \delta, R, \delta_c$ of bitsizes upper bounded by $\mathcal{O}(\tau \cdot (4d+2)^{4n+3})$ such that* `intsos`*$(f, \varepsilon, \delta, R, \delta_c)$ runs in boolean time $\mathcal{O}(\tau^2 \cdot (4d+2)^{15n+6})$.*

## Comparison with the rounding-projection algorithm of Peyrl and Parrilo

We recall the algorithm designed in [237]. We denote this rounding-projection algorithm by `RoundProject`.

The first main step in Step 5 consists of rounding the approximation $\tilde{\mathbf{G}}$ of a Gram matrix associated to $f$ up to precision $\delta_i$. The second main step in Step 8 consists of computing the orthogonal projection $\mathbf{G}$ of $\mathbf{G}'$ on an adequate affine subspace in such a way that $\sum_{\alpha+\beta=\gamma} \mathbf{G}_{\alpha,\beta} = f_\gamma$,

```
**Require:** $f \in \mathbb{Z}[\mathbf{x}]$, rounding precision $\delta_i \in \mathbb{N}$, precision parameters $\delta, R \in \mathbb{N}$ for the SDP solver
**Ensure:** list c_list of numbers in $\mathbb{Q}$ and list s_list of polynomials in $\mathbb{Q}[\mathbf{x}]$
 1: $P := \mathcal{C}(f), \mathscr{B} := P/2 \cap \mathbb{N}^n$
 2: ok := false
 3: while not ok do
 4:     $(\tilde{\mathbf{G}}, \tilde{\lambda}) \leftarrow$ sdp$(f, \delta, R)$
 5:     $\mathbf{G}' \leftarrow$ round $\left(\tilde{\mathbf{G}}, \delta_i\right)$
 6:     for $\alpha, \beta \in \mathscr{B}$ do
 7:         $\eta(\alpha+\beta) \leftarrow \#\{(\alpha', \beta') \in \mathscr{B}^2 \mid \alpha' + \beta' = \alpha + \beta\}$
 8:         $\mathbf{G}(\alpha, \beta) := \mathbf{G}'(\alpha, \beta) - \frac{1}{\eta(\alpha+\beta)} \left(\sum_{\alpha'+\beta'=\alpha+\beta} \mathbf{G}'(\alpha', \beta') - f_{\alpha+\beta}\right)$
 9:     end for
10:     $(c_1, \dots, c_l, s_1, \dots, s_l) \leftarrow$ ldl$(\mathbf{G})$                    ▷ $f = \sum_{i=1}^{l} c_i s_i^2$
11:     if $c_1, \dots, c_l \in \mathbb{Q}_{>0}, s_1, \dots, s_l \in \mathbb{Q}[\mathbf{x}]$ then ok := true
12:     else $\delta \leftarrow 2\delta$, $R \leftarrow 2R$, $\delta_c \leftarrow 2\delta_c$
13:     end if
14: end while
15: c_list $\leftarrow [c_1, \dots, c_l]$, s_list $\leftarrow [s_1, \dots, s_l]$
16: return c_list, s_list
```

Figure 2.5: `RoundProject`

for all $\gamma \in \operatorname{supp}(f)$. For more details on this orthogonal projection, we refer to [237, Proposition 7]. The algorithm then performs in (10) an exact diagonalization of the matrix $\mathbf{G}$ via the $\mathbf{LDL}^T$ decomposition (see, e.g., [97, § 4.1]). It is proved in [237, Proposition 8] that for $f \in \mathring{\Sigma}[\mathbf{x}]$, Algorithm `RoundProject` returns a weighted SOS decomposition of $f$ with rational coefficients when the precision of the rounding and SDP solving steps are large enough. The main differences w.r.t. Algorithm `intsos` are that `RoundProject` does not perform a perturbation of the input polynomial $f$ and computes an exact $\mathbf{LDL}^T$ decomposition of a Gram matrix $\mathbf{G}$. In our case, we compute an approximate Cholesky's decomposition of $\tilde{\mathbf{G}}$ instead of a projection, then perform an exact compensation of the error terms, thanks to the initial perturbation.

The next result gives upper bounds on the bitsize of the coefficients involved in the SOS decomposition returned by `RoundProject` as well as on the boolean running time. Even though `intsos` and `RoundProject` have the same exponential bit complexity, the upper bound estimates are larger in the case of `RoundProject`. It would be worth investigating if these bounds are tight in general.

**Theorem 2.3.3** *For $f$ as above, there exist $\delta_i$, $\delta$, $R$ of bitsizes $\leq \mathcal{O}\left(\tau \cdot (4d+2)^{4n+3}\right)$ such that `RoundProject`$(f, \delta_i, \delta, R)$ outputs a rational SOS decomposition of $f$ with rational coefficients. The maximum bitsize of the coefficients involved in this SOS decomposition is upper bounded by $\mathcal{O}\left(\tau \cdot (4d+2)^{6n+3}\right)$ and the boolean running time is $\mathcal{O}\left(\tau^2 \cdot (4d+2)^{15n+6}\right)$.*

## Exact Reznick's representations

Next, we show how to apply Algorithm `intsos` to decompose positive definite forms into SOS of rational functions.

Let $G_n := \sum_{i=1}^{n} x_i^2$ and let us consider the unit $(n-1)$-sphere $\{\mathbf{x} \in \mathbb{R}^n : G_n(\mathbf{x}) = 1\}$. A positive definite form $f \in \mathbb{R}[\mathbf{x}]$ is a homogeneous polynomial which is positive over the unit $(n-1)$-sphere.

```
Require: $f \in \mathbb{Z}[\mathbf{x}]$, positive $\varepsilon \in \mathbb{Q}$, precision parameters $\delta, R \in \mathbb{N}$ for the SDP solver, precision
    $\delta_c \in \mathbb{N}$ for the Cholesky's decomposition
Ensure: list c_list of numbers in $\mathbb{Q}$ and list s_list of polynomials in $\mathbb{Q}[\mathbf{x}]$
$r := 0$
while interiorSOScone($f\, G_n, r$) = false do $r \leftarrow r + 1$
end while
return intsos($f\, G_n^r, \varepsilon, \delta, R, \delta_c$)
```

Figure 2.6: `Reznicksos`

For such a form, we define $\varepsilon(f)$ as the ratio between the minimum and maximum values of $f$ on the unit $(n-1)$-sphere. This ratio measures how close $f$ is to having a zero in the unit sphere. While there is no guarantee that $f \in \Sigma[\mathbf{x}]$, [17] proved that for large enough $r \in \mathbb{N}$, $f G_n^r \in \Sigma[\mathbf{x}]$. Such SOS decompositions are called *Reznick's representations* and $2r$ is called the *Reznick's degree*. The next result states that for large enough $r \in \mathbb{N}$, $f G_n^r \in \mathring{\Sigma}[\mathbf{x}]$, as a direct consequence of [17].

**Lemma 2.3.7** *Let $f$ be a positive definite form of degree $d = 2k$ in $\mathbb{Z}[\mathbf{x}]$ and $r \geq \frac{nd(d-1)}{4 \log 2\, \varepsilon(f)} - \frac{n+d}{2} + 1$. Then $f\, G_n^r \in \mathring{\Sigma}[\mathbf{x}]$.*

Algorithm `Reznicksos` takes as input $f \in \mathbb{Z}[\mathbf{x}]$, finds the smallest $r \in \mathbb{N}$ such that $f\, G_n^r \in \mathring{\Sigma}[\mathbf{x}]$, thanks to an oracle which decides if some given polynomial is a positive definite form. Further, we denote by `interiorSOScone` a routine which takes as input $f, G_n$ and $r$ and returns `true` if and only if $f\, G_n^r \in \mathring{\Sigma}[\mathbf{x}]$, else it returns false. Then, `intsos` is applied on $f\, G_n^r$.

**Example 2.3.2** *Let us apply `Reznicksos` on the perturbed Motzkin polynomial*

$$f = (1 + 2^{-20})(x_3^6 + x_1^4 x_2^2 + x_1^2 x_2^4) - 3 x_1^2 x_2^2 x_3^2.$$

*With $r = 1$, one has $f\, G_n = (x_1^2 + x_2^2 + x_3^2)\, f \in \mathring{\Sigma}[\mathbf{x}]$ and `intsos` yields an SOS decomposition of $f\, G_n$ with $\varepsilon = 2^{-20}$, $\delta = R = 60$, $\delta_c = 10$.*

**Theorem 2.3.4** *Let $f \in \mathbb{Z}[\mathbf{x}]$ be a positive definite form of degree $d$, coefficients of bitsize at most $\tau$. On input $f$, Algorithm `Reznicksos` terminates and outputs a weighted SOS decomposition for $f$. The maximum bitsize of the coefficients involved in the decomposition and the boolean running time of the procedure are both upper bounded by $2^{\mathcal{O}\,(\tau \cdot (4d+2)^{4n+3})}$.*

The bit complexity of `Reznicksos` is polynomial in the Reznick's degree $2r$ of the representation. In all the examples we tackled, this degree was rather small as shown.

### Exact Putinar's representations

We let $f, g_1, \ldots, g_m$ in $\mathbb{Z}[\mathbf{x}]$ of degrees less than $d \in \mathbb{N}$ and $\tau \in \mathbb{N}$ be a bound on the bitsize of their coefficients. Assume that $f$ is positive over $\mathbf{X} := \{\mathbf{x} \in \mathbb{R}^n : g_1(\mathbf{x}) \geq 0, \ldots, g_m(\mathbf{x}) \geq 0\}$ and reaches its infimum with $f_{\min} := \min_{\mathbf{x} \in \mathbf{X}} f(\mathbf{x}) > 0$. With $f = \sum_{|\alpha| \leq d} f_\alpha \mathbf{x}^\alpha$, we set $\|f\| := \max_{|\alpha| \leq d} \frac{f_\alpha \alpha_1! \cdots \alpha_n!}{|\alpha|!}$ and $g_0 := 1$.

As usual, we consider the quadratic module $\mathcal{M}(\mathbf{X}) := \left\{\sum_{j=0}^m \sigma_j g_j : \sigma_j \in \Sigma[\mathbf{x}]\right\}$ and, for $r \in \mathbb{N}$, the $2r$-truncated quadratic module $\mathcal{M}(\mathbf{X})_r := \left\{\sum_{j=0}^m \sigma_j g_j : \sigma_j \in \Sigma[\mathbf{x}]\,,\ \deg(\sigma_j g_j) \leq 2r\right\}$ generated by $g_1, \ldots, g_m$. As done several times earlier on, we rely on the following assumption.

**Assumption 2.3.8** *The set $\mathbf{X}$ is a basic compact semialgebraic set with nonempty interior, included in $[-1,1]^n$ and $\mathcal{M}(\mathbf{X})$ is archimedean.*

Under Assumption 2.3.8, $f$ is positive over $\mathbf{X}$ only if $f \in \mathcal{M}(\mathbf{X})_r$ for some $r \in \mathbb{N}$ (see [245]). In this case, there exists a Putinar's representation $f = \sum_{i=0}^m \sigma_j g_j$ with $\sigma_j \in \Sigma[\mathbf{x}]$ for $0 \leq j \leq m$. One can certify that $f \in \mathcal{M}(\mathbf{X})_r$ by solving the next SDP with $r \geq \lceil d/2 \rceil$:

$$\begin{aligned} \sup_{\mathbf{G}_j, b} \{ b : \quad & f_\alpha - b\, 1_{\alpha=0} = \sum_{j=0}^m \langle \mathbf{C}_\alpha^j, \mathbf{G}_j \rangle \,, \quad \alpha \in \mathbb{N}_{2r}^n \,, \\ & \mathbf{G}_j \succeq 0 \,, \quad j = 0, \ldots, m \, \} \end{aligned} \,, \tag{2.37}$$

which the same as SDP (2.7) from Section 2.1. SDP (2.37) is a reformulation of the problem

$$f_{\min}^r := \sup\{ b : f - b \in \mathcal{M}(\mathbf{X})_r \} \,.$$

Thus $f_{\min}^r$ is also the optimal value of SDP (2.37). The next result follows from [180, Theorem 4.2]:

**Theorem 2.3.9** *We use the notation introduced above. Under Assumption 2.3.8, for $r \in \mathbb{N}$ large enough, one has*

$$0 < f_{\min}^r \leq f_{\min} \,.$$

*In addition, SDP (2.37) has an optimal solution $(\mathbf{G}_0, \mathbf{G}_1, \ldots, \mathbf{G}_m)$, yielding the following Putinar's representation:*

$$f - f_{\min}^r = \sum_{i=1}^{l} \lambda_{i0} q_{i0}^2 + \sum_{j=1}^{m} g_j \sum_{i=1}^{l_j} \lambda_{ij} q_{ij}^2 \,,$$

*where the vectors of coefficients of the polynomials $q_{ij}$ are the eigenvectors of $\mathbf{G}_j$ with respective eigenvalues $\lambda_{ij}$, for all $j = 0, \ldots, m$.*

The complexity of Putinar's Positivstellensätz was analyzed by [226]:

**Theorem 2.3.10** *With the notation and assumptions introduced above, there exists a real $\chi_{\mathbf{X}} > 0$ depending on $\mathbf{X}$ such that*

*(i) for all even $2r \geq \chi_{\mathbf{X}} \exp\big(d^2 n^d \frac{\|f\|}{f_{\min}}\big)^{\chi_{\mathbf{X}}}$, $f \in \mathcal{M}(\mathbf{X})_r$.*

*(ii) for all even $2r \geq \chi_{\mathbf{X}} \exp\big(2d^2 n^d\big)^{\chi_{\mathbf{X}}}$, $0 \leq f_{\min} - f_{\min}^r \leq \frac{6d^3 n^{2d} \|f\|}{\chi_{\mathbf{X}} \sqrt{\log \frac{2r}{\chi_{\mathbf{X}}}}}$.*

From a computational viewpoint, one can certify that $f$ lies in $\mathcal{M}(\mathbf{X})_r$ for $r$ large enough, by solving SDP (2.37). Next, we show how to ensure the existence of a strictly feasible solution for SDP (2.37) after replacing the set defined by our initial constraints $\mathbf{X}$ by the following one

$$\mathbf{X}' := \{ \mathbf{x} \in \mathbf{X} : 1 - \mathbf{x}^{2\alpha} \geq 0 \,, \forall \alpha \in \mathbb{N}_r^n \} \,.$$

We first give a lower bound for $f_{\min}$.

**Proposition 2.3.11** *With the above notation and assumptions, one has*

$$f_{\min} \geq 2^{-(\tau + d + d \log_2 n + 1) d^{n+1}} d^{-(n+1) d^{n+1}} \geq 2^{-\mathcal{O}(\tau \cdot d^{2n+2})} \,.$$

**Theorem 2.3.5** *We use the notation and assumptions introduced above. There exists $r \in \mathbb{N}$ such that:*
*(i) $f \in \mathcal{M}(\mathbf{X})_r$ with the representation*

$$f = f^r_{\min} + \sum_{j=0}^{m} \sigma_j g_j \,,$$

*for $f^r_{\min} > 0$, $\sigma_j \in \Sigma[\mathbf{x}]$ with $\deg(\sigma_j g_j) \leq 2r$ for all $j = 0, \ldots, m$.*
*(ii) $f \in \mathcal{M}(\mathbf{X}')_r$ with the representation*

$$f = \sum_{j=0}^{m} \mathring{\sigma}_j g_j + \sum_{|\alpha| \leq r} c_\alpha (1 - \mathbf{x}^{2\alpha}) \,,$$

*for $\mathring{\sigma}_j \in \mathring{\Sigma}[\mathbf{x}]$ with $\deg(\mathring{\sigma}_j g_j) \leq 2r$, for all $j = 0, \ldots, m$, and some sequence of positive numbers $(c_\alpha)_{|\alpha| \leq r}$.*
*(iii) There exists a real $C_{\mathbf{X}} > 0$ depending on $\mathbf{X}$ and $\varepsilon = \frac{1}{2^N}$ with positive $N \in \mathbb{N}$ such that*

$$f - \varepsilon \sum_{|\alpha| \leq r} \mathbf{x}^{2\alpha} \in \mathcal{M}(\mathbf{X}')_r \,, \quad N \leq 2^{C_{\mathbf{X}} \tau d^{2n+2}} \,,$$

*where $\tau$ is the maximal bitsize of the coefficients of $f, g_1, \ldots, g_m$.*

## Algorithm `Putinarsos`

We can now present Algorithm `Putinarsos`.

For $f \in \mathbb{Z}[\mathbf{x}]$ positive over a basic compact semialgebraic set $\mathbf{X}$ satisfying Assumption 2.3.8, the first loop outputs the smallest positive integer $2r$ such that $f \in \mathcal{M}(\mathbf{X})_r$.
Then the procedure is similar to `intsos`. As for the first loop of `intsos`, the loop from line 6 to line 7 allows us to obtain a perturbed polynomial $f_\varepsilon \in \mathcal{M}(\mathbf{X}')_r$, with $\mathbf{X}' := \{\mathbf{x} \in \mathbf{X} : 1 - \mathbf{x}^{2\alpha} \geq 0, \forall \alpha \in \mathbb{N}^n_r\}$.
Then one solves SDP (2.37) with the `sdp` procedure and performs Cholesky's decomposition to obtain an approximate Putinar's representation of $f_\varepsilon = f - \varepsilon t$ and a remainder $u$.
Next, we apply the `absorb` subroutine as in `intsos`. The rationale is that with large enough precision parameters for the procedures `sdp` and `cholesky`, one finds an exact weighted SOS decomposition of $u + \varepsilon t$, which yields in turn an exact Putinar's representation of $f$ in $\mathcal{M}(\mathbf{X}')_r$ with rational coefficients.

**Example 2.3.3** *Let us apply `Putinarsos` to $f = -x_1^2 - 2x_1x_2 - 2x_2^2 + 6$, $\mathbf{X} := \{(x_1, x_2) \in \mathbb{R}^2 : 1 - x_1^2 \geq 0, 1 - x_2^2 \geq 0\}$ and the same precision parameters as in Example 2.3.1. The first and second loop yield $r = 1$ and $\varepsilon = 1$. After running `absorb`, we obtain the exact Putinar's representation*

$$f = \frac{23853407}{292204836} + \frac{23}{49} x_1^2 + \frac{130657269}{291009481} x_2^2 + \frac{1}{2442^2} + (x_1 - x_2)^2 + (\frac{x_2}{2437})^2 + \left(\frac{11}{7}\right)^2 (1 - x_1^2) \\ + \left(\frac{13}{7}\right)^2 (1 - x_2^2) \,.$$

```
Require: $f, g_1, \dots, g_m \in \mathbb{Z}[\mathbf{x}]$ of degrees less than $d \in \mathbb{N}$, $\mathbf{X} := \{\mathbf{x} \in \mathbb{R}^n : g_1(\mathbf{x}) \geq 0, \dots, g_m(\mathbf{x}) \geq 0\}$, positive $\varepsilon \in \mathbb{Q}$, precision parameters $\delta, R \in \mathbb{N}$ for the SDP solver , precision $\delta_c \in \mathbb{N}$ for the Cholesky's decomposition
Ensure: lists c_list$_0, \dots,$ c_list$_m$, c_alpha of numbers in $\mathbb{Q}$ and lists s_list$_0, \dots,$ s_list$_m$ of polynomials in $\mathbb{Q}[\mathbf{x}]$
r ← $\lceil d/2 \rceil$, $g_0 := 1$
while $f \notin \mathcal{M}(\mathbf{X})_r$ do $r \leftarrow r + 1$
end while
$P := \mathbb{N}^n_{2r}$, $\mathscr{B} := \mathbb{N}^n_r$, $\mathbf{X}' := \{\mathbf{x} \in \mathbf{X} : 1 - \mathbf{x}^{2\alpha} \geq 0, \forall \alpha \in \mathbb{N}^n_r\}$
$t := \sum_{\alpha \in \mathscr{B}} \mathbf{x}^{2\alpha}$, $f_\varepsilon \leftarrow f - \varepsilon t$
while $f_\varepsilon \notin \mathcal{M}(\mathbf{X}')_r$ do $\varepsilon \leftarrow \frac{\varepsilon}{2}$, $f_\varepsilon \leftarrow f - \varepsilon t$
end while
ok := false
while not ok do
    $[\tilde{G}_0, \dots, \tilde{\mathbf{G}}_m, \tilde{\lambda}_0, \dots, \tilde{\lambda}_m, (\tilde{c}_\alpha)_{|\alpha| \leq r}], \leftarrow$ sdp($f_\varepsilon, \delta, R, \mathbf{X}'$)
    c_alpha ← $(\tilde{c}_\alpha)_{|\alpha| \leq r}$
    for $j \in \{0, \dots, m\}$ do
        $(s_{1j}, \dots, s_{l_j j}) \leftarrow$ cholesky($\tilde{\mathbf{G}}_j, \tilde{\lambda}_j, \delta_c$), $\tilde{\sigma}_j := \sum_{i=1}^{l_j} s_{ij}^2$
        c_list$_j$ ← $[1, \dots, 1]$, s_list$_j$ ← $[s_{1j}, \dots, s_{l_j j}]$
    end for
    $u \leftarrow f_\varepsilon - \sum_{j=0}^m \tilde{\sigma}_j g_j - \sum_{|\alpha| \leq r} \tilde{c}_\alpha (1 - \mathbf{x}^{2\alpha})$
    for $\alpha \in \mathscr{B}$ do $\varepsilon_\alpha := \varepsilon$
    end for
    c_list, s_list, $(\varepsilon_\alpha)$ ← absorb($u, \mathscr{B}, (\varepsilon_\alpha)$, c_list, s_list)
    if $\min_{\alpha \in \mathscr{B}} \{\varepsilon_\alpha\} \geq 0$ then ok := true
    else $\delta \leftarrow 2\delta$, $R \leftarrow 2R$, $\delta_c \leftarrow 2\delta_c$
    end if
end while
for $\alpha \in \mathscr{B}$ do
    c_list$_0$ ← c_list$_0 \cup \{\varepsilon_\alpha\}$, s_list$_0$ ← s_list$_0 \cup \{\mathbf{x}^\alpha\}$
end for
return c_list$_0, \dots,$ c_list$_m$, c_alpha, s_list$_0, \dots,$ s_list$_m$
```

Figure 2.7: `Putinarsos`

## Bit complexity analysis

**Theorem 2.3.6** *We use the notation and assumptions introduced above. For some constants $C_{\mathbf{X}} > 0$ and $K_{\mathbf{X}}$ depending on S, there exist $\varepsilon$, $\delta$, R, $\delta_c$ and r of bitsizes less than $\mathcal{O}\left(2^{C_{\mathbf{X}} \tau d^{2n+2}}\right)$ for which `Putinarsos`$(f, \mathbf{X}, \varepsilon, \delta, R, \delta_c)$ terminates and outputs an exact Putinar's representation with rational coefficients of $f \in \mathcal{Q}(\mathbf{X}')$, with $\mathbf{X}' := \{\mathbf{x} \in \mathbf{X} : 1 - \mathbf{x}^{2\alpha} \geq 0, \forall \alpha \in \mathbb{N}^n_r\}$. The maximum bitsize of these coefficients is bounded by $\mathcal{O}\left(2^{C_{\mathbf{X}} \tau d^{2n+2}}\right)$ and the procedure runs in boolean time $\mathcal{O}\left(2^{2^{K_{\mathbf{X}} \tau d^{2n+2}}}\right)$.*

As for `Reznicksos`, the complexity is polynomial in the degree $2r$ of the representation. On all the examples we tackled, it was close to the degrees of the involved polynomials.

```
Require: $f, g_1, \dots, g_m \in \mathbb{Z}[\mathbf{x}]$ of degrees less than $d \in \mathbb{N}$, $\mathbf{X} := \{\mathbf{x} \in \mathbb{R}^n : g_1(\mathbf{x}) \geq 0, \dots, g_m(\mathbf{x}) \geq 0\}$, rounding precision $\delta_i \in \mathbb{N}$, precision parameters $\delta, R \in \mathbb{N}$ for the SDP solver, precision $\delta_c \in \mathbb{N}$ for the Cholesky's decomposition
Ensure: lists $\texttt{c\_list}_0, \dots, \texttt{c\_list}_m$ of numbers in $\mathbb{Q}$ and lists $\texttt{s\_list}_0, \dots, \texttt{s\_list}_m$ of polynomials in $\mathbb{Q}[\mathbf{x}]$
 1: $r \leftarrow \lceil d/2 \rceil$, $g_0 := 1$
 2: while $f \notin \mathcal{M}(\mathbf{X})_r$ do $r \leftarrow r + 1$
 3: end while
 4: ok := false
 5: while not ok do
 6:     $[\tilde{\mathbf{G}}_0, \dots, \tilde{\mathbf{G}}_m, \tilde{\lambda}_0, \dots, \tilde{\lambda}_m], \leftarrow \texttt{sdp}(f, \delta, R, \mathbf{X})$
 7:     $\mathbf{G}' \leftarrow \texttt{round}\left(\tilde{\mathbf{G}}_0, \delta_i\right)$
 8:     for $j \in [m]$ do
 9:         $(s_{1j}, \dots, s_{l_j j}) \leftarrow \texttt{cholesky}(\tilde{\mathbf{G}}_j, \tilde{\lambda}_j, \delta_c)$, $\tilde{\sigma}_j := \sum_{i=1}^{l_j} s_{ij}^2$
10:         $\texttt{c\_list}_j \leftarrow [1, \dots, 1]$, $\texttt{s\_list}_j \leftarrow [s_{1j}, \dots, s_{l_j j}]$
11:     end for
12:     $u \leftarrow f - \sum_{j=1}^m \tilde{\sigma}_j$
13:     $\mathscr{B} := \mathbb{N}_r^n$
14:     for $\alpha, \beta \in \mathscr{B}$ do $\eta(\alpha + \beta) \leftarrow \#\{(\alpha', \beta') \in \mathscr{B}^2 \mid \alpha' + \beta' = \alpha + \beta\}$
15:         $\mathbf{G}(\alpha, \beta) := \mathbf{G}'(\alpha, \beta) - \frac{1}{\eta(\alpha+\beta)}\left(\sum_{\alpha'+\beta'=\alpha+\beta} \mathbf{G}'(\alpha', \beta') - u_{\alpha+\beta}\right)$
16:     end for
17:     $(c_{10}, \dots, c_{l_0 0}, s_{10}, \dots, s_{l_0 0}) \leftarrow \texttt{ldl}(\mathbf{G})$      $\triangleright\ f = \sum_{i=1}^{l_0} c_{i0} s_{i0}^2 + \sum_{j=1}^m \tilde{\sigma}_j$
18:     if $c_{10}, \dots, c_{m l_m} \in \mathbb{Q}_{\geq 0}, s_{01}, \dots, s_{m l_m} \in \mathbb{Q}[\mathbf{x}]$ then ok := true
19:     else $\delta_i \leftarrow 2\delta_i,\ \delta \leftarrow 2\delta,\ R \leftarrow 2R,\ \delta_c \leftarrow 2\delta_c$
20:     end if
21: end while
22: $\texttt{c\_list}_0 \leftarrow [c_{10}, \dots, c_{l_0 0}]$, $\texttt{s\_list}_0 \leftarrow [s_{10}, \dots, s_{l_0 0}]$
23: return $\texttt{c\_list}_0, \dots, \texttt{c\_list}_m, \texttt{s\_list}_0, \dots, \texttt{s\_list}_m$
```

Figure 2.8: `RoundProjectPutinar`

## Comparison with the rounding-projection algorithm of Peyrl and Parrilo

We now state a constrained version of the rounding-projection algorithm from [237].

For $f \in \mathbb{Z}[\mathbf{x}]$ positive over a basic compact semialgebraic set $\mathbf{X}$ satisfying Assumption 2.3.8, Algorithm `RoundProjectPutinar` starts as in Algorithm `Putinarsos`: it outputs the smallest $r$ such that $f \in \mathcal{M}(\mathbf{X})_r$, solves SDP (2.37) in Step 6, and performs Cholesky's factorization in Step 9 to obtain an approximate Putinar's representation of $f$. Note that the approximate Cholesky's factorization is performed to obtain weighted SOS decompositions associated to the constraints $g_1, \dots, g_m$ (i.e. $\tilde{\sigma}_1, \dots, \tilde{\sigma}_m$, respectively).

Next, the algorithm applies in Step 15 the same projection procedure of Algorithm `RoundProject` on the polynomial $u := f - \sum_{j=1}^m \tilde{\sigma}_j g_j$. Note that when there are no constraints, one retrieves exactly the projection procedure from Algorithm `RoundProject`. Exact $\mathbf{LDL}^T$ is then performed on the matrix $\mathbf{G}$ corresponding to $u$.

If all input precision parameters are large enough, $\mathbf{G}$ is a Gram matrix associated to $u$ and $\tilde{\sigma}_1, \dots, \tilde{\sigma}_m$ are weighted SOS polynomals, yielding the exact Putinar's representation $f = u + \sum_{j=1}^m \tilde{\sigma}_j g_j$. As for Theorem 2.3.3 and Theorem 2.3.6, Algorithm `RoundProjectPutinar` has a similar bit complexity than `Putinarsos`.

Table 2.6: `multivsos` vs `univsos2` (Table 2.1) for benchmarks from [61].

| Id | $d$ | $\tau$ (bits) | `multivsos` | | `univsos2` | |
|---|---|---|---|---|---|---|
| | | | $\tau_1$ (bits) | $t_1$ (s) | $\tau_2$ (bits) | $t_2$ (s) |
| # 1 | 13 | 22 682 | 387 178 | 0.84 | 51 992 | 0.83 |
| # 3 | 32 | 269 958 | − | − | 580 335 | 2.64 |
| # 4 | 22 | 47 019 | 1 229 036 | 2.08 | 106 797 | 1.78 |
| # 5 | 34 | 117 307 | 10 271 899 | 69.3 | 265 330 | 5.21 |
| # 6 | 17 | 26 438 | 713 865 | 1.15 | 59 926 | 1.03 |
| # 7 | 43 | 67 399 | 10 360 440 | 16.3 | 152 277 | 11.2 |
| # 8 | 22 | 27 581 | 1 123 152 | 1.95 | 63 630 | 1.86 |
| # 9 | 20 | 30 414 | 896 342 | 1.54 | 68 664 | 1.61 |
| # 10 | 25 | 42 749 | 2 436 703 | 3.02 | 98 926 | 2.76 |

## Practical experiments

We provide experimental results for Algorithms `intsos`, `Reznicksos` and `Putinarsos`. These are implemented in a procedure, called `multivsos`, and integrated in the `RealCertify` library by [C9], written in Maple. We use the Maple `Convex` package[4] to compute Newton polytopes. Our subroutine `sdp` relies on the arbitrary-precision solver SDPA-GMP by [221] and the `cholesky` procedure is implemented with `LUDecomposition` available within Maple. Most of the time is spent in the `sdp` procedure for all benchmarks. To decide nonnegativity of polynomials, we use either `RAGLib` or the `sdp` procedure as oracles. Recall that `RAGLib` relies on critical point methods whose runtime strongly depends on the number of (complex) solutions to polynomial systems encoding critical points. While these methods are more versatile, this number is generically exponential in $n$. Hence, we prefer to rely at first on a *heuristic* strategy based on using `sdp` first (recall that it does not provide an exact answer).

In Table 2.6, we compare the performance of `multivsos` for nine univariate polynomials being positive over compact intervals. More details about these benchmarks are given in [61, Section 6] and [J17, Section 5]. In this case, we use `Putinarsos`. The main difference is that we use SDP in `multivsos` instead of complex root isolation in `univsos2`. The results emphasize that `univsos2` is faster and provides more concise SOS certificates, especially for high degrees (see, e.g., # 5). For # 3, we were not able to obtain a decomposition within a day of computation with `multivsos`, as meant by the symbol − in the corresponding column entries. Large values of $d$ and $\tau$ require more precision. The values of $\varepsilon$, $\delta$ and $\delta_c$ are respectively between $2^{-80}$ and $2^{-240}$, 30 and 100, 200 and 2000.

Next, we compare the performance of `multivsos` with other tools in Table 2.7. The two first benchmarks are built from the polynomial $f = (x_1^2+1)^2 + (x_2^2+1)^2 + 2(x_1+x_2+1)^2 - 268849736/10^8$ from [180, Example 1], with $f_{12} := f^3$ and $f_{20} := f^5$. For these two benchmarks, we apply `intsos`. We use `Reznicksos` to handle $M_{20}$ (resp. $M_{100}$), obtained as in Example 2.3.2 by adding $2^{-20}$ (resp. $2^{-100}$) to the positive coefficients of the Motzkin polynomial and $r_i$, which is a randomly generated positive definite quartic with $i$ variables. We implemented in Maple the projection and rounding algorithm from [237] also relying on SDP, denoted by `RoundProject`. For `multivsos`, the values of $\varepsilon$, $\delta$ and $\delta_c$ lie between $2^{-100}$ and $2^{-10}$, 60 and 200, 10 and 60.

In most cases, `multivsos` is more efficient than `RoundProject` and outputs more concise representations. The reason is that `multivsos` performs approximate Cholesky's decompositions while `RoundProject` computes exact $\mathbf{LDL}^T$ decompositions of Gram matrices obtained after the two steps of rounding and projection. This observation matches with the theoretical complexity esti-

[4] http://www.home.math.uwo.ca/faculty/franz/convex

Table 2.7: `multivsos` vs `RoundProject` [237] vs `RAGLib` vs `CAD` (Reznick).

| Id | $n$ | $d$ | `multivsos` | | `RoundProject` | | `RAGLib` | `CAD` |
|---|---|---|---|---|---|---|---|---|
| | | | $\tau_1$ (bits) | $t_1$ (s) | $\tau_2$ (bits) | $t_2$ (s) | $t_3$ (s) | $t_4$ (s) |
| $f_{12}$ | 2 | 12 | 316 479 | 3.99 | 3 274 148 | 3.87 | 0.15 | 0.07 |
| $f_{20}$ | 2 | 20 | 754 168 | 113. | 53 661 174 | 137. | 0.16 | 0.03 |
| $M_{20}$ | 3 | 8 | 4 397 | 0.14 | 3 996 | 0.16 | 0.13 | 0.05 |
| $M_{100}$ | 3 | 8 | 56 261 | 0.26 | 12 200 | 0.20 | 0.15 | 0.03 |
| $r_2$ | 2 | 4 | 1 680 | 0.11 | 1 031 | 0.12 | 0.09 | 0.01 |
| $r_4$ | 4 | 4 | 13 351 | 0.14 | 47 133 | 0.15 | 0.32 | – |
| $r_6$ | 6 | 4 | 52 446 | 0.24 | 475 359 | 0.37 | 623. | – |
| $r_8$ | 8 | 4 | 145 933 | 0.70 | 2 251 511 | 1.08 | – | – |
| $r_{10}$ | 10 | 4 | 317 906 | 3.38 | 8 374 082 | 4.32 | – | – |
| $r_6^2$ | 6 | 8 | 1 180 699 | 13.4 | 146 103 466 | 112. | 10.9 | – |

mates established in Proposition 2.3.6 and Theorem 2.3.3. Note that we could not solve the examples of Table 2.7 with less precision.

We compare with `RAGLib` [261] based on critical point methods (see, e.g., [258, 141]) and the `SamplePoints` procedure [195] (abbreviated as `CAD`) based on CAD [67], both available in Maple. Observe that `multivsos` can tackle examples which have large degree but a rather small number of variables ($n \leq 3$) and then return certificates of nonnegativity. The runtimes are slower than what can be obtained with `RAGLib` and/or `CAD` (which in this setting have polynomial complexity when $n \leq 3$ is fixed). Note that the bitsize of the certificates which are obtained here is quite large which explains this phenomenon.

When the number of variables increases, `CAD` cannot reach many of the problems we considered. Note that `multivsos` becomes not only faster but can solve problems which are not tractable by `RAGLib`.

Recall that `multivsos` relies first on solving numerically LMI ; this is done at finite precision in time polynomial in the size of the input matrix, which, here is bounded by $\binom{n+d}{d}$. Hence, at fixed degree, that quantity evolves polynomially in $n$. On the other hand, the quantity which governs the behaviour of fast implementations based on the critical point method is the degree of the critical locus of some map. On the examples considered, this degree matches the worst case bound which is the Bézout number $d^n$. Besides, the doubly exponential theoretically proven complexity of CAD is also met on these examples.

These examples illustrate the potential of `multivsos` and more generally SDP-based methods: at fixed degree, one can hope to take advantage of fast numerical algorithms for SDP and tackle examples involving more variables than what could be achieved with more general tools.

Recall however that `multivsos` computes rational certificates of nonnegativity in some "easy" situations: roughly speaking, these are the situations where the input polynomial lies in the interior of the SOS cone and has coefficients of moderate bitsize. This fact is illustrated by Table 2.8.

This table reports on problems appearing enumerative geometry (polynomials $S_1$ and $S_2$ communicated by Sottile and appearing in the proof of the Shapiro conjecture [275]), computational geometry (polynomials $V_1$ and $V_2$ appear in [87]) and in the proof of the monotone permanent conjecture in [114] ($M_1$ to $M_4$).

We were not able to compute certificates of nonnegativity for these problems which we presume do not lie in the interior of the SOS cone. This illustrates the current theoretical limitation of `multivsos`. These problems are too large for `CAD` but `RAGLib` can handle them. Note that some of these examples involve 8 variables ; we observed that the Bézout number is far above the degree

Table 2.8: `multivsos` vs `RAGLib` vs `CAD` for nonnegative polynomials which are presumably not in $\mathring{\Sigma}[X]$.

| Id | $n$ | $d$ | `multivsos` | | `RAGLib` | `CAD` |
|---|---|---|---|---|---|---|
| | | | $\tau_1$ (bits) | $t_1$ (s) | $t_2$ (s) | $t_3$ (s) |
| $S_1$ | 4 | 24 | – | – | 1788. | – |
| $S_2$ | 4 | 24 | – | – | 1840. | – |
| $V_1$ | 6 | 8 | – | – | 5.00 | – |
| $V_2$ | 5 | 18 | – | – | 1180. | – |
| $M_1$ | 8 | 8 | – | – | 351. | – |
| $M_2$ | 8 | 8 | – | – | 82.0 | – |
| $M_3$ | 8 | 8 | – | – | 120. | – |
| $M_4$ | 8 | 8 | – | – | 84.0 | – |

of the critical loci computed by the critical point algorithms in `RAGLib`. This explains the efficiency of such tools on these problems. Recall however that `RAGLib` did not provide a certificate of nonnegativity.

This whole set of examples illustrates first the efficiency and usability of `multivsos` as well as its complementarity with other more general and versatile methods. This shows also the need of further research to handle in a systematic way more general nonnegative polynomials than what it does currently. For instance, we emphasize that certificates of nonnegativity were computed for $M_i$ $(1 \leq i \leq 4)$ in [150] (see also [152]).

Table 2.9: `multivsos` vs `RoundProjectPutinar` vs `RAGLib` vs `CAD` (Putinar).

| Id | $n$ | $d$ | `multivsos` | | | `RoundProject` | | `RAGLib` | `CAD` |
|---|---|---|---|---|---|---|---|---|---|
| | | | $k$ | $\tau_1$ (bits) | $t_1$ (s) | $\tau_2$ (bits) | $t_2$ (s) | $t_3$ (s) | $t_4$ (s) |
| $p_{46}$ | 2 | 4 | 3 | 45 168 | 0.17 | 230 101 | 0.19 | 0.15 | 0.81 |
| $f_{260}$ | 6 | 3 | 2 | 251 411 | 2.35 | 5 070 043 | 3.60 | 0.12 | – |
| $f_{491}$ | 6 | 3 | 2 | 245 392 | 4.63 | 4 949 017 | 5.63 | 0.01 | 0.05 |
| $f_{752}$ | 6 | 2 | 2 | 23 311 | 0.16 | 74 536 | 0.15 | 0.07 | – |
| $f_{859}$ | 6 | 7 | 4 | 13 596 376 | 299. | 2 115 870 194 | 5339. | 5896. | – |
| $f_{863}$ | 4 | 2 | 1 | 12 753 | 0.13 | 30 470 | 0.13 | 0.01 | 0.01 |
| $f_{884}$ | 4 | 4 | 3 | 423 325 | 13.7 | 10 122 450 | 16.1 | 0.21 | – |
| $f_{890}$ | 4 | 4 | 2 | 80 587 | 0.48 | 775 547 | 0.56 | 0.08 | – |
| butcher | 6 | 3 | 2 | 538 184 | 1.36 | 8 963 044 | 3.35 | 47.2 | – |
| heart | 8 | 4 | 2 | 1 316 128 | 3.65 | 35 919 125 | 14.1 | 0.54 | – |
| magnetism | 7 | 2 | 1 | 19 606 | 0.29 | 16 022 | 0.28 | 434. | – |

Finally, we compare the performance of `multivsos` (`Putinarsos`) on positive polynomials over basic compact semialgebraic sets in Table 2.9. The first benchmark is from [180, Problem 4.6]. Each benchmark $f_i$ comes from an inequality of the Flyspeck project [117]. The three last benchmarks are from [56]. The maximal degree of the polynomials involved in each system is denoted by $d$. We emphasize that the degree $2r$ of each Putinar representation obtained in practice with `Putinarsos` is very close to $d$, which is in contrast with the theoretical complexity estimates obtained earlier. The values of $\varepsilon$, $\delta$ and $\delta_c$ lie between $2^{-30}$ and $2^{-10}$, 60 and 200, 10 and 30.
As for Table 2.7, `RAGLib` and `multivsos` can both solve large problems (involving, e.g., 8 variables) but note that `multivsos` outputs certificates of emptiness which cannot be computed with

implementations based on the critical point method. In terms of timings, `multivsos` is sometimes way faster (e.g. magnetism, $f_{859}$) but that it is hard here to draw some general rules. Again, it is important to keep in mind the parameters which influence the runtimes of both techniques. As before, for `multivsos`, the size of the SDP to be solved is clearly the key quantity. Also, it is important to write the systems in an appropriate way also to limit the size of those matrices (e.g. write $1 - x^2 \leq 0$ to model $-1 \leq x \leq 1$). For `RAGLib`, it is way better to write $-1 \leq x$ and $x \leq 1$ to better control the Bézout bounds governing the difficulty of solving systems with purely algebraic methods. Note also that the number of inequalities increase the combinatorial complexity of those techniques.

Finally, note that `CAD` can only solve 3 benchmarks out of 10 and all in all `multivsos` and `RAGLib` solve a similar amount of problems; the latter one however does not provide certificates of emptiness. As for Table 2.7, `multivsos` and `RoundProjectPutinar` yield similar performance, while the former provides more concise output than the latter.

## 2.4 Exact SONC and SAGE certificates

In this section, we provide a hybrid numeric-symbolic framework, in a similar spirit as [237] and the two previous sections. Our motivation is to improve the scalability of existing certification frameworks, especially for large-size problems, which are currently out of reach when relying on SOS-based methods. Most material presented here has been published in [C10]. We focus on certifying exactly nonnegativity of certain classes of polynomials with sparse support, namely *sums of nonnegative circuits* and *arithmetic-geometric-mean-exponentials*. Such polynomials have a support, i.e., number of monomial terms, which is small in comparison to $\binom{n+d}{n}$, namely the maximal support size of fully dense $n$-variate polynomials of degree $d$. Alternative relaxations based on geometric programming (GP) [247], and, more generally, relative entropy programming (REP), potentially allow to obtain lower bounds in a more efficient way than SDP relaxations. Both GP and REP are (equivalent to) convex optimization problems over the exponential cone. However, one often encounters numerical issues, even for problems of modest size where the SDP relaxations can be implemented. These alternative relaxations also provide the possibility to obtain answers when the SDP relaxations cannot be implemented because their size is too large for state-of-the art SDP solvers.

### Alternative nonnegativity certificates

A first class of alternative certificates is given by SONC polynomials. A *circuit polynomial* is a polynomial with support containing only monomial squares, at the exception of at most one term, whose exponent is a strict convex combination of the other exponents. In [256], the authors derive a necessary and sufficient condition to prove that a given circuit polynomial is nonnegative. When the input polynomial has a more general support, a first attempt is given in [205, 204] to compute lower bounds while relying on GP. This approach is generalized in [203] to compute SONC certificates when the set of constraints is defined as a finite conjunction of polynomial inequalities. In [202] the authors provide a bounded degree hierarchy, which can be computed via REP. In [264], the authors develop an algorithm computing SONC certificates for sparse unconstrained polynomials with arbitrary support, together with a software library [265], called POEM *(Effective Methods in Polynomial Optimization)*. Although this framework yields a very efficient way to obtain a lower bound for a given polynomial, a drawback is that it currently remains unclear whether the number of circuits involved in a SONC relaxation is exponential in the number of terms of this polynomial. A second class of alternative certificates is given by SAGE polynomials. An *AGE polynomial* refers to a *signomial*, i.e., a weighted sum of exponentials composed with linear functionals of the variables, which is globally nonnegative with at most one negative coefficient. The framework from [290] derives a hierarchy of convex relaxations providing a sequence of increasing lower bounds for the optimal value of signomial programs. For an input polynomial belonging to the SAGE cone, one can compute a SAGE decomposition by solving an REP, involving linear and relative entropy functions. Furthermore, it is shown in [220, Theorem 20] that the cones of SAGE and SONC polynomials are related through their equivalence in terms of extreme rays. Namely, the extreme rays of the SAGE cone are supported on either a single coordinate or a set of coordinates inducing a simplicial circuit (a circuit with $t$ elements containing $t-1$ extreme points). Hence, both cones contain the same polynomials.

However, these alternative schemes share the same certification issues than the ones based on SDP relaxations. GP/REP solvers rely on interior-point algorithms, implemented in finite-precision. Thus, they output only approximate certificates.

## Sparse polynomials

We mostly regard *sparse polynomials* $f \in \mathbb{R}[\mathbf{x}]$ supported on a finite set $A \subset \mathbb{N}^n$; we write $\operatorname{supp}(f)$ if a clarification is necessary. Thus, $f$ is of the form $f(\mathbf{x}) = \sum_{\alpha \in A} b_\alpha \mathbf{x}^\alpha$ with $b_\alpha \in \mathbb{R} \setminus \{0\}$ and $\mathbf{x}^\alpha = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$. Unless stated differently, we follow the convention $t = \#A$. Sparsity then means $t \ll \binom{n+2d}{2d} = \dim(\mathbb{R}[\mathbf{x}]_{2d})$. The support of $f$ can be expressed as an $n \times t$ matrix, which we denote by $\mathbf{A}$, such that the $j$-th column of $\mathbf{A}$ is $\alpha(j)$. Hence, $f$ is uniquely described by the pair $(\mathbf{A}, \mathbf{b})$, written $f = \operatorname{poly}(\mathbf{A}, \mathbf{b})$. If $\mathbf{0} \in \mathbf{A}$, then $f(0)$ is called the *constant term*. Let $\mathcal{C}(f) := \operatorname{conv}(\{\alpha \in \mathbb{N}^n : b_\alpha \neq 0\})$ be the *Newton polytope* of $f$ and $\operatorname{Vert}(f)$ be its set of vertices. We define $\operatorname{MoSq}(f) := \{\alpha \in \operatorname{supp}(f) : \alpha \in (2\mathbb{N})^n, b_\alpha > 0\}$ as the set of monomial squares in the support of $f$. Moreover, we use the notation $\operatorname{NoSq}(f) = \operatorname{supp}(f) \setminus \operatorname{MoSq}(f)$ for all elements of the support of $f$, which are not monomial squares.

## SONC polynomials

We now introduce the fundamental facts of SONC polynomials, which we use in this article. SONC are constructed by *circuit polynomials*, which were first introduced in [256]:

**Definition 2.4.1** *A* circuit polynomial $f = \operatorname{poly}(\mathbf{A}, \mathbf{b}) \in \mathbb{R}[\mathbf{x}]$ *is of the form* $f(\mathbf{x}) = \sum_{j=1}^{l} b_{\alpha(j)} \mathbf{x}^{\alpha(j)} + b_\beta \mathbf{x}^\beta$, *with* $0 \leq l < n$, *coefficients* $b_{\alpha(j)} \in \mathbb{R}_{>0}$, $b_\beta \in \mathbb{R}$, *exponents* $\alpha(j) \in (2\mathbb{Z})^n$, $\beta \in \mathbb{Z}^n$, *such that the following condition holds: there exist unique, positive* barycentric coordinates $\lambda_j$ *relative to the* $\alpha(j)$ *with* $j \in [l]$ *satisfying*

$$\beta = \sum_{j=1}^{l} \lambda_j \alpha(j) \ \ \text{with} \ \ \lambda_j > 0 \ \ \text{and} \ \ \sum_{j=1}^{l} \lambda_j = 1. \tag{2.38}$$

*For every circuit polynomial $f$ we define the corresponding circuit number as* $\Theta_f = \prod_{j=1}^{l} \left( b_{\alpha(j)} / \lambda_j \right)^{\lambda_j}$.

Condition (2.38) implies that $\mathbf{A}(f)$ forms a minimal affine dependent set. Those sets are called *circuits*, see e.g., [145]. More specifically, Condition (2.38) yields that $\mathcal{C}(f)$ is a simplex with even vertices $\alpha(1), \ldots, \alpha(l)$ and that the exponent $\beta$ is in the strict interior of $\mathcal{C}(f)$ if $\dim(\mathcal{C}(f)) \geq 1$. Therefore, we call $b_\beta \mathbf{x}^\beta$ the *inner term* of $f$.

Circuit polynomials are proper building blocks for nonnegativity certificates since the circuit number alone determines whether they are nonnegative.

**Theorem 2.4.2 ([256], Theorem 3.8)** *Let $f$ be a circuit polynomial as in Definition 2.4.1. Then $f$ is nonnegative if and only if:*

1. *$f$ is a sum of monomial squares, or*
2. *the coefficient $b_\beta$ of the inner term of $f$ satisfies $|b_\beta| \leq \Theta_f$.*

The set of *sums of nonnegative circuit polynomials* (SONC) is a convex cone. For further details about SONC see [282, 256, 202].

Let us consider $f = \operatorname{poly}(\mathbf{A}, \mathbf{b}) = \sum_{j=1}^{t} b_j \mathbf{x}^{\alpha(j)}$ and assume that $\mathcal{C}(f)$ is a simplex of dimension $h \leq n$. To compute a lower bound of $f$ via SONC, we first compute a *covering*, which is a sequence of sets $A_1, \dots, A_{t-h} \subseteq A$ such that $\operatorname{NoSq}(f) \subseteq \bigcup_j A_j$ and each $A_j$ is the support of a nonnegative circuit polynomial $f_j$. To obtain a covering, we write each non-square as a minimal convex combination of monomial squares, by solving a sequence of LP. For more details see [264, § 3.1]. Then, we solve the following GP, stated in [264, § 3.2]:

$$\begin{aligned} f_{\mathrm{SONC}} = \min_{\mathbf{G}} \quad & \sum_{j=1}^{t-h} \mathbf{G}_{0,j} \\ \text{s.t.} \quad & \sum_{j=1}^{t-h} \mathbf{G}_{i,j} \leq b_i \,, \quad i = 2, \dots, h \,, \\ & \prod_{i=1}^{h} \left( \frac{\mathbf{G}_{i,j}}{\lambda_{i,j}} \right)^{\lambda_{i,j}} = -b_{h+j} \,, \quad j \in [t-h] \,, \\ & \mathbf{G}_{i,j} \geq 0 \,, \quad i, j \in [t-h] \,. \end{aligned} \tag{SONC}$$

For an overview of GP, see [254, 253]. If $f_{\mathrm{SONC}}$ is attained at $\mathbf{G}$, then one has $f_j = \sum_{i=1}^{h} \mathbf{G}_{i,j} \cdot \mathbf{x}^{\alpha(i)} + b_{h+j} \mathbf{x}^{\alpha(h+j)} \geq 0$ by Theorem 2.4.2, and $f + f_{\mathrm{SONC}} - b_1 = \sum_{j=1}^{t-h} f_j \geq 0$. Hence, $b_1 - f_{\mathrm{SONC}}$ is a lower bound of $f$ on $\mathbb{R}^n$.

## SAGE polynomials

Let $e := \exp(1)$. The *relative entropy* function is defined for $\boldsymbol{\nu}, \mathbf{c} \in \mathbb{R}_+^t$ by $D(\boldsymbol{\nu}, \mathbf{c}) := \sum_{j=1}^{t} \nu_j \log \frac{\nu_j}{c_j}$. A *signomial* $f$ is a weighted sum of exponentials composed with linear functionals of a variable $\mathbf{x} \in \mathbb{R}^n$: given $t \in \mathbb{N}$, $c_1, \dots, c_t \in \mathbb{Q}$ and $\alpha(1), \dots, \alpha(t) \in \mathbb{N}^n$, we write $f(\mathbf{x}) = \sum_{j=1}^{t} c_j \exp(\alpha(j) \cdot \mathbf{x})$. Note that for general signomials, one considers $c_1, \dots, c_t \in \mathbb{R}$ and $\alpha(1), \dots, \alpha(t) \in \mathbb{R}^n$. However, for certification purpose, we restrict the coefficients to the set of rationals and the exponents to tuples of nonnegative integers. A globally nonnegative signomial with at most one negative coefficient is called an *arithmetic-geometric-mean-exponential (AGE)*. Certifying the nonnegativity of an AGE is done by verifying an arithmetic-geometric-mean inequality. This is recalled in the following result, stated in [290, Lemma 2.2].

**Lemma 2.4.3** *Let $f(\mathbf{x}) = \sum_{j=1}^{t} c_j \exp(\alpha(j) \cdot \mathbf{x}) + \beta \exp(\alpha(0) \cdot \mathbf{x})$, with $c_1, \dots, c_t \in \mathbb{Q}_{>0}$, $\beta \in \mathbb{Q}$ and $\alpha(0), \alpha(1), \dots, \alpha(t) \in \mathbb{N}^n$. Then $f(\mathbf{x}) \geq 0$ for all $\mathbf{x} \in \mathbb{R}^n$ if and only if there exists $\boldsymbol{\nu} \in \mathbb{R}_+^t$ such that $D(\boldsymbol{\nu}, e\mathbf{c}) \leq \beta$ and $\sum_{j=1}^{t} \alpha(j)\nu_j = (\mathbf{1} \cdot \boldsymbol{\nu})\, \alpha(0)$.*

Given $\alpha(0), \alpha(1), \dots, \alpha(t) \in \mathbb{N}^n$, the set of AGE signomials is a convex cone, denoted by CAGE, and defined as follows:

$$\begin{aligned} \mathrm{CAGE} := \Big\{ (\mathbf{c}, \beta) \in \mathbb{R}_+^t \times \mathbb{R} : \exists \boldsymbol{\nu} \in \mathbb{R}_+^t \text{ s.t. } D(\boldsymbol{\nu}, e\mathbf{c}) \leq \beta \,, \\ \sum_{j=1}^{t} \alpha(j)\nu_j = (\mathbf{1} \cdot \boldsymbol{\nu})\, \alpha(0) \Big\} \,. \end{aligned}$$

Given a vector $\boldsymbol{\nu} \in \mathbb{R}^n$, let us denote by $\boldsymbol{\nu}_{\backslash j} \in \mathbb{R}^{n-1}$ the vector obtained from $\boldsymbol{\nu}$ after removing $\nu_j$. The set of *sums of AGE (SAGE)* polynomials is also a convex cone, denoted by SAGE. By [290, Proposition 2.4], one has the following characterisation.

**Theorem 2.4.4** *A signomial $f = \sum_{i=1}^{t} b_j \exp\left(\alpha(j) \cdot \mathbf{x}\right)$ lies in SAGE if and only if there is $\mathbf{c}^{(1)}, \dots, \mathbf{c}^{(t)}$, $\boldsymbol{\nu}^{(1)}, \dots, \boldsymbol{\nu}^{(t)} \in \mathbb{R}^t$ satisfying the following conditions:*

$$\begin{gathered}\sum_{j=1}^{t} \mathbf{c}^{(j)} = \mathbf{b}\,, \quad \sum_{i=1}^{t} \alpha(i) \boldsymbol{\nu}_i^{(j)} = \mathbf{0}\,, \quad -\mathbf{1} \cdot \boldsymbol{\nu}_{\backslash j}^{(j)} = \nu_j^{(j)}\,, \\ \mathbf{c}_{\backslash j}^{(j)}, \boldsymbol{\nu}_{\backslash j}^{(j)} \geq \mathbf{0}\,, \quad D\left(\boldsymbol{\nu}_{\backslash j}^{(j)}, e\mathbf{c}_{\backslash j}^{(j)}\right) \leq c_j^{(j)}\,, \quad j \in [t]\,. \end{gathered} \tag{SAGE-feas}$$

One way to obtain lower bounds of a signomial $f$ is to solve the following REP:

$$f_{\mathrm{SAGE}} = \sup_{b \in \mathbb{R}} \{ b : f - b \in \mathrm{SAGE} \}\,. \tag{SAGE}$$

The constraints of (SAGE) correspond to (SAGE-feas), after replacing $\mathbf{b}$ by the vector of coefficients of $f - b$.

In the sequel , we present two algorithms for converting a numerical solution for SONC and SAGE into a lower bound in exact arithmetic. For a polynomial $f = \sum_{j=1}^{t} b_j \mathbf{x}^{\alpha(j)}$, we assume $\alpha(1) = \mathbf{0}$, so the first term $b_1$ of $f$ is the constant term $f(0)$. Furthermore, we require that every non-square monomial lies in the interior of $\mathcal{C}(f)$ or on a face of $\mathcal{C}(f)$ including the origin.

## Symbolic post-processing for SONC

We focus on certifying exactly lower bounds of a given polynomial via SONC decompositions. We rely on the numerical procedure from [264], which starts to compute a covering of the Newton polytope of this polynomial. We strengthen the last assumption further and assume that every circuit polynomial related to this covering contains the point $\mathbf{0}$, which means that every circuit includes a non-zero constant term. Under these assumptions, we design an algorithm, called `optsonc`, to convert numerical lower bounds, corresponding to SONC decompositions obtained via GP, into exact lower bounds.

The procedure to compute a covering, yielding the entries of the matrix $\boldsymbol{\lambda}$ in Step 1, is the one from [264, § 3.1]. In Step 2, The `gp` function calls a GP solver to compute a $\tilde{\delta}$-approximation $\tilde{\mathbf{G}}$ of (SONC). This approximation is then rounded in Step 3 to a rational point $\hat{\mathbf{G}}$ with a prescribed maximal relative error of $\hat{\delta}$. The projection step from Step 5 scales the entries of $\hat{\mathbf{G}}$, yielding $\sum_{j=1}^{|\operatorname{Cov}|} \mathbf{G}_{i,j} = b_i$, for all $i = 2, \dots, h$, to satisfy the first set of inequality constraints of (SONC). In Step 9, one relies on the `round-up` procedure in such a way that $\mathbf{G}_{1,j} \geq \lambda_{1,j} \cdot \left( b_{\mathrm{NoSq}(f)(j)} \cdot \prod_{\substack{i \in \operatorname{Cov}_j \\ i > 1}} \left( \frac{\lambda_{i,j}}{\mathbf{G}_{i,j}} \right)^{\lambda_{i,j}} \right)^{\frac{1}{\lambda_{1,j}}}$.

By Theorem 2.4.2, this yields a valid lower bound $b_1 - \sum_{j=1}^{|\operatorname{Cov}|} \mathbf{G}_{1,j}$ for $f$. Our assumption that every circuit polynomial contains a constant term, is necessary to ensure, that all $\lambda_{1,j} \neq 0$ in our above computations.

## Symbolic post-Processing for SAGE

Similarly to Figure 2.9, our algorithm `optsage` takes a given polynomial as input, obtains a numerical lower bound related to a SAGE decomposition computed via REP, and applies a post-processing to find a certified lower bound. Given a polynomial $g(\mathbf{y}) = \sum_{j=1}^{t} b_j \mathbf{y}^{\alpha(j)}$, one could

```
Require: $f = \sum_{i=1}^{t} b_i \mathbf{x}^{\alpha(i)} \in \mathbb{Q}[\mathbf{x}]$, rounding precision $\hat{\delta} \in \mathbb{Q}_{>0}$, precision parameter $\tilde{\delta} \in \mathbb{Q}_{>0}$ for
    the GP solver.
Ensure: Matrix $\mathbf{G}$ of rational numbers, coefficients of the decomposition, certified lower bound
    $b \in \mathbb{Q}$ of $f$ on $\mathbb{R}^n$.
 1: $\boldsymbol{\lambda}, \mathrm{Cov} \leftarrow \texttt{cover}(f)$
 2: $\tilde{\mathbf{G}} \leftarrow \texttt{gp}(f, \tilde{\delta}, \boldsymbol{\lambda}, \mathrm{Cov})$
 3: $\hat{\mathbf{G}} \leftarrow \texttt{round}\left(\tilde{\mathbf{G}}, \hat{\delta}\right)$                              ▷ rounding step
 4: for $i \in \{2, \ldots, t\}$ and $j \in \{1, \ldots, |\mathrm{Cov}|\}$ do
 5:     $\mathbf{G}_{i,j} \leftarrow b_i \cdot \hat{\mathbf{G}}_{i,j} / \sum_{k=1}^{|\mathrm{Cov}|} \hat{\mathbf{G}}_{i,k}$                 ▷ projection step
 6: end for
 7: for $j \in \{1, \ldots, |\mathrm{Cov}|\}$ do
 8:     $\mathrm{coeff} = \lambda_{1,j} \cdot \left( b_{\mathrm{NoSq}(f)(j)} \cdot \prod_{\substack{i \in \mathrm{Cov}_j \\ i>1}} \left( \frac{\lambda_{i,j}}{\mathbf{G}_{i,j}} \right)^{\lambda_{i,j}} \right)^{\frac{1}{\lambda_{1,j}}}$
 9:     $\mathbf{G}_{1,j} \leftarrow \texttt{round-up}\left(\mathrm{coeff}, \hat{\delta}\right)$                    ▷ adjust constant term
10: end for
11: $b \leftarrow b_1 - \sum_{j=1}^{|\mathrm{Cov}|} X_{1,j}$
12: return $\mathbf{G}, b$
```

Figure 2.9: `optsonc`

```
Require: $g = \sum_{i=1}^{t} b_i \mathbf{x}^{\alpha(i)} \in \mathbb{Q}[\mathbf{x}]$, rounding precision $\hat{\delta} \in \mathbb{Q}_{>0}$, precision parameter $\tilde{\delta} \in \mathbb{Q}_{>0}$ for
    the REP solver.
Ensure: Matrices $\mathbf{c}, \boldsymbol{\nu}$ of rational numbers, coefficients of the decomposition, certified lower bound
    $C \in \mathbb{Q}$ of $g$ on $\mathbb{R}^n$.
 1: $f \leftarrow g(\exp \mathbf{x} - \exp(-\mathbf{x}))$
 2: Build the $(n+1) \times t$ matrix $\mathbf{G}$ with columns $(\alpha(1), 1), \ldots, (\alpha(t), 1)$
 3: $\tilde{\mathbf{c}}, \tilde{\boldsymbol{\nu}} \leftarrow \texttt{rep}(f, \tilde{\delta})$
 4: $\hat{\mathbf{c}} \leftarrow \texttt{round}\left(\tilde{\mathbf{c}}, \hat{\delta}\right)$ ,   $\hat{\boldsymbol{\nu}} \leftarrow \texttt{round}\left(\tilde{\boldsymbol{\nu}}, \hat{\delta}\right)$                  ▷ rounding step
 5: for $j \in [t]$ do
 6:     $LP \leftarrow \left\{ \mathbf{G} \cdot \boldsymbol{\nu}^{(j)} = \mathbf{0}, \boldsymbol{\nu}_{\backslash j}^{(j)} \geq \mathbf{0}, \|\boldsymbol{\nu}^{(j)} - \tilde{\boldsymbol{\nu}}^{(j)}\|_\infty \leq \hat{\delta}, \nu_1^{(j)} \geq \hat{\delta} \right\}$
 7:     $\boldsymbol{\nu}^{(j)} \leftarrow$ some element from LP                                   ▷ projection step
 8:     $\mathbf{c}_{\backslash j}^{(j)} \leftarrow \hat{\mathbf{c}}_{\backslash j}^{(j)}$   $, c_j^{(j)} \leftarrow b_j - \mathbf{1} \cdot \mathbf{c}_{\backslash j}^{(j)}$
 9: end for
10: for $j \in [t]$ do
11:     $\mathrm{power} \leftarrow 1 - \log \nu_1^{(j)} - \frac{1}{\nu_1^{(j)}} \left( c_j^{(j)} - \sum_{i>1, i \neq j} \nu_i^{(j)} \log \frac{\nu_i^{(j)}}{e c_i^{(j)}} \right)$
12:     $c_1^{(j)} \leftarrow \texttt{round-up}\left(\exp\left(\mathrm{power}\right), \hat{\delta}\right)$                ▷ adjust constant term
13: end for
14: $b \leftarrow b_1 - \sum_{j=1}^{t} c_1^{(j)}$
15: return $\mathbf{c}, \boldsymbol{\nu}, b$
```

Figure 2.10: `optsage`

apply the change of variables $y_i := \exp x_i$ when $\mathbf{y} \in \mathbb{R}_{>0}^n$. Since this transformation is only valid on the nonnegative orthant, one workaround used in `optsage` is to define the signomial $f(\mathbf{x}) = g(\exp \mathbf{x} - \exp(-\mathbf{x}))$ from Step 1, in a such a way that a lower bound of $f$ yields a lower bound of $g$. The `rep` function in Step 3 calls an REP solver to compute a $\tilde{\delta}$-approximation $(\tilde{\boldsymbol{\nu}}, \tilde{\mathbf{c}})$

of (SAGE). This approximation is then rounded to a rational point $(\hat{\boldsymbol{\nu}}, \hat{\mathbf{c}})$ with a prescribed maximal relative error of $\hat{\delta}$. Let us build the $(n+1) \times t$ matrix $\mathbf{G}$ with columns $(\alpha(1), 1), \dots, (\alpha(t), 1)$. The projection steps in Step 7 and Step 8 ensure that $(\boldsymbol{\nu}, \mathbf{c})$ satisfies exactly the linear equality constraints of (SAGE), i.e., $\mathbf{G}\boldsymbol{\nu}^{(j)} = \mathbf{0}$ and $\sum_{j=1}^{t} \mathbf{c}^{(j)} = \mathbf{b}$. The first projection step boils down to exactly solve an LP with the constraint that $\nu_1^{(j)} > 0$, for all $j \in [t]$, to ensure that further computation in Step 12 are well-defined. Note that this projection could be done while relying on the pseudo-inverse of $\mathbf{G}$, but one obtains better practical results via this procedure. To ensure that the relative entropy inequality constraints of (SAGE) are satisfied, the last step of `optsage` aims at finding $c_j^{(1)}$ such that $c_j^{(j)} \geq D\left(\boldsymbol{\nu}_{\setminus j}^{(j)}, e\mathbf{c}_{\setminus j}^{(j)}\right) = \sum_{i>1, i\neq j} \nu_i^{(j)} \log \frac{\nu_i^{(j)}}{ec_i^{(j)}} + \nu_1^{(j)} \log \frac{\nu_1^{(j)}}{ec_1^{(j)}}$. Thus, one relies on the `round-up` procedure in Step 12 to compute $c_1^{(j)} \geq \exp\left(1 - \log \nu_1^{(j)} - \frac{1}{\nu_1^{(j)}}\left(c_j^{(j)} - \sum_{i>1, i\neq j} \nu_i^{(j)} \log \frac{\nu_i^{(j)}}{ec_i^{(j)}}\right)\right)$. Eventually, one has $\sum_{j=1}^{t} c_i^{(j)} = b_i$, for all $i > 1$ and $\sum_{j=1}^{t} c_1^{(j)} = b_1 - b$, which certifies that $f - b \geq 0$ on $\mathbb{R}^n$.

## Deciding Nonnegativity via SAGE

We denote by INTSAGE the interior of the cone SAGE of SAGE signomials. A signomial $f = \sum_{j=1}^{t} b_j \exp\left(\alpha(j) \cdot \mathbf{x}\right)$ lies in INTSAGE if and only there is $\mathbf{c}^{(1)}, \dots, \mathbf{c}^{(t)}, \boldsymbol{\nu}^{(1)}, \dots, \boldsymbol{\nu}^{(t)} \in \mathbb{R}^t$ such that

$$\begin{gathered} \sum_{j=1}^{t} \mathbf{c}^{(j)} = \mathbf{b}\,, \quad \sum_{i=1}^{t} \alpha(i)\boldsymbol{\nu}_i^{(j)} = \mathbf{0}\,, \quad -\mathbf{1} \cdot \boldsymbol{\nu}_{\setminus j}^{(j)} = \nu_j^{(j)}\,, \\ \mathbf{c}_{\setminus j}^{(j)}, \boldsymbol{\nu}_{\setminus j}^{(j)} > \mathbf{0}\,, \quad D\left(\boldsymbol{\nu}_{\setminus j}^{(j)}, e\mathbf{c}_{\setminus j}^{(j)}\right) < c_j^{(j)}\,, \quad j \in [t]\,. \end{gathered} \tag{INTSAGE-feas}$$

Without the assumptions from the previous subsection, we state and analyze a decision algorithm to certify nonnegativity of signomials belonging to the interior INTSAGE of the SAGE cone. The resulting hybrid numeric-symbolic algorithm, called `intsage`, computes exact rational SAGE decompositions of such signomials. We start with the preliminary result:

**Lemma 2.4.5** *Let $f = \sum_{j=1}^{t} b_j \exp\left(\alpha(j) \cdot \mathbf{x}\right) \in$ INTSAGE of degree $d$ with $\tau = \tau(f)$. Then, there exists $N \in \mathbb{N}$ such that for $\varepsilon := 2^{-N}$, $f - \varepsilon \sum_{j=1}^{t} \exp\left(\alpha(j) \cdot \mathbf{x}\right) \in$ SAGE, with $N \leq \tau(\varepsilon) \in \mathcal{O}\left(\tau \cdot (4d+6)^{3n+3}\right)$.*

We present our algorithm `intsage` computing exact rational SAGE decompositions for signomials in INTSAGE.

**Algorithm 2.4.6** *`intsage`*

```
Require: $f = \sum_{j=1}^{t} b_j \exp(\alpha(j) \cdot \mathbf{x}) \in$ INTSAGE, rounding precision $\hat{\delta} \in \mathbb{Q}_{>0}$, precision parameter
    $\tilde{\delta} \in \mathbb{Q}_{>0}$ for the REP solver.
Ensure: Matrices $\mathbf{c}, \boldsymbol{\nu}$ of rational numbers.
 1: Build the $(n+1) \times t$ matrix $\mathbf{G}$ with columns $(\alpha(1), 1), \dots, (\alpha(t), 1)$
 2: $\mathbf{G}^{+} \leftarrow$ pseudoinv$(\mathbf{G})$
 3: ok ← false
 4: while not ok do
 5:     $(\tilde{\mathbf{c}}, \tilde{\boldsymbol{\nu}}) \leftarrow$ rep$(f, \tilde{\delta})$
 6:     $\hat{\mathbf{c}} \leftarrow$ round$(\tilde{\mathbf{c}}, \hat{\delta})$ ,   $\hat{\boldsymbol{\nu}} \leftarrow$ round$(\tilde{\boldsymbol{\nu}}, \hat{\delta})$        ▷ rounding step
 7:     for $j \in [t]$ do                                                  ▷ projection step
 8:         $\boldsymbol{\nu}^{(j)} \leftarrow (\mathrm{I} - \mathbf{G}^{+}\mathbf{G})\, \hat{\boldsymbol{\nu}}^{(j)}$
```

```
9:         $\mathbf{c}_{\setminus j}^{(j)} \leftarrow \hat{\mathbf{c}}_{\setminus j}^{(j)}, \quad c_j^{(j)} \leftarrow b_j - \mathbf{1} \cdot \mathbf{c}_{\setminus j}^{(j)}$
10:    end for
11:    if for all $j \in [t], \boldsymbol{\nu}_{\setminus j}^{(j)}, \mathbf{c}_{\setminus j}^{(j)} \geq \mathbf{0}, c_j^{(j)} \geq D\left(\boldsymbol{\nu}_{\setminus j}^{(j)}, e\mathbf{c}_{\setminus j}^{(j)}\right)$, then ok $\leftarrow$ true        ▷ verification step
12:    else $\tilde{\delta} \leftarrow \tilde{\delta}/2, \hat{\delta} \leftarrow \hat{\delta}/2$
13:    end if
14: end while
15: return $\mathbf{c}, \boldsymbol{\nu}$
```

The routine `pseudoinv` in Step 2 computes the *pseudo-inverse* of $\mathbf{G}$, i.e., a matrix $\mathbf{G}^+$ such that $\mathbf{G}\mathbf{G}^+\mathbf{G} = \mathbf{G}$. Next, we enter in the loop starting from Step 4. The `rep` function calls an REP solver to compute a $\tilde{\delta}$-approximation $(\tilde{\boldsymbol{\nu}}, \tilde{\mathbf{c}})$ of (INTSAGE-feas). The projection steps ensure that $(\boldsymbol{\nu}, \mathbf{c})$ satisfies exactly the linear equality constraints of (SAGE-feas), i.e., $\mathbf{G}\boldsymbol{\nu}^{(j)} = \mathbf{G}(\mathrm{I} - \mathbf{G}^+\mathbf{G})\boldsymbol{\nu}^{(j)} = \mathbf{G} - \mathbf{G}\mathbf{G}^+\mathbf{G} = \mathbf{0}$ and $\sum_{j=1}^t \mathbf{c}^{(j)} = b$. If the inequality constraints are not verified in Step 11, the rounding-projection procedure is performed again with more accuracy.

Before analyzing the arithmetic complexity of `intsage`, we first establish lower bounds for the nonnegative components of the solutions related to SAGE decompositions of polynomials in INTSAGE.

**Lemma 2.4.7** *Let $f = \sum_{j=1}^t b_j \exp(\alpha(j) \cdot \mathbf{x}) \in$ INTSAGE of degree $d$ with $\tau = \tau(f)$. Let $\varepsilon$ be as in Lemma 2.4.5.*

1. *There exists a solution of $(\boldsymbol{\nu}, \mathbf{c})$ of (INTSAGE-feas) and $\delta \in \mathbb{Q}_{>0}$ such that $\delta \leq 1$, $(\boldsymbol{\nu}, \mathbf{c})$ satisfies, $D\left(\boldsymbol{\nu}_{\setminus j}^{(j)}, e\left(\mathbf{c}_{\setminus j}^{(j)} + \delta\mathbf{1}\right)\right) + \frac{\varepsilon}{2} \leq c_j^{(j)}$, $\sum_{j=1}^t \mathbf{c}^{(j)} = \mathbf{b}$, for all $i, j \in [t]$.*

2. *There exists a solution $(\boldsymbol{\nu}, \mathbf{c})$ of (INTSAGE-feas) and $\delta \in \mathbb{Q}_{>0}$ such that $(\boldsymbol{\nu}, \mathbf{c})$ satisfies $D\left(\boldsymbol{\nu}_{\setminus j}^{(j)} + \delta\mathbf{1}, e\mathbf{c}_{\setminus j}^{(j)}\right) + \frac{\varepsilon}{2} \leq c_j^{(j)}$, for all $j \in [t]$.*

3. *There exists a solution $(\boldsymbol{\nu}, \mathbf{c})$ of (INTSAGE-feas) and $\delta \in \mathbb{Q}_{>0}$ such that $(\boldsymbol{\nu}, \mathbf{c})$ satisfies $D\left((1+\delta)\boldsymbol{\nu}_{\setminus j}^{(j)}, e\mathbf{c}_{\setminus j}^{(j)}\right) + \frac{\varepsilon}{2} \leq c_j^{(j)}$, for all $j \in [t]$.*

*In each case, $\tau(\delta) \in \mathcal{O}\left(\tau \cdot (4d+6)^{3n+3}\right)$.*

**Theorem 2.4.1** *Let $f$ as in Lemma 2.4.7. There exist $\hat{\delta}$ and $\tilde{\delta}$ of bit size less than $\mathcal{O}\left(\tau \cdot (4d+6)^{3n+3}\right)$, such that `intsos`$(f, \hat{\delta}, \tilde{\delta})$ terminates and outputs a rational SAGE decomposition of $f$ within $\mathcal{O}\left(\tau \cdot (4d+6)^{3n+3}\, t^7 \log t\right)$ arithmetic operations.*

## Experimental comparisons

We discuss the actual bit sizes and physical running time of `optsonc` and `optsage` procedures, given by Figure 2.9 and Figure 2.10. We describe the setup of our experiment and explain how our random instances were created. Afterwards, we discuss a few selected examples, which exhibit well the differences of the methods, and present how the program behaved on a large set of examples. The entire experiment was steered by the PYTHON 3.7 based software POEM 0.2.0.0(a) [265], POEM is open source, under GNU public license, and available online [5]. For our experiment, POEM calls a range of further software and solvers for computing the certificates. For the

[5] https://www3.math.tu-berlin.de/combi/RAAGConOpt/poem.html

$d = 8, t = 20$

| $n$ | bit size | time |
|---|---|---|
| 2 | 27723 | 5.82 |
| 3 | 23572 | 4.99 |
| 4 | 22965 | 4.83 |
| 8 | 5678 | 1.12 |
| 10 | 1749 | 0.35 |

$d = 10, t = 30$

| $n$ | bit size | time |
|---|---|---|
| 2 | 61198 | 12.41 |
| 3 | 57833 | 11.81 |
| 4 | 53596 | 10.82 |
| 8 | 13343 | 2.55 |
| 10 | 6974 | 1.41 |

Table 2.10: Dependency of the average bit size and the average running time of `optsage`, with the number of variables, for fixed values of degree $d$ and number of terms $t$; For $n = 8$ we observe a drastic drop both in running time and bit size.

numerical solutions of SONC and SAGE, we use CVXPY 1.0.12 [79], to create the convex optimization problems, together with the solver ECOS 2.0.7 [82]. The symbolic computations were done in SymPy 1.3 [149].
The experiment was carried out on a database containing 2020 randomly generated polynomials. The possible numbers of variables are $n = 2, 3, 4, 8, 10$; the degree takes values $d = 6, 8, 10, 18, 20, 26, 28$ and the number of terms can be $t = 6, 9, 12, 20, 24, 30, 50$. For each combinations we create instances, where the number of negative terms is one of a few fixed ratios of $t$. In particular, the size of (SAGE) grows quadratically in $t$. We created the database using POEM, and it is available in full at the homepage cited above. Our instances are exactly those from [264] where the parameters are as above. The overall running time for all our instances was 6780.0 seconds.
For the accuracy of the solver and the precision of the rounding in Python we used a tolerance of $\varepsilon = 2^{-23}$. The restriction $t \leq 50$ was chosen, since otherwise we already encounter problem in the numerical solution of (SAGE). The bound $d < 30$ was chosen, because for large degree we had a significant increase in the memory required to perform the rounding. Both thresholds were obtained experimentally.

In this section we present and evaluate the results of our experiment and highlight our most important findings, when investigating the computational data. We focus on the results given by the procedure `optsage` via SAGE decompositions and in the end give a comparison to `optsonc`.
**Running time *decreases* with growing number of variables.** The formulation of (SAGE) shows that the size of the problem only depends on the number of terms $t$, but in the SAGE decomposition, the number of summands is the number of monomial non-squares. Most significantly, for more variables, our generating algorithm simply results in a smaller number of these terms. Additionally, for $n \geq 8$ and $d \leq 10$, most exponents lie on faces of the Newton polytope. This leads to a simpler combinatorial structure, which we believe to result in lower bit sizes and thus in faster solving faster the exact LP from Step 6 of `optsage`. Next, we have more equality constraints in this LP, which could also improve the running time. Lastly, the exponential upper bound is just the worst case, which does not seem to actually happen among our examples. For some selected parameters, we exhibit that behavior in Table 2.10.
**Dependency of bit size and running time of degree and terms** To illustrate how bit size and running time of `optsage` vary for different degrees and numbers of terms, we restrict ourselves to at most 4 variables. Our numbers from the previous point show, that in these cases bit size and time are similar for fixed $(d, t)$, hence we may aggregate those instances. The results are shown in Table 2.11. We can see that running time and bit size roughly have a linear dependency. On the one hand, their growth is quadratic in the number of terms, which matches with the growth of the problem size in (SAGE). On the other hand, bit size and running time are basically unaffected by the degree. This shows that the bound, given in the worst case analysis, usually is not met.
**Quality of the rounding-projection** Our experiments verify that in the majority of cases the sym-

| $t \setminus d$ | 6 | 8 | 10 | 18 | 20 | 26 | 28 |
|---|---|---|---|---|---|---|---|
| 6 | 912 | 1000 | 1002 | 1170 | 1014 | 955 | 900 |
| | 0.24 | 0.26 | 0.26 | 0.28 | 0.26 | 0.28 | 0.26 |
| 9 | 2731 | 2673 | 2808 | 2890 | 2621 | 3166 | 2471 |
| | 0.66 | 0.65 | 0.70 | 0.68 | 0.61 | 0.82 | 0.62 |
| 12 | 5599 | 6054 | 5449 | 5747 | 5478 | 6007 | 5027 |
| | 1.30 | 1.40 | 1.27 | 1.27 | 1.21 | 1.53 | 1.18 |
| 20 | 9990 | 24078 | 20985 | 21364 | 19324 | 24096 | 17210 |
| | 2.26 | 5.08 | 4.40 | 4.43 | 3.96 | 5.23 | 3.59 |
| 24 | × | 36301 | 33414 | 37080 | 29266 | 37618 | 28090 |
| | | 7.62 | 6.99 | 7.49 | 5.87 | 7.87 | 5.43 |
| 30 | × | 57744 | 56354 | 61564 | 48622 | 59975 | 55000 |
| | | 11.90 | 11.44 | 12.57 | 9.32 | 12.76 | 10.80 |
| 50 | × | × | 180971 | 174464 | 146218 | 196511 | 183598 |
| | | | 36.11 | 34.64 | 27.80 | 38.19 | 38.36 |

Table 2.11: Bit size (upper part) and running time (lower part) of `optsage` in dependency of the degree $d$ and the number of terms $t$ for up to 4 variables; A "×" indicates, that we do not have instances with these parameters in our data set.

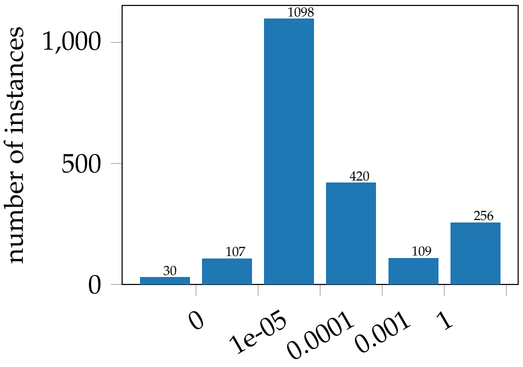


Figure 2.11: Number of instances where the difference of numerical lower bound and exact lower bound lies in the given interval; note that the exact bound sometimes is better.

bolical lower bound does not diverge far from the numerical bound. The detailed distribution is shown in Figure 2.11. Most notably, in 30 instances, the exact lower bound is even *better* than the numerical bound. In 81.9% of the instances, the exact bound differs by at most 0.001 from the numerical value. Only in 256 instances the difference lies above 1. Thus, in the clear majority of examples, the lower bound in exact arithmetic does not differ much from the numerical bound. Also, among the instances with large difference, it can also be that the numerical solution actually lies far away from an exact solution. So it is unclear, whether a large difference is due to bad behavior of the numerical solution, or a large error in the rounding algorithm.

**Rounding time versus solving time** In nearly every case the rounding procedure takes longer than the numerical solving. Only in 8 instances, the rounding took less time. The ratio of the rounding time to the total time ranges from 21.6% to 96.8%, with an average of 88.6%. However, one can implement the rounding procedure much closer to the hardware level, instead of working in Python. Thus, we expect that these ratio can be significantly improved.

**Comparison between SONC and SAGE** In their qualitative behavior, `optsonc` and `optsage` are similar. However, `optsonc` runs faster and has smaller certificates than `optsage`, as shown in Ta-

| $t$ | bit size SONC | bit size SAGE | time SONC | time SAGE |
|---|---|---|---|---|
| 6 | 432 | 1005 | 0.06 | 0.26 |
| 9 | 806 | 2696 | 0.19 | 0.66 |
| 12 | 1261 | 5568 | 0.37 | 1.29 |
| 20 | 2592 | 19203 | 0.64 | 4.00 |
| 24 | 3826 | 32543 | 0.97 | 6.66 |
| 30 | 5029 | 53160 | 1.34 | 10.58 |
| 50 | 10622 | 167971 | 3.95 | 32.78 |

Table 2.12: Comparison of running time and bit size of the certificates between `optsonc` and `optsage`; `optsonc` runs faster and has significantly smaller certificates than `optsage`.

ble 2.12. But one should note that `optsonc` only computes *some* lower bound (not necessarily the optimal SONC bound), whereas `optsage` computes the best bound, that can be obtained via this approach. Still it shows, that for very large instances, SONC is the method of choice, when other approaches fail due to the problem size.

**Comparison with SOS** For polynomials lying in the interior of the SOS cone from Table 2.7, we performed preliminary experiments with `optsage` and `optsage`, which are currently unable to provide nonnegativity certificates. For benchmarks from our database with $n \geq 8$ and $d \geq 10$, `RealCertify` often fails to provide SOS certificates.

CHAPTER 3

# Efficient polynomial optimization

## Contents

As mentioned in the last section of Chapter 2, for optimization problems involving $n$-variate polynomials of degree less than $d$, the size of the matrices involved at step $r \geq d$ of Lasserre's hierarchy of SDP relaxations is proportional to $\binom{n+r}{n}$. Overall, the size of the SDP problems arising from the hierarchy grows rapidly. Section 2.4 proposed a partial remedy to handle polynomial with sparse support without relying on SDP. In this chapter, we outline several other techniques based on SDP to exploit sparsity.

*A scientific challenge with important computational implications is to develop alternative positivity certificates that scale well in terms of computational complexity, at least in some identified class of problems.*

For unconstrained problems involving a large number of variables $n$, a remedy consists of reducing the size of the SDP matrices by discarding the monomials which never appear in the support of the SOS decompositions. This technique, based on a result by Reznick [16], consists of computing the Newton polytope of the input polynomial that we already saw in Section 2.4 (that is, the convex hull of the support of this polynomial) and selecting only monomials with support lying in half of this polytope. For constrained optimization, existing workarounds are based on exploiting a potential symmetry [251] pattern arising in the input polynomials, or bounded degree LP/SDP relaxations [302]. In [179] (see also [300] and the related `SparsePOP` solver [299]), the author derives a sparse version of Putinar's representation [245] for polynomials positive on compact semialgebraic sets. See also [106] for a simpler proof. This variant can be used for cases where the objective function can be written as a sum of polynomials, each of them involving a small number of variables. Sparse polynomial optimization techniques enable us to successfully handle various concrete applications. In energy networks, it is now possible to compute the solution of large-scale power flow problems with up to thousand variables [148]. In [283], the authors derive the sparse analogue of [130] to obtain a hierarchy of upper bounds for the volume of large-scale semialgebraic sets. Specific extensions are given in [C4, R1] to bound the Lipschitz constants of ReLU networks. This latter problem has been investigated in the context of the supervision of T. Chen, one of my PhD students, co-advised with E. Pauwels (assistant Professor, UPS Toulouse) and J.-B. Lasserre. I am also supervising the PhD of N. A. H. Mai on the related topic of sparse decompositions of positive definite forms [R9] .

- After presenting background on correlative sparsity in Section 3.1, I present some new applications in the context of computer arithmetics. First, I outline in Section 3.2 a framework relying on correlative sparsity to produce a hierarchy of upper bounds converging to the absolute roundoff error of a numerical program involving polynomial operations. These results have been published in [J14, J13], and come together with the `Real2Float` software library. I have also investigated in [J11, C11] the use of concurrent techniques, based on Bernstein expansions and sparse Krivine-Stengle representations, during the (unofficial) supervision of the PhD thesis of A. Rocca with T. Dang (senior researcher, CNRS VERIMAG).

- Section 3.3 focuses on optimization of polynomials in noncommuting variables, while taking into account sparsity in the input data. A converging hierarchy of semidefinite relaxations for eigenvalue and trace optimization is provided. This hierarchy is a noncommutative analogue of results due to Lasserre [179] and Waki et al. [300]. The GNS construction is applied to extract optimizers if flatness and irreducibility conditions are satisfied. Among the main techniques used are amalgamation results from operator algebra. The theoretical results are utilized to compute lower bounds on minimal eigenvalue and trace of noncommutative polynomials from the literature. This work was done in collaboration with experts in noncommutative optimization: I. Klep and J. Povh, both professors at the University of Ljubljana.

- Next we show in Section 3.4 how to exploit term (or monomial) sparsity of the input polynomials to obtain a new converging hierarchy of SDP relaxations. The novelty of such relaxations is to involve block-diagonal matrices obtained in an iterative procedure performing completion of the connected components of certain adjacency graphs. The graphs are related to the terms arising in the original data and not to the links between variables. Eventually we show in Section 3.5 how to combine correlative and term sparsity, and successfully apply this combined strategy to solve large-scale optimal power flow instances. This latter part also relates with the ongoing PhD of N. A. H. Mai, currently under my supervision. Contributions on term sparsity, initiated when I started to supervise the postdoc of J. Wang, have led to several academic and industrial collaborations. The first academic collaboration was with experts in control from LTH: Martina Maggio, her PhD student Nils Vreman and Paolo Pazzaglia (postdoc, Saarland University), which allowed us to analyze the stability of control systems under deadline constraints [C13, R12]. The second one is an ongoing collaboration with quantum information physicists from the group of A. Acín at IFCO Barcelona. Last but not least, we are currently investigating more applications to energy networks with industrial colleagues from RTE, Paris.

## 3.1 Correlative sparsity in polynomial optimization

### Chordal graphs and sparse matrices

We briefly recall some basic notions from graph theory. An *(undirected) graph* $G(V, E)$ or simply $G$ consists of a set of nodes $V$ and a set of edges $E \subseteq \{\{v_i, v_j\} \mid (v_i, v_j) \in V \times V\}$. Note that we admit self-loops (i.e. edges that connect the same node) in the edge set $E$. If $G$ is a graph, we will use $V(G)$ and $E(G)$ to indicate the set of nodes of $G$ and the set of edges of $G$, respectively. For two graphs $G, H$, we say that $G$ is a *subgraph* of $H$ if $V(G) \subseteq V(H)$ and $E(G) \subseteq E(H)$, denoted by $G \subseteq H$. For a graph $G(V, E)$, a *cycle* of length $k$ is a set of nodes $\{v_1, v_2, \ldots, v_k\} \subseteq V$ with $\{v_k, v_1\} \in E$ and $\{v_i, v_{i+1}\} \in E$, for $i \in [k-1]$. A *chord* in a cycle $\{v_1, v_2, \ldots, v_k\}$ is an edge $\{v_i, v_j\}$ that joins two nonconsecutive nodes in the cycle. A *clique* $C \subseteq V$ of $G$ is a subset of nodes where $\{v_i, v_j\} \in E$ for any $v_i, v_j \in C$. If a clique $C$ is not a subset of any other clique, then it is called a *maximal clique*.

A graph is called a *chordal graph* if all its cycles of length at least four have a chord. Note that any non-chordal graph $G(V, E)$ can always be extended to a chordal graph $\overline{G}(V, \overline{E})$ by adding appropriate edges to $E$, which is called a *chordal extension* of $G(V,E)$. The chordal extension of $G$ is usually not unique. We use $\overline{G}$ to indicate any specific chordal extension of $G$. For graphs $G \subseteq H$, we assume that $\overline{G} \subseteq \overline{H}$ always holds in the sequel. It is known that maximal cliques of a chordal graph can be enumerated efficiently in linear time in the number of nodes and edges of the graph. See, e.g., [42, 94, 98] for the details.

Suppose $G(V,E)$ is a graph with the node set $V \subseteq \mathbb{N}^n$. We define the *support* of $G$ by

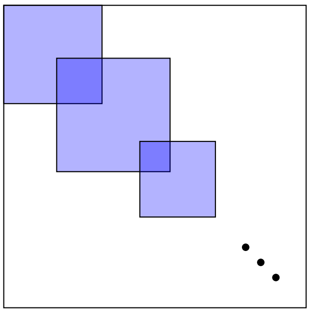
$$\mathrm{supp}(G) := \{\beta + \gamma \mid \{\beta, \gamma\} \in E\}.$$

For a positive $k \in \mathbb{N}$, recall that $\mathbb{S}_k$ is the set of real symmetric matrices of size $k$, and let $\mathbb{S}_k^+$ be its subset of PSD matrices. Given a graph $G(V,E)$, a symmetric matrix $\mathbf{Q}$ with row and column indices labeled by $V$ is said to have sparsity pattern $G$ if $\mathbf{Q}_{\beta\gamma} = \mathbf{Q}_{\gamma\beta} = 0$ whenever $\{\beta, \gamma\} \notin E$. Let $\mathbb{S}(G)$ be the set of real symmetric matrices with sparsity pattern $G$. The PSD matrices with sparsity pattern $G$ form a convex cone

$$\mathbb{S}_{|V|}^+ \cap \mathbb{S}(G) = \{\mathbf{Q} \in \mathbb{S}_G \mid \mathbf{Q} \succeq 0\}. \tag{3.1}$$

A matrix in $\mathbb{S}(G)$ exhibits a *quasi block-diagonal* structure (after an appropriate permutation of rows and columns) as illustrated in Figure 3.1. Each block corresponds to a maximal clique of $G$. The maximal block size is the maximal size of maximal cliques of $G$, namely, the *clique number* of $G$. Note that there might be overlaps between blocks because different maximal cliques may share nodes. For a graph $G$, among all chordal extensions of $G$, there is a maximal one $\overline{G}$: making every connected component of $G$ to be a complete subgraph. Accordingly, the matrix with sparsity pattern $\overline{G}$ is block diagonal (up to permutation). We hereafter refer to this chordal extension as the *maximal* chordal extension. In this chapter, we consider chordal extensions that are subgraphs of the maximal chordal extension.

Figure 3.1: The quasi block-diagonal structure of matrices in $\mathbb{S}_G$. The blue area indicates the positions of possible nonzero entries.



Given a maximal clique $C$ of $G(V,E)$, we define a matrix $\mathbf{R}_C \in \mathbb{R}^{|C| \times |V|}$ as

$$(\mathbf{R}_C)_{i\beta} = \begin{cases} 1, & \text{if } C(i) = \beta, \\ 0, & \text{otherwise.} \end{cases} \tag{3.2}$$

where $C(i)$ denotes the $i$-th node in $C$, sorted with respect to an ordering compatible with $V$. Note that $\mathbf{Q}_C = \mathbf{R}_C \mathbf{Q} \mathbf{R}_C^T \in \mathbb{S}_{|C|}$ extracts a principal submatrix $\mathbf{Q}_C$ defined by the indices in the clique $C$ from a symmetry matrix $\mathbf{Q}$, and $\mathbf{Q} = \mathbf{R}_C^T \mathbf{Q}_C \mathbf{R}_C$ inflates a $|C| \times |C|$ matrix $\mathbf{Q}_C$ into a sparse $|V| \times |V|$ matrix $\mathbf{Q}$.

When the sparsity pattern graph $G$ is chordal, the cone $\mathbb{S}_{|V|}^+ \cap \mathbb{S}(G)$ can be decomposed as a sum of simple convex cones, as stated in the following theorem.

**Theorem 3.1.1 ([293], Theorem 9.2)** *Let $G(V,E)$ be a chordal graph and assume that $C_1,\ldots,C_t$ are all of the maximal cliques of $G(V,E)$. Then a matrix $\mathbf{Q}\in\mathbb{S}^+_{|V|}\cap\mathbb{S}(G)$ if and only if there exist $\mathbf{Q}_k\in\mathbb{S}^+_{|C_k|}$ for $k\in[t]:=\{1,\ldots,t\}$ such that $\mathbf{Q}=\sum_{k=1}^t\mathbf{R}_{C_k}^T\mathbf{Q}_k\mathbf{R}_{C_k}$.*

Given a graph $G(V,E)$, let $\Pi_G$ be the projection from $\mathbb{S}_{|V|}$ to the subspace $\mathbb{S}(G)$, i.e., for $\mathbf{Q}\in\mathbb{S}_{|V|}$,

$$\Pi_G(\mathbf{Q})_{\beta\gamma}=\begin{cases}\mathbf{Q}_{\beta\gamma}, & \text{if } \{\beta,\gamma\}\in E,\\ 0, & \text{otherwise.}\end{cases} \tag{3.3}$$

We denote by $\Pi_G(\mathbb{S}^+_{|V|})$ the set of matrices in $\mathbb{S}(G)$ that have a PSD completion, i.e.,

$$\Pi_G(\mathbb{S}^+_{|V|})=\{\Pi_G(\mathbf{Q})\mid\mathbf{Q}\in\mathbb{S}^+_{|V|}\}. \tag{3.4}$$

One can check that the PSD completable cone $\Pi_G(\mathbb{S}^+_{|V|})$ and the PSD cone $\mathbb{S}^+_{|V|}\cap\mathbb{S}(G)$ form a pair of dual cones in $\mathbb{S}(G)$. Moreover, for a chordal graph $G$, the decomposition result for the cone $\mathbb{S}^+_{|V|}\cap\mathbb{S}(G)$ in Theorem 3.1.1 leads to the following characterization of the PSD completable cone $\Pi_G(\mathbb{S}^+_{|V|})$.

**Theorem 3.1.2 ([293], Theorem 10.1)** *Let $G(V,E)$ be a chordal graph and assume that $C_1,\ldots,C_t$ are all of the maximal cliques of $G(V,E)$. Then a matrix $\mathbf{Q}\in\Pi_G(\mathbb{S}^+_{|V|})$ if and only if $\mathbf{Q}_k=\mathbf{R}_{C_k}\mathbf{Q}\mathbf{R}_{C_k}^T\succeq 0$ for $k\in[t]$.*

For more details about sparse matrices and chordal graphs, the reader may refer to [293].

## Correlative sparsity

To exploit correlative sparsity (CS) in the moment-SOS hierarchy for POP, one proceeds in two steps: 1) partition the set of variables into cliques according to the links between variables emerging in the input polynomial system, and 2) construct a quasi block moment-SOS hierarchy with respect to the former partition of variables [300]. Let us recall the general formulation (2.1) of POP:

$$\mathbf{P}:\quad f_{\min}\;=\;\min_{\mathbf{x}\in\mathbb{R}^n}\{\,f(\mathbf{x}):\mathbf{x}\in\mathbf{X}\,\}, \tag{3.5}$$

where $\mathbf{X}=\{\mathbf{x}\in\mathbb{R}^n:g_1(\mathbf{x})\geq 0,\ldots,g_m(\mathbf{x})\geq 0\}$, is defined as in (1.1). Concretely, we define the correlative sparsity pattern (csp) graph associated to POP (3.5) to be the graph $G^{\text{csp}}$ with nodes $V=[n]:=\{1,2,\ldots,n\}$ and edges $E$ satisfying $\{i,j\}\in E$ if one of followings holds:

(i) there exists $\alpha\in\text{supp}(f)$ s.t. $\alpha_i>0,\alpha_j>0$;

(ii) there exists $k$, with $1\leq k\leq m$, s.t. $x_i,x_j\in\text{var}(g_k)$, where $\text{var}(g_k)$ is the set of variables involved in $g_k$.

Let $\overline{G}^{\text{csp}}$ be a chordal extension of $G^{\text{csp}}$ and $I_l, l\in[p]$ be the maximal cliques of $\overline{G}^{\text{csp}}$ with cardinal denoted by $n_l$. Let $\mathbb{R}[\mathbf{x},I_l]$ denote the ring of polynomials in the $n_l$ variables $\mathbf{x},I_l=\{x_i\mid i\in I_l\}$. By construction, one can decompose the objective function as $f=f_1+\cdots+f_p$, with $f_l\in\mathbb{R}[\mathbf{x},I_l]$, for all $l\in[p]$. Let $\Sigma[\mathbf{x},I_l]\subset\mathbb{R}[\mathbf{x},I_l]$ be the corresponding cone of SOS polynomials. We then partition the polynomials $g_1,\ldots,g_m$ involved in the constraints, into groups $\{g_j\mid j\in J_l\},l\in[p]$ which satisfy:

(i) $J_1,\ldots,J_p\subseteq[m]:=\{1,2,\ldots,m\}$ are pairwise disjoint and $\cup_{l=1}^p J_l=[m]$;

(ii) for any $l\in[p]$ and $j\in J_l$, $\text{var}(g_j)\subseteq I_l$.

Possibly after some reordering of the cliques, the above subsets $\{I_1, \ldots, I_p\}$ satisfy the so-called *running intersection property (RIP)*, i.e., for all $l \in [p-1]$, one has

$$I_{l+1} \cap \bigcup_{j \leq l} I_j \subseteq I_k \quad \text{for some } k \leq l\,. \tag{3.6}$$

Note also that (3.6) always holds when $p = 2$. For the applications presented in this chapter, we will not necessarily have to rely on the chordal extension of the csp graph. One sufficient way to ensure the convergence of the CS-based hierarchies of SDP relaxations is to check that (3.6) holds. In addition, all variables involved in POP (3.5) will be bounded. More specifically, we will rely on the following assumption:

**Assumption 3.1.3** *Let* $\mathbf{X}$ *be a basic compact semialgebraic set as in* (1.1) *and* $\{I_1, \ldots, I_p\}$ *be a partition of* $[n]$. *For all* $l \in [p]$, *there exists a real* $N_l > 0$ *such that one of the polynomials describing* $\mathbf{X}$ *is* $N_l - \sum_{i \in I_l} x_i^2$.

Next, with $l \in [p]$ fixed, $r$ a positive integer and $g \in \mathbb{R}[\mathbf{x}, I_l]$, let $\mathbf{M}_r(\mathbf{y}, I_l)$ (resp. $\mathbf{M}_r(g\mathbf{y}, I_l)$) be the moment (resp. localizing) submatrix obtained from $\mathbf{M}_r(\mathbf{y})$ (resp. $\mathbf{M}_r(g\mathbf{y})$) by retaining only those rows (and columns) $\beta = (\beta_i) \in \mathbb{N}_r^n$ of $\mathbf{M}_r(\mathbf{y})$ (resp. $\mathbf{M}_r(g\mathbf{y})$) with $\operatorname{supp}(\beta) \subseteq I_l$, where $\operatorname{supp}(\beta) = \{i \mid \beta_i \neq 0\}$.

Then with $r \geq r_{\min} := \max\{\lceil \deg(f)/2 \rceil, r_1, \ldots, r_m\}$, the moment-SOS hierarchy based on CS for (3.5), is defined as:

$$\mathbf{P}_{\text{cs}}^r : \begin{cases} \inf & L_{\mathbf{y}}(f) \\ \text{s.t.} & \mathbf{M}_r(\mathbf{y}, I_l) \succeq 0, \quad l \in [p], \\ & \mathbf{M}_{r-r_j}(g_j\mathbf{y}, I_l) \succeq 0, \quad j \in J_l, \quad l \in [p], \\ & y_0 = 1, \end{cases} \tag{3.7}$$

with optimal value denoted by $f_{\min,\text{cs}}^r$ or sometimes by $f_{\min}^r$, when it is clear from the context that this bound is obtained thanks to an SDP after exploiting CS. The dual of (3.7) is

$$\begin{array}{rl} \sup\limits_{b,\sigma_j} & b \\ \text{s.t.} & f - b = \sum_{l=1}^p (\sigma_l + \sum_{j \in J_l} \sigma_{jl} g_j)\,, \\ & b \in \mathbb{R}\,,\ \sigma_l, \sigma_{jl} \in \Sigma[\mathbf{x}, I_l]\,, \quad j \in J_l, \quad l \in [p], \\ & \deg(\sigma_l), \deg(\sigma_{jl} g_j) \leq 2r\,, \quad j \in J_l, \quad l \in [p]\,. \end{array} \tag{3.8}$$

In the following, we refer to (3.7)-(3.8) as the CS primal-dual (moment-SOS) hierarchy for the POP (3.5), also abbreviated as CSSOS hierarchy. To prove that the sequence $(f_{\text{cs}}^r)_r$ converges to the global optimum $f_{\min}$ of the original POP. (3.5), we rely on the following theorem, which is a sparse version of Putinar's Positivstellensatz.

**Theorem 3.1.4 ([179], Corollary 3.9)** *Let* $f \in \mathbb{R}[\mathbf{x}]$ *be positive on a basic compact semialgebraic set* $\mathbf{X}$ *as in Assumption* 3.1.3. *Assume that both index sets* $[n]$ *and* $[m]$ *are partitioned into p disjoint sets* $I_1, \ldots, I_p$ *and* $J_1, \ldots, J_p$, *respectively. Assume that the two partitions satisfy:*

(i) *For all* $l \in [p]$ *and* $j \in J_l$, $g_j \in \mathbb{R}[\mathbf{x}, I_l]$;

(ii) *The function f can be decomposed as* $f = f_1 + \cdots + f_p$, *with* $f_l \in \mathbb{R}[\mathbf{x}, I_l]$, *for all* $l \in [p]$;

(iii) *RIP* (3.6) *holds.*

*Then*

$$f = \sum_{l=1}^p (\sigma_l + \sum_{j \in J_l} \sigma_{jl} g_j)\,,$$

*for some polynomials* $\sigma_l, \sigma_{jl} \in \Sigma[\mathbf{x}, I_l]$, $j \in J_l$, $l \in [p]$.

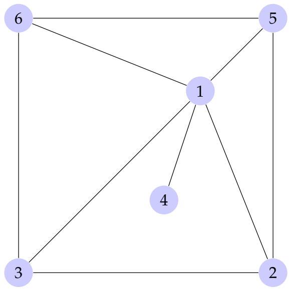


Figure 3.2: csp graph for the variables of $f$ from Example 3.1.1.

**Example 3.1.1** *Consider an instance of POP (3.5) with* $f(\mathbf{x}) = x_2x_5 + x_3x_6 - x_2x_3 - x_5x_6 + x_1(-x_1 + x_2 + x_3 - x_4 + x_5 + x_6)$ *and* $\mathbf{X} = [4, 6.36]^6$*, which can be written as:*

$$\mathbf{X} = \{\, \mathbf{x} \in \mathbb{R}^n \,:\, g_1(\mathbf{x}) \geq 0, \ldots, g_7(\mathbf{x}) \geq 0 \,\},$$

*with* $g_i(\mathbf{x}) = (6.36 - x_i)(x_i - 4)$ *for each* $i \in [6]$ *and* $g_7(\mathbf{x}) := 243 - \sum_{i=0}^{6} x_i^2$*. Here one can choose a constant* $M = 243$ *so that* $M \geq 6 \times 6.36^2$ *and Assumption 1.1.1 is fulfilled. Here,* $n = 6$ *and the number of optimization constraints is* 7*. For* $r = 1$*, the dense SDP relaxation involves* $\binom{n+2r}{2r} = \binom{6+2}{2} = 28$ *variables and provides a lower bound of* 20.755 *for* $f_{\min}$*. The dense SDP relaxation at* $r = 2$ *involves* $\binom{6+4}{4} = 210$ *variables and provides a tighter lower bound of* 20.8608*. The* $6 \times 6$ *CS matrix associated to the csp graph* $G^{csp}$*, depicted in Figure 3.2 is*

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

*The graph* $G^{csp}$ *is chordal with maximal cliques* $I_1 = \{1, 4\}$, $I_2 = \{1, 2, 3\}$, $I_3 = \{1, 2, 5\}$, $I_4 = \{1, 5, 6\}$ *and* $I_5 = \{1, 3, 6\}$*. For* $r = 2$*, the dense SDP relaxation involves* $\binom{6+4}{4} = 210$ *variables against* $\binom{2+4}{4} + 4\binom{3+4}{4} = 155$ *for the sparse variant (3.8). The dense SDP relaxation at* $r = 3$ *involves* 924 *variables against* 364 *for the sparse variant (3.8). This difference becomes significant while considering that the time complexity of SDP is polynomial w.r.t. the number of SDP variables with an exponent greater than 3 (see [35, Chapter 4] for more details).*

## 3.2 Application to roundoff errors

In this section, we describe an optimization framework to provide upper bounds on absolute roundoff errors of floating-point nonlinear programs, involving polynomials. The efficiency of this framework is based on the CSSOS hierarchy which exploits CS of the input polynomial data, as described in Section 3.1.

### Classical approaches

Constructing numerical programs which perform accurate computation turns out to be difficult, due to finite numerical precision of implementations such as floating-point or fixed-point representations. Finite-precision numbers induce roundoff errors, and knowledge of the range of these roundoff errors is required to fulfill safety criteria of critical programs, as typically arising in modern embedded systems such as aircraft controllers. Such a knowledge can be used in general for developing accurate numerical software, but is also particularly relevant when considering migration of algorithms onto hardware (e.g. FPGAs). The advantage of architectures based on FPGAs is that they allow more flexible choices in number representations, rather than limiting the choice between IEEE standard single or double precision. Indeed, in this case, we benefit from a more flexible number representation while still ensuring guaranteed bounds on the program output.

To obtain lower bounds on roundoff errors, one can rely on testing approaches, such as meta-heuristic search [45] or under-approximation tools (e.g. `s3fp` [62]). Here, we are interested in efficiently handling the complementary over-approximation problem, namely to obtain precise upper bounds on the error. This problem boils down to finding tight abstractions of linearities or non-linearities while being able to bound the resulting approximations in an efficient way. For computer programs consisting of linear operations, automatic error analysis can be obtained with well-studied optimization techniques based on SAT/SMT solvers [118] and affine arithmetic [77]. However, non-linear operations are key to many interesting computational problems arising in physics, biology, controller implementations and global optimization. Two promising frameworks have been designed to provide upper bounds for roundoff errors of nonlinear programs. The corresponding algorithms rely on Taylor-interval methods [273], implemented in the `FPTaylor` tool, and on combining SMT with interval arithmetic [72], implemented in the ROSA real compiler.

Our method to bound the error is a decision procedure based on a specialized variant of the Lasserre hierarchy [179], outlined in Section 3.1. The procedure relies on SDP to provide sparse SOS decompositions of nonnegative polynomials. Our framework handles polynomial program analysis (involving the operations $+, \times, -$) as well as extensions to the more general class of semi-algebraic and transcendental programs (involving $\sqrt{\cdot}, /, \min, \max, \arctan, \exp$), following the approximation scheme described in [J18]. For the sake of conciseness, we focus in this manuscript on polynomial programs only. The interested reader can find more details in the related publication [J14].

## Polynomial programs

Here we consider a given program that implements a polynomial expression $f$ with input variables $\mathbf{x}$ satisfying a set of constraints $\mathbf{X}$. We assume that $\mathbf{X}$ is included in a box (i.e. a product of closed intervals) and that $\mathbf{X}$ is encoded as in (1.1):

$$\mathbf{X} := \{ \mathbf{x} \in \mathbb{R}^n : g_1(\mathbf{x}) \geq 0, \ldots, g_k(\mathbf{x}) \geq 0 \},$$

for polynomial functions $g_1, \ldots, g_k$.

The type of numerical constants is denoted by `C`. In our current implementation, the user can choose either 64 bit floating-point or arbitrary-size rational numbers. The inductive type of polynomial expressions $f, g_1, \ldots, g_k$ with coefficients in `C` is `pExprC` defined as follows:

```
type pexprC =
 Pc of C
|Px of positive
|Psub of pexprC * pexprC | Pneg of pexprC
|Padd of pexprC * pexprC
|Pmul of pexprC * pexprC
```

The constructor `Px` takes a positive integer as argument to represent either an input or local variable. One obtains rounded expressions using a recursive procedure `round`. We adopt the standard practice [139] to approximate a real number $x$ with its closest floating-point representation

```
Require: input variables $\mathbf{x}$, input constraints $\mathbf{X}$, nonlinear expression $f$, rounded expression $\hat{f}$,
    error variables $\mathbf{e}$, error constraints $\mathbf{E}$, relaxation order $r$
Ensure: interval enclosure of the error $\hat{f} - f$ over $\mathbf{K} := \mathbf{X} \times \mathbf{E}$
1: Define the absolute error $\Delta(\mathbf{x}, \mathbf{e}) := \hat{f}(\mathbf{x}, \mathbf{e}) - f(\mathbf{x})$
2: Compute $\ell(\mathbf{x}, \mathbf{e}) := \Delta(\mathbf{x}, 0) + \sum_{j=1}^{m} \frac{\partial \Delta(\mathbf{x}, \mathbf{e})}{\partial e_j}(\mathbf{x}, 0)\, e_j$
3: Define $h := \Delta - \ell$
4: $[\underline{h}, \overline{h}] :=$ ia_bound$(h, \mathbf{K})$                    ▷ Compute bounds for $h$
5: $[\ell^r_{\text{min}}, \ell^r_{\text{max}}] :=$ cs_sdp$(\ell, \mathbf{K}, r)$              ▷ Compute bounds for $\ell$
6: return $[\ell^r_{\text{min}} + \underline{h}, \ell^r_{\text{max}} + \overline{h}]$
```

Figure 3.3: `bound`: our algorithm to compute roundoff errors bounds of nonlinear programs.

$\hat{x} = x(1+e)$, with $|e|$ is less than the machine precision $\varepsilon$. In the sequel, we neglect both overflow and denormal range values. The operator $\hat{\cdot}$ is called the rounding operator and can be selected among rounding to nearest, rounding toward zero (resp. $\pm\infty$). In the sequel, we assume rounding to nearest. The scientific notation of a binary (resp. decimal) floating-point number $\hat{x}$ is a triple $(\mathrm{s}, \mathrm{sig}, \mathrm{exp})$ consisting of a sign bit s, a *significand* sig $\in [1, 2)$ (resp. $[1, 10)$) and an *exponent* exp, yielding numerical evaluation $(-1)^{\mathrm{s}}\, \mathrm{sig}\, 2^{\mathrm{exp}}$ (resp. $(-1)^{\mathrm{s}}\, \mathrm{sig}\, 10^{\mathrm{exp}}$).

The upper bound on the relative floating-point error is given by $\varepsilon = 2^{-\mathrm{prec}}$, where prec is called the *precision*, referring to the number of significand bits used. For single precision floating-point, one has prec $= 24$. For double (resp. quadruple) precision, one has prec $= 53$ (resp. prec $= 113$). Let $\mathbf{F}$ be the set of binary floating-point numbers.

For each real-valued operation $\mathtt{bop}_{\mathbb{R}} \in \{+, -, \times\}$, the result of the corresponding floating-point operation $\mathtt{bop}_{\mathbf{F}} \in \{\oplus, \ominus, \otimes\}$ satisfies the following when complying with IEEE 754 standard arithmetic [143] (without overflow, underflow and denormal occurrences):

$$\mathtt{bop}_{\mathbf{F}}(\hat{x}, \hat{y}) = \mathtt{bop}_{\mathbb{R}}(\hat{x}, \hat{y})\,(1+e) \ , \quad |\, e \,| \leq \varepsilon = 2^{-\mathrm{prec}} \ . \tag{3.9}$$

Then, we denote by $\hat{f}(\mathbf{x}, \mathbf{e})$ the rounded expression of $f$ after applying the `round` procedure, introducing additional error variables $\mathbf{e}$.

### Upper bounds on roundoff errors

The algorithm `bound`, depicted in Figure 3.3, takes as input $\mathbf{x}$, $\mathbf{X}$, $f$, $\hat{f}$, $\mathbf{e}$ as well as the set $\mathbf{E}$ of bound constraints over $\mathbf{e}$. For a given machine $\varepsilon$, one has $\mathbf{E} := [-\varepsilon, \varepsilon]^m$, with $m$ being the number of error variables. This algorithm actually relies on the CS-based SDP hierarchy (3.8) from Section 3.1, thus `bound` also takes as input a relaxation order $r \in \mathbb{N}$. The algorithm provides as output an interval enclosure of the error $\hat{f}(\mathbf{x}, \mathbf{e}) - f(\mathbf{x})$ over $\mathbf{K} := \mathbf{X} \times \mathbf{E}$. From this interval $[f^r_{\text{min}}, f^r_{\text{max}}]$, one can compute $|f|^r_{\text{max}} := \max\{-f^r_{\text{min}}, f^r_{\text{max}}\}$, which is a sound upper bound of the maximal absolute error $|\Delta|_{\text{max}} := \max_{(\mathbf{x}, \mathbf{e}) \in \mathbf{K}} |\, \hat{f}(\mathbf{x}, \mathbf{e}) - f(\mathbf{x}) \,|$.

After defining the absolute roundoff error $\Delta := \hat{f} - f$ (Step 1), one decomposes $\Delta$ as the sum of an expression $\ell$ which is affine w.r.t. the error variable $\mathbf{e}$ and a remainder $h$. One way to obtain $\ell$ is to compute the vector of partial derivatives of $\Delta$ w.r.t. $\mathbf{e}$ evaluated at $(\mathbf{x}, 0)$ and finally to take the inner product of this vector and $\mathbf{e}$ (Step 2). Then, the idea is to compute a precise bound of $\ell$ and a coarse bound of $h$. The underlying reason is that $h$ involves error term products of degree greater than 2 (e.g. $e_1 e_2$), yielding an interval enclosure of *a priori* much smaller width, compared to the interval enclosure of $\ell$. One obtains the interval enclosure of $h$ using the procedure `ia_bound` implementing basic interval arithmetic (Step 4) to bound the remainder of the multivariate Taylor expansion of $\Delta$ w.r.t. $\mathbf{e}$, expressed as a combination of the second-error derivatives (similar as in [273]). The main algorithm presented in Figure 3.3 is very similar to the algorithm of `FPTaylor` [273], except that SDP based techniques are used instead of the global optimization procedure from [273]. Note that overflow and denormal are neglected here but one could handle them, as in [273], by adding additional error variables and discarding the related terms using naive interval arithmetic.

The bound of $\ell$ is provided through the `cs_sdp` procedure, which solves two CS-based SDP instances of (3.8), at relaxation order $r$. We now give more explanation about this procedure. We can map each input variable $x_i$ to the integer $i$, for all $i \in [n]$, as well as each error variable $e_j$ to $n + j$, for all $j \in [m]$. Then, define the sets $I_1 := [n] \cup \{n+1\}, \dots, I_m := [n] \cup \{n+m\}$. Here, we take advantage of the csp of $\ell$ by using $m$ distinct sets of cardinality $n+1$ rather than a single one of cardinality $n+m$, i.e., the total number of variables. Note that these subsets satisfy (3.6) and one can write $\ell(\mathbf{x}, \mathbf{e}) = \Delta(\mathbf{x}, 0) + \sum_{j=1}^{m} \frac{\partial \Delta(\mathbf{x},\mathbf{e})}{\partial e_j}(\mathbf{x}, 0)\, e_j$. After noticing that $\Delta(\mathbf{x}, 0) = \hat{f}(\mathbf{x}, 0) - f(\mathbf{x}) = 0$, one can scale the optimization problems by writing

$$\ell(\mathbf{x}, \mathbf{e}) = \sum_{j=1}^{m} s_j(\mathbf{x}) e_j = \varepsilon \sum_{j=1}^{m} s_j(\mathbf{x}) \frac{e_j}{\varepsilon}\,, \tag{3.10}$$

with $s_j(\mathbf{x}) := \frac{\partial \Delta(\mathbf{x},\mathbf{e})}{\partial e_j}(\mathbf{x}, 0)$, for all $j \in [m]$. Replacing $\mathbf{e}$ by $\mathbf{e}/\varepsilon$ leads to computing an interval enclosure of $\ell/\varepsilon$ over $\mathbf{K}' := \mathbf{X} \times [-1, 1]^m$. As usual from Assumption 1.1.1, there exists an integer $N > 0$ such that $N - \sum_{i=1}^{n} x_i^2 \geq 0$, as the input variables satisfy box constraints. Moreover, to fulfill Assumption 3.1.3, one encodes $\mathbf{K}'$ as follows:

$$\begin{aligned}\mathbf{K}' := \{\, (\mathbf{x}, \mathbf{e}) \in \mathbb{R}^{n+m} \;:\; & g_1(\mathbf{x}) \geq 0, \dots, g_k(\mathbf{x}) \geq 0\,, \\ & g_{k+1}(\mathbf{x}, e_1) \geq 0, \dots, g_{k+m}(\mathbf{x}, e_m) \geq 0 \,\}\,,\end{aligned}$$

with $g_{k+j}(\mathbf{x}, e_j) := N + 1 - \sum_{i=1}^{n} x_i^2 - e_j^2$, for all $j \in [m]$. The index set of variables involved in $g_j$ is $[n]$ for all $j \in [k]$. The index set of variables involved in $g_{k+j}$ is $I_j$ for all $j \in [m]$.

Then, one can compute a lower bound of the minimum of $\ell'(\mathbf{x}, \mathbf{e}) := \ell(\mathbf{x}, \mathbf{e})/\varepsilon = \sum_{j=1}^{m} s_j(\mathbf{x}) e_j$ over $\mathbf{K}'$ by solving the following CS-based SDP problem:

$$\begin{aligned}\ell'^{r}_{\min} := \sup_{b, \sigma_j} \quad & b \\ \text{s.t.} \quad & \ell' - b = \sigma_0 + \textstyle\sum_{j=1}^{k+m} \sigma_j g_j\,, \\ & b \in \mathbb{R}\,,\ \sigma_0 \in \textstyle\sum_{j=1}^{m} \Sigma[(\mathbf{x}, \mathbf{e}), I_j]\,, \\ & \sigma_j \in \Sigma[(\mathbf{x}, \mathbf{e}), J_j]\,,\ j \in [k+m]\,, \\ & \deg(\sigma_j g_j) \leq 2r\,,\ j = 0, \dots, k+m\,.\end{aligned} \tag{3.11}$$

A feasible solution of Problem (3.11) ensures the existence of $\sigma^1 \in \Sigma[(\mathbf{x}, e_1)], \dots, \sigma^m \in \Sigma[(\mathbf{x}, e_m)]$ such that $\sigma_0 = \sum_{j=0}^{m} \sigma^j$, allowing the following reformulation:

$$\begin{aligned}\ell'^{r}_{\min} = \sup_{b, \sigma^j, \sigma_j} \quad & b \\ \text{s.t.} \quad & \ell' - b = \textstyle\sum_{j=1}^{m} \sigma^j + \sum_{j=1}^{k+m} \sigma_j g_j\,, \\ & b \in \mathbb{R}\,,\ \sigma_j \in \Sigma[\mathbf{x}]\,,\ j \in [m]\,, \\ & \sigma^j \in \Sigma[(\mathbf{x}, e_j)]\,, \deg(\sigma^j) \leq 2r\,,\ j \in [m]\,, \\ & \deg(\sigma_j g_j) \leq 2r\,,\ j \in [k+m]\,.\end{aligned} \tag{3.12}$$

An upper bound $\ell'^{r}_{\max}$ can be obtained by replacing sup with inf and $\ell' - b$ by $b - \ell'$ in Problem (3.12). Our optimization procedure `cs_sdp` computes the lower bound $\ell'^{r}_{\min}$ as well as an upper bound $\ell'^{r}_{\max}$ of $\ell'$ over $\mathbf{K}'$ then returns the interval $[\varepsilon\, \ell'^{r}_{\min}, \varepsilon\, \ell'^{r}_{\max}]$, which is a sound enclosure of the values of $\ell$ over $\mathbf{K}$.

We emphasize two advantages of the decomposition $\Delta = \ell + h$ and more precisely of the linear dependency of $\ell$ w.r.t. $\mathbf{e}$: scalability and robustness to SDP numerical issues. First, no computation is required to determine the correlation sparsity pattern of $\ell$, by comparison to the general case. Thus, it becomes much easier to handle the optimization of $\ell$ with the sparse SDP Problem (3.12) rather than with the corresponding instance of the dense relaxation $(\mathbf{P}^r)$, given in (2.6). While the latter involves $\binom{n+m+2r}{2r}$ SDP variables, the former involves only $m\binom{n+1+2r}{2r}$ variables, ensuring the scalability of our framework. In addition, the linear dependency of $\ell$ w.r.t. $\mathbf{e}$ allows us to scale the error variables and optimize over a set of variables lying in $\mathbf{K}' := \mathbf{X} \times [-1,1]^m$. It ensures that the range of input variables does not significantly differ from the range of error variables. This condition is mandatory while considering SDP relaxations because most SDP solvers (e.g. MOSEK [7]) are implemented using double precision floating-point. It is impossible to optimize $\ell$ over $\mathbf{K}$ (rather than $\ell'$ over $\mathbf{K}'$) when the maximal value $\varepsilon$ of error variables is less than $2^{-53}$, due to the fact that SDP solvers would treat each error variable term as 0, and consequently $\ell$ as the zero polynomial. Thus, this decomposition insures our framework against numerical issues related to finite-precision implementation of SDP solvers.

Let us consider the interval enclosure $[\ell_{\min}, \ell_{\max}]$, with $\ell_{\min} := \inf_{(\mathbf{x},\mathbf{e})\in\mathbf{K}} \ell(\mathbf{x},\mathbf{e})$ and $\ell_{\max} := \sup_{(\mathbf{x},\mathbf{e})\in\mathbf{K}} \ell(\mathbf{x},\mathbf{e})$. The next lemma states that one can approximate this interval as closely as desired using the `cs_sdp` procedure.

**Lemma 3.2.1 (Convergence of the `cs_sdp` procedure)** *Let $[\ell^r_{\min}, \ell^r_{\max}]$ be the interval enclosure returned by the procedure* `cs_sdp`$(\ell, \mathbf{K}, r)$. *The sequence $([\ell^r_{\min}, \ell^r_{\max}])_{r\in\mathbb{N}}$ converges to $[\ell_{\min}, \ell_{\max}]$ when $r$ goes to infinity.*

The proof of Lemma 3.2.1 is based on the fact that the assumptions of Theorem 3.1.4 are fulfilled for our specific roundoff error problem. This result guarantees asymptotic convergence to the exact enclosure of $\ell$ when the relaxation order $r$ tends to infinity. However, it is more reasonable in practice to keep this order as small as possible to obtain tractable SDP relaxations. Hence, we generically solve each instance of Problem (3.12) at the minimal relaxation order, that is $r_{\min} = \max\{\lceil \deg \ell/2\rceil), \max_{1\le j\le k+m}\{\lceil \deg(g_j)/2\rceil)\}\}$. Afterwards, we rely on the COQ computer assistant to obtain formally certified upper bounds for the roundoff error; see [J14, § 2.3] for more details.

## Overview of numerical experiments

We present an overview of our method and of the capabilities of related techniques, using an example. Consider a program implementing the following polynomial expression $f$:

$$\begin{aligned} f(\mathbf{x}) := {} & x_2 \times x_5 + x_3 \times x_6 - x_2 \times x_3 - x_5 \times x_6 \\ & + x_1 \times (-x_1 + x_2 + x_3 - x_4 + x_5 + x_6), \end{aligned}$$

where the six-variable vector $\mathbf{x} := (x_1, x_2, x_3, x_4, x_5, x_6)$ is the input of the program. Here the set $\mathbf{X}$ of possible input values is a product of closed intervals: $\mathbf{X} = [4.00, 6.36]^6$. This function $f$ together with the set $\mathbf{X}$ appear in many inequalities arising from the the proof of the Kepler Conjecture [116], yielding challenging global optimization problems.

The polynomial expression $f$ is obtained by performing 15 basic operations (1 negation, 3 subtractions, 6 additions and 5 multiplications). When executing this program with a set of floating-point numbers $\hat{\mathbf{x}} := (\hat{x}_1, \hat{x}_2, \hat{x}_3, \hat{x}_4, \hat{x}_5, \hat{x}_6) \in \mathbf{X}$, one actually computes a floating-point result $\hat{f}$, where all operations $+, -, \times$ are replaced by the respectively associated floating-point operations $\oplus, \ominus, \otimes$. The results of these operations comply with IEEE 754 standard arithmetic [143]. Here, for the sake of clarity, we do not consider real input variables. For instance, (in the absence of underflow) one can write $\hat{x}_2 \otimes \hat{x}_5 = (x_2 \times x_5)(1 + e_1)$, by introducing an error variable $e_1$ such that

$-\varepsilon \leq e_1 \leq \varepsilon$, where the bound $\varepsilon$ is the machine precision (e.g. $\varepsilon = 2^{-24}$ for single precision). One would like to bound the absolute roundoff error $|\Delta(\mathbf{x}, \mathbf{e})| := |\hat{f}(\mathbf{x}, \mathbf{e}) - f(\mathbf{x})|$ over all possible input variables $\mathbf{x} \in \mathbf{X}$ and error variable $e_1, \ldots, e_{15} \in [-\varepsilon, \varepsilon]$. Let us define $\mathbf{E} := [-\varepsilon, \varepsilon]^{15}$ and $\mathbf{K} := \mathbf{X} \times \mathbf{E}$. Then our bound problem can be cast as finding the maximum $|\Delta|_{\max}$ of $|\Delta|$ over $\mathbf{K}$, yielding the following nonlinear optimization problem:

$$\begin{aligned} |\Delta|_{\max} &:= \max_{(\mathbf{x},\mathbf{e}) \in \mathbf{K}} |\Delta(\mathbf{x}, \mathbf{e})| \\ &= \max\{-\min_{(\mathbf{x},\mathbf{e}) \in \mathbf{K}} \Delta(\mathbf{x}, \mathbf{e}), \max_{(\mathbf{x},\mathbf{e}) \in \mathbf{K}} \Delta(\mathbf{x}, \mathbf{e})\} \ , \end{aligned} \tag{3.13}$$

One can directly try to solve these two POP using classical SDP relaxations [180]. As in [273], one can also decompose the error term $\Delta$ as the sum of a term $\ell(\mathbf{x}, \mathbf{e})$, which is affine w.r.t. $\mathbf{e}$, and a nonlinear term $h(\mathbf{x}, \mathbf{e}) := \Delta(\mathbf{x}, \mathbf{e}) - \ell(\mathbf{x}, \mathbf{e})$. Then the triangular inequality yields:

$$|\Delta|_{\max} \leq \max_{(\mathbf{x},\mathbf{e}) \in \mathbf{K}} |\ell(\mathbf{x}, \mathbf{e})| + \max_{(\mathbf{x},\mathbf{e}) \in \mathbf{K}} |h(\mathbf{x}, \mathbf{e})| \ . \tag{3.14}$$

It follows for this example that $\ell(\mathbf{x}, \mathbf{e}) = x_2 x_5 e_1 + x_3 x_6 e_2 + (x_2 x_5 + x_3 x_6) e_3 + \cdots + f(\mathbf{x}) e_{15} = \sum_{i=1}^{15} s_i(\mathbf{x}) e_i$, with $s_1(\mathbf{x}) := x_2 x_5, s_2(\mathbf{x}) := x_3 x_6, \ldots, s_{15}(\mathbf{x}) := f(\mathbf{x})$. The *Symbolic Taylor Expansions* method [273] consists of using a simple branch and bound algorithm based on interval arithmetic to compute a rigorous interval enclosure of each polynomial $s_i$, $i \in [15]$, over $\mathbf{X}$ and finally obtain an upper bound of $|\ell| + |h|$ over $\mathbf{K}$. In contrast, our method uses sparse semidefinite relaxations for polynomial optimization (derived from [179]) to bound $\ell$ and basic interval arithmetic as in [273] to bound $|h|$ (i.e. we use interval arithmetic to bound second-order error terms in the multivariate Taylor expansion of $\Delta$ w.r.t. $\mathbf{e}$).

- A direct attempt to solve the two polynomial problems occurring in Equation (3.13) fails as the SDP solver (in our case SDPA [310]) runs out of memory.
- Using our method implemented in the `Real2Float` tool[1], one obtains an upper bound of $760\varepsilon$ for $|\ell| + |h|$ over $\mathbf{K}$ in 0.15 seconds. This bound is provided together with a certificate which can be formally checked inside the COQ proof assistant in 0.20 seconds.
- After normalizing the polynomial expression and using basic interval arithmetic, one obtains 8 times more quickly a coarser bound of $922\varepsilon$.
- Symbolic Taylor expansions implemented in `FPTaylor` [273] provide a more precise bound of $721\varepsilon$, but the analysis time is 28 times slower than with our implementation. Formal verification of this bound inside the HOL-LIGHT proof assistant takes 27.7 seconds, which is 139 times slower than proof checking with `Real2Float` inside COQ. One can obtain an even more precise bound of $528\varepsilon$ (but 37 times slower than with our implementation) by turning on the improved rounding model of `FPTaylor` and limiting the number of branch and bound iterations to 10000. The drawback of this bound is that it cannot be formally verified.
- Finally, a slightly coarser bound of $762\varepsilon$ is obtained with the ROSA real compiler [72], but the analysis is 19 times slower than with our implementation and we cannot get formal verification of this bound.

We refer the interested reader to [J14, § 4] for more details on the extensive experimental evaluation that we performed.

## 3.3 Application to noncommutative polynomials

Here, we handle a specific class of sparse POP with noncommuting variables. This section outlines the main results published in [J3]. In this context, a given noncommutative polynomial in $n$ variables and degree $d$ is positive semidefinite if and only if it decomposes as an SOHS [123, 212]. In

[1] https://forge.ocamlcore.org/projects/nl-certify/

practice, an SOHS decomposition can be computed by solving an SDP of size $O(n^d)$, which is even larger than the size of the matrices involved in the commutative case. SOHS decompositions are also used for constrained optimization, either to minimize eigenvalues or traces of noncommutative polynomial objective functions, under noncommutative polynomial (in)equality constraints. The optimal value of such constrained problems can be approximated, as closely as desired, while relying on the noncommutative analogue of Lasserre's hierarchy [238, 57, 54]. The `NCSOStools` [58, 53] library can compute such approximations for optimization problems involving polynomials in noncommuting variables. By comparison with the commutative case, the size $O(n^r)$ of the SDP matrices at a given step $r$ of the noncommutative hierarchy becomes intractable even faster, typically for $r, n \simeq 6$ on a standard laptop.

## Existing approaches

A remedy for unconstrained problems is to rely on the adequate noncommutative analogue of the standard Newton polytope method, which is called the *Newton chip method* (see, e.g., [53, §2.3]) and can be further improved with the *augmented Newton chip method* (see, e.g., [53, §2.4]), by removing certain terms which can never appear in an SOHS decomposition of a given input. As in the commutative case, the Newton polytope method cannot be applied for constrained problems. When one cannot go from step $r$ to step $r+1$ in the hierarchy because of the computational burden, one can always consider matrices indexed by all terms of degree $r$ plus a fixed percentage of terms of degree $r+1$. This is used for instance to compute tighter upper bounds for maximum violation levels of Bell inequalities [232]. Another trick, implemented in the NCPOL2SDPA library [306], consists of exploiting simple equality constraints, such as "$x^2 = 1$", to derive substitution rules for variables involved in the SDP relaxations. Similar substitutions are performed in the commutative case by GLOPTIPOLY [132].

Apart from such heuristic procedures, there is, to the best of our knowledge, no general method to exploit additional structure, such as sparsity, of (un)constrained noncommutative POP.

## Noncommutative polynomials

We use similar notation to Section 1.5, where we consider a finite alphabet $x_1, \dots, x_n$ and generate all possible words of finite length in these letters. The *degree* of a noncommutative (nc) polynomial $f \in \mathbb{R}\langle \underline{x} \rangle$ is the length of the longest word involved in $f$. For $r \in \mathbb{N}$, $\langle \underline{x} \rangle_r$ is the set of all words of degree at most $r$. Let us denote by $\mathbf{W}_r$ the vector of all words of $\langle \underline{x} \rangle_r$ w.r.t. to the lexicographic order. Note that the dimension of $\mathbb{R}\langle \underline{x} \rangle_r$ is the length of $\mathbf{W}_r$, which is $\sigma(n,r) := \sum_{i=0}^r n^i = \frac{n^{r+1}-1}{n-1}$. The set of all *symmetric elements* is defined as $\operatorname{Sym} \mathbb{R}\langle \underline{x} \rangle := \{ f \in \mathbb{R}\langle \underline{x} \rangle : f = f^\star \}$. An nc polynomial of the form $g^\star g$ is called a *hermitian square*. A given $f \in \mathbb{R}\langle \underline{x} \rangle$ is an SOHS if there exist nc polynomials $h_1, \dots, h_t \in \mathbb{R}\langle \underline{x} \rangle$, with $t \in \mathbb{N}$, such that $f = h_1^\star h_1 + \cdots + h_t^\star h_t$. Let $\Sigma\langle \underline{x} \rangle$ stands for the set of SOHS. We denote by $\Sigma\langle \underline{x} \rangle_r \subseteq \Sigma\langle \underline{x} \rangle$ the set of SOHS polynomials of degree at most $2r$. We now recall how to check whether a given $f \in \operatorname{Sym} \mathbb{R}\langle \underline{x} \rangle$ is an SOHS. The existing procedure, known as the *Gram matrix method*, relies on the following proposition (see, e.g., [123, §2.2]):

**Proposition 3.3.1** *Assume that $f \in \operatorname{Sym} \mathbb{R}\langle \underline{x} \rangle_{2d}$. Then $f \in \Sigma\langle \underline{x} \rangle$ if and only if there exists $\mathbf{G}_f \succeq 0$ satisfying*

$$f = \mathbf{W}_d^\star \, \mathbf{G}_f \, \mathbf{W}_d \,. \tag{3.15}$$

*Conversely, given such $\mathbf{G} \succeq 0$ with rank $t$, one can construct $g_1, \dots, g_t \in \mathbb{R}\langle \underline{x} \rangle_d$ such that $f = \sum_{i=1}^t g_i^\star g_i$.*

Any symmetric matrix $\mathbf{G}$ (not necessarily positive semidefinite) satisfying (3.15) is called a *Gram matrix* of $f$.

Given a positive integer $m$ and $S = \{g_1, \ldots, g_m\} \subseteq \operatorname{Sym} \mathbb{R}\langle \underline{x} \rangle$, the semialgebraic set $\mathcal{D}_S$ associated to $S$ is defined as follows:

$$\mathcal{D}_S := \bigcup_{k \in \mathbb{N}} \{\underline{A} = (A_1, \ldots, A_n) \in \mathbb{S}_k^n : g_j(\underline{A}) \succeq 0 \,, \quad j \in [m]\} \,. \tag{3.16}$$

When considering only tuples of $N \times N$ symmetric matrices, we use the notation $\mathcal{D}_S^N := \mathcal{D}_S \cap \mathbb{S}_N^n$. The operator semialgebraic set $\mathcal{D}_S^\infty$ is the set of all bounded self-adjoint operators $\underline{A}$ on a Hilbert space $\mathcal{H}$ endowed with a scalar product $\langle \cdot \mid \cdot \rangle$, making $g(\underline{A})$ a positive semidefinite operator for all $g \in S$, i.e., $\langle g(\underline{A}) v \mid v \rangle \geq 0$, for all $v \in \mathcal{H}$. We say that a noncommutative polynomial $f$ is positive (denoted by $f \succ 0$) on $\mathcal{D}_S^\infty$ if for all $\underline{A} \in \mathcal{D}_S^\infty$ the operator $f(\underline{A})$ is positive definite, i.e., $\langle f(\underline{A}) v \mid v \rangle > 0$, for all nonzero $v \in \mathcal{H}$. The quadratic module $\mathcal{M}(S)$, generated by $S$, is defined by

$$\mathcal{M}(S) := \left\{ \sum_{i=1}^K a_i^\star g_i a_i : K \in \mathbb{N} \,, a_i \in \mathbb{R}\langle \underline{x} \rangle \,, g_i \in S \cup \{1\} \right\} \,. \tag{3.17}$$

Given $r \in \mathbb{N}$, the truncated quadratic module $\mathcal{M}(S)_r$ of order $r$, generated by $S$, is

$$\mathcal{M}(S)_r := \left\{ \sum_{i=1}^K a_i^\star g_i a_i : K \in \mathbb{N} \,, a_i \in \mathbb{R}\langle \underline{x} \rangle \,, g_j \in S \cup \{1\} \,, \deg(a_i^\star g_i a_i) \leq 2r \right\} \,. \tag{3.18}$$

Let $\mathbf{1}$ stands for the unit polynomial. A quadratic module $\mathcal{M}$ is called *archimedean* if for each $a \in \mathbb{R}\langle \underline{x} \rangle$, there exists $N \in \mathbb{R}_{\geq 0}$ such that $N \cdot \mathbf{1} - a^\star a \in \mathcal{M}$. One can show that this is equivalent to the existence of an $N \in \mathbb{R}_{\geq 0}$ such that $N \cdot \mathbf{1} - \sum_{i=1}^n x_i^2 \in \mathcal{M}$.

The noncommutative analog of Putinar's Positivstellensatz [245] describing noncommutative polynomials positive on $\mathcal{D}_S^\infty$ with archimedean $\mathcal{M}(S)$ is due to Helton and McCullough:

**Theorem 3.3.2 ([126, Theorem 1.2])** *Let $S \cup \{f\} \subseteq \operatorname{Sym} \mathbb{R}\langle \underline{x} \rangle$ and assume that $\mathcal{M}(S)$ is archimedean. If $f(\underline{A}) \succ 0$ for all $\underline{A} \in \mathcal{D}_S^\infty$, then $f \in \mathcal{M}(S)$.*

## Sparsity patterns

We adapt the concepts of CS from Section 3.1 in the nc setting. With $[n] := \{1, \ldots, n\}$ and $p \in \mathbb{N}$ consider $I_1, \ldots, I_p \subseteq [n]$ satisfying $\bigcup_{l=1}^p I_l = [n]$. Let $n_l$ be the size of $I_l$, for each $l \in [p]$. We denote by $\langle \underline{x}(I_l) \rangle$ (resp. $\mathbb{R}\langle \underline{x}, I_l \rangle$) the set of words (resp. nc polynomials) in the $n_l$ variables $\underline{x}(I_l) = \{x_i : i \in I_l\}$. The dimension of $\mathbb{R}\langle \underline{x}, I_l \rangle_r$ is $\sigma(n_l, r) = \frac{n_l^{r+1} - 1}{n_l - 1}$. Note that $\mathbb{R}\langle \underline{x}, [n] \rangle = \mathbb{R}\langle \underline{x} \rangle$. We also define $\operatorname{Sym} \mathbb{R}\langle \underline{x}, I_l \rangle := \operatorname{Sym} \mathbb{R}\langle \underline{x} \rangle \cap \mathbb{R}\langle \underline{x}, I_l \rangle$, let $\Sigma\langle \underline{x}, I_l \rangle$ stands for the set of SOHS in $\mathbb{R}\langle \underline{x}, I_l \rangle$ and we denote by $\Sigma\langle \underline{x}, I_l \rangle_r$ the restriction of $\Sigma\langle \underline{x}, I_l \rangle$ to nc polynomials of degree at most $2r$. In the sequel, we will rely on two specific assumptions. The first one is as follows:

**Assumption 3.3.3 (Boundedness)** *Let $\mathcal{D}_S$ be as in (3.16). There is $N \in \mathbb{R}_{>0}$ such that $\sum_{i=1}^n x_i^2 \preceq N \cdot \mathbf{1}$, for all $\underline{x} \in \mathcal{D}_S^\infty$.*

Then, Assumption 3.3.3 implies that $\sum_{j \in I_l} x_j^2 \preceq N \cdot \mathbf{1}$, for all $l \in [p]$. Thus we define

$$g_{m+k} := N \cdot \mathbf{1} - \sum_{j \in I_l} X_j^2 \,, \quad l \in [p] \,, \tag{3.19}$$

and set $m' = m + p$ in order to describe the same set $\mathcal{D}_S$ again as:

$$\mathcal{D}_S := \bigcup_{k \in \mathbb{N}} \{\underline{A} \in \mathbb{S}_k^n : g_j(\underline{A}) \succeq 0, \quad j \in [m']\} \,, \tag{3.20}$$

as well as the operator semialgebraic set $\mathcal{D}_S^\infty$.

The second assumption is as follows:

**Assumption 3.3.4 (RIP)** *Let $\mathcal{D}_S$ be as in (3.20) and let $f \in \mathbb{R}\langle \underline{x} \rangle$. The index set $J := \{1, \ldots, m'\}$ is partitioned into p disjoint sets $J_1, \ldots, J_p$ and the two collections $\{I_1, \ldots, I_p\}$ and $\{J_1, \ldots, J_p\}$ satisfy:*

(i) *For all $j \in J_l$, $g_j \in \operatorname{Sym} \mathbb{R}\langle \underline{x}, I_l \rangle$;*

(ii) *The objective function can be decomposed as $f = f_1 + \cdots + f_p$, with $f_l \in \mathbb{R}\langle \underline{x}, I_l \rangle$, for all $l \in [p]$;*

(iii) *The running intersection property (RIP) (3.6) holds.*

Even though we assume that $I_1, \ldots, I_p$ are explicitly given, one can compute such subsets using the procedure in [300]. Roughly speaking, this procedure consists of two steps. The first step provides the csp graph of the variables involved in the input polynomial data. The second step computes the maximal cliques of a chordal extension of this csp graph. Even if the computation of all maximal cliques of a graph is an NP hard problem in general, it turns out that this procedure is efficient in practice, due to the properties of chordal graphs (see, e.g., [42] for more details on the properties of chordal graphs).

## Hankel and localizing matrices

To $g \in \operatorname{Sym} \mathbb{R}\langle \underline{x} \rangle$ and a linear functional $L : \mathbb{R}\langle \underline{x} \rangle_{2r} \to \mathbb{R}$, one associates the following two matrices:

(1) the *noncommutative Hankel matrix* $\mathbf{M}_r(L)$ is the matrix indexed by words $u, v \in \langle \underline{x} \rangle_r$, with $(\mathbf{M}_d(L))_{u,v} = L(u^\star v)$;

(2) the *localizing matrix* $\mathbf{M}_{r-\lceil \deg g/2 \rceil}(gL)$ is the matrix indexed by words $u, v \in \langle \underline{X} \rangle_{r-\lceil \deg g/2 \rceil}$, with $(\mathbf{M}_{r-\lceil \deg g/2 \rceil}(gL))_{u,v} = L(u^\star g v)$.

The functional $L$ is called *unital* if $L(1) = 1$ and is called *symmetric* if $L(f^\star) = L(f)$, for all $f$ belonging to the domain of $L$. We also recall the following useful facts.

**Lemma 3.3.5 ([53, Lemma 1.44])** *Let $g \in \operatorname{Sym} \mathbb{R}\langle \underline{x} \rangle$ and let $L : \mathbb{R}\langle \underline{x} \rangle_{2r} \to \mathbb{R}$ be a symmetric linear functional. Then, one has:*

(1) *$L(h^\star h) \geq 0$ for all $h \in \mathbb{R}\langle \underline{x} \rangle_r$, if and only if, $\mathbf{M}_d(L) \succeq 0$;*

(2) *$L(h^\star g h) \geq 0$ for all $h \in \mathbb{R}\langle \underline{x} \rangle_{r-\lceil \deg g/2 \rceil}$, if and only if, $\mathbf{M}_{r-\lceil \deg g/2 \rceil}(gL) \succeq 0$.*

**Definition 3.3.6** *Suppose $L : \mathbb{R}\langle \underline{x} \rangle_{2r+2\delta} \to \mathbb{R}$ is a linear functional with restriction $\tilde{L} : \mathbb{R}\langle \underline{x} \rangle_{2r} \to \mathbb{R}$. We associate to L and $\tilde{L}$ the Hankel matrices $\mathbf{M}_{r+\delta}(L)$ and $\mathbf{M}_r(\tilde{L})$ respectively, and get the block form*

$$\mathbf{M}_{r+\delta}(L) = \begin{bmatrix} \mathbf{M}_r(\tilde{L}) & B \\ B^T & C \end{bmatrix}.$$

*We say that L is $\delta$-*flat *or that L is a* flat extension *of $\tilde{L}$, if $\mathbf{M}_{r+\delta}(L)$ is flat over $\mathbf{M}_r(\tilde{L})$, i.e., if* $\operatorname{rank} \mathbf{M}_{r+\delta}(L) = \operatorname{rank} \mathbf{M}_r(\tilde{L})$.

For a subset $I \subseteq [p]$, let us define $\mathbf{M}_r(L, I)$ to be the Hankel submatrix obtained from $\mathbf{M}_r(L)$ after retaining only those rows and columns indexed by $w \in \langle \underline{X}(I) \rangle_r$. When $I \subseteq I_l$ and $g \in \mathbb{R}\langle \underline{x}, I_l \rangle$, for $l \in [p]$ , we define the localizing submatrix $\mathbf{M}_{r-\lceil \deg g/2 \rceil}(gL, I)$ in a similar fashion. In particular, $\mathbf{M}_r(L, I_l)$ and $\mathbf{M}_{r-\lceil \deg g/2 \rceil}(gL, I_l)$ can be seen as Hankel and localizing matrices with rows and columns indexed by a basis of $\mathbb{R}\langle \underline{x}, I_l \rangle_r$ and $\mathbb{R}\langle \underline{x}, I_l \rangle_{r-\lceil \deg g/2 \rceil}$, respectively.

$$\begin{tikzcd}
& (\mathcal{C},\varphi) & \\
(\mathcal{B}_1,\varphi_1) \arrow[ur, dashed, "j_1"] & & (\mathcal{B}_2,\varphi_2) \arrow[ul, dashed, "j_2"'] \\
& (\mathcal{A},\varphi_0) \arrow[ul, "\iota_1"] \arrow[ur, "\iota_2"'] &
\end{tikzcd}$$

Figure 3.4: Illustration of Theorem 3.3.7 in the case $I = \{1,2\}$.

## Sparse representations of nc positive polynomials

Here, we state our main theoretical result, which is a sparse version of the Helton-McCullough archimedean Positivstellensatz (Theorem 3.3.2). For this, we rely on amalgamation theory for $C^\star$-algebras, see, e.g., [41, 295].

Given a Hilbert space $\mathcal{H}$, we denote by $\mathcal{B}(\mathcal{H})$ the set of bounded operators on $\mathcal{H}$. A $C^\star$-algebra is a complex Banach algebra $\mathcal{A}$ thus also a Banach space), endowed with a norm $\|\cdot\|$, and with an involution $\star$ satisfying $\|xx^\star\| = \|x\|^2$ for all $x \in \mathcal{A}$. Equivalently, it is a norm closed subalgebra with involution of $\mathcal{B}(\mathcal{H})$ for some Hilbert space $\mathcal{H}$. Given a $C^\star$-algebra $\mathcal{A}$, a *state* $\varphi$ is defined to be a positive linear functional of unit norm on $\mathcal{A}$, and we write often $(\mathcal{A},\varphi)$ when $\mathcal{A}$ comes together with the state $\varphi$. Given two $C^\star$-algebras $(\mathcal{A}_1,\varphi_1)$ and $(\mathcal{A}_2,\varphi_2)$, a homomorphism $\iota : \mathcal{A}_1 \to \mathcal{A}_2$ is called *state-preserving* if $\varphi_2 \circ \iota = \varphi_1$. Given a $C^\star$-algebra $\mathcal{A}$, a *unitary representation* of $\mathcal{A}$ in $\mathcal{H}$ is a $*$-homomorphism $\pi : \mathcal{A} \to \mathcal{B}(\mathcal{H})$ which is *strongly continuous*, i.e., the mapping $\mathcal{A} \to \mathcal{H}, g \mapsto \pi(g)\xi$ is continuous for every $\xi \in \mathcal{H}$.

**Theorem 3.3.7 ([41] or [295, Section 5])** *Let $(\mathcal{A},\varphi_0)$ and $\{(\mathcal{B}_k,\varphi_k) : k \in I\}$ be $C^\star$-algebras with states, and let $\iota_k$ be a state-preserving embedding of $\mathcal{A}$ into $\mathcal{B}_k$, for each $k \in I$. Then there exists a $C^\star$-algebra $\mathcal{C}$ amalgamating the $(\mathcal{B}_k,\varphi_k)$ over $(\mathcal{A},\varphi_0)$. That is, there is a state $\varphi$ on $\mathcal{C}$, and state-preserving homomorphisms $j_k : \mathcal{B}_k \to \mathcal{C}$, such that $j_k \circ \iota_k = j_i \circ \iota_i$, for all $k, i \in I$, and such that $\bigcup_{k\in I} j_k(\mathcal{B}_k)$ generates $\mathcal{C}$.*

Theorem 3.3.7 is illustrated in Figure 3.4 in the case $I = \{1,2\}$. We also recall the GNS construction establishing a correspondence between $\star$-representations of a $C^\star$-algebra and positive linear functionals on it. In our context, the next result [53, Theorem 1.27] restricts to linear functionals on $\mathbb{R}\langle \underline{x} \rangle$ which are positive on an archimedean quadratic module.

**Theorem 3.3.8** *Let $S \subseteq \operatorname{Sym}\mathbb{R}\langle \underline{x} \rangle$ be given such that its quadratic module $\mathcal{M}(S)$ is archimedean. Let $L : \mathbb{R}\langle \underline{x} \rangle \to \mathbb{R}$ be a nontrivial linear functional with $L(\mathcal{M}(S)) \subseteq \mathbb{R}_{\geq 0}$. Then there exists a tuple $\underline{A} = (A_1,\dots,A_n) \in \mathcal{D}_S^\infty$ and a vector $\mathbf{v}$ such that $L(f) = \langle f(\underline{A})\mathbf{v},\mathbf{v}\rangle$, for all $f \in \mathbb{R}\langle \underline{x} \rangle$.*

For $l \in [p]$, let us define

$$\mathcal{M}(S)^l := \left\{ \sum_{i=1}^{K} a_i^\star g_i a_i : K \in \mathbb{N}\,, a_i \in \mathbb{R}\langle \underline{x}, I_l \rangle,\, g_i \in (S \cap \operatorname{Sym}\mathbb{R}\langle \underline{x}, I_l \rangle) \cup \{1\} \right\},$$

and

$$\mathcal{M}(S)^{\mathrm{cs}} := \mathcal{M}(S)^1 + \cdots + \mathcal{M}(S)^p\,. \tag{3.21}$$

Next, we state the main foundational result of this section.

**Theorem 3.3.1** *Let $S \cup \{f\} \subseteq \operatorname{Sym} \mathbb{R}\langle \underline{x} \rangle$ and let $\mathcal{D}_S$ be as in (3.20) with the additional quadratic constraints (3.19). Suppose Assumption 3.3.4 holds. If $f(\underline{A}) \succ 0$ for all $\underline{A} \in \mathcal{D}_S^\infty$, then $f \in \mathcal{M}(S)^{\mathrm{cs}}$.*

We provide an example demonstrating that sparsity without a RIP-type condition is not sufficient to deduce sparsity in SOHS decompositions.

**Example 3.3.1** Consider the case of three variables $\underline{x} = (x_1, x_2, x_3)$ and the polynomial

$$f \;=\; (x_1 + x_2 + x_3)^2 \;=\; x_1^2 + x_2^2 + x_3^2 + x_1x_2 + x_2x_1 + x_1x_3 + x_3x_1 + x_2x_3 + x_3x_2 \;\in\; \Sigma\langle \underline{x} \rangle.$$

Then $f = f_1 + f_2 + f_3$, with

$$\begin{aligned} f_1 &= \frac{1}{2}x_1^2 + \frac{1}{2}x_2^2 + x_1x_2 + x_2x_1 \in \mathbb{R}\langle x_1, x_2 \rangle, \\ f_2 &= \frac{1}{2}x_2^2 + \frac{1}{2}x_3^2 + x_2x_3 + x_3x_2 \in \mathbb{R}\langle x_2, x_3 \rangle, \\ f_3 &= \frac{1}{2}x_1^2 + \frac{1}{2}x_3^2 + x_1x_3 + x_3x_1 \in \mathbb{R}\langle x_1, x_3 \rangle. \end{aligned}$$

However, the sets $I_1 = \{1, 2\}$, $I_2 = \{2, 3\}$ and $I_3 = \{1, 3\}$ do not satisfy the RIP condition (3.6) and $f \notin \Sigma\langle \underline{x} \rangle^{\mathrm{cs}} := \Sigma\langle x_1, x_2 \rangle + \Sigma\langle x_2, x_3 \rangle + \Sigma\langle x_1, x_3 \rangle$ since it has a unique Gram matrix by homogeneity.

Now consider $S = \{1 - x_1^2, 1 - x_2^2, 1 - x_3^2\}$. Then $\mathcal{D}_S$ is as in (3.20), $\mathcal{M}(S)^{\mathrm{cs}}$ is as in (3.21) and $f|_{\mathcal{D}_S^\infty} \succeq 0$. However, we claim that $f - b \in \mathcal{M}(S)^{\mathrm{cs}}$ iff $b \leq -3$. Clearly,

$$f + 3 = (x_1 + x_2)^2 + (x_1 + x_3)^2 + (x_2 + x_3)^2 + (1 - x_1^2) + (1 - x_2^2) + (1 - x_3^2) \in \mathcal{M}(S)^{\mathrm{cs}}.$$

So one has $-3 \leq \sup\{b : f - b \in \mathcal{M}(S)^{\mathrm{cs}}\}$, and the dual of this latter problem is given by

$$\begin{aligned} \inf_{L_l} \quad & \sum_{l=1}^{3} L_l(f_l) \\ \text{s.t.} \quad & L_l(1) = 1\,, \quad l = 1,2,3\,, \\ & L_l(h^\star h) \succeq 0 \quad \forall h \in \mathbb{R}\langle \underline{x}, I_l \rangle\,, \quad l = 1,2,3\,, \\ & L_l(h^\star(1 - x_l^2)h) \succeq 0 \quad \forall h \in \mathbb{R}\langle \underline{x}, I_l \rangle\,, \quad l = 1,2,3\,, \\ & L_j|_{\mathbb{R}\langle \underline{X}(I_j \cap I_l) \rangle} = L_l|_{\mathbb{R}\langle \underline{X}(I_j \cap I_l) \rangle}\,, \quad j,l = 1,2,3\,. \end{aligned} \tag{3.22}$$

Hence, by weak duality, it suffices to show that there exist linear functionals $L_l : \mathbb{R}\langle \underline{x}, I_l \rangle \to \mathbb{R}$ satisfying the constraints of problem (3.22) and such that $\sum_l L_l(f_l) = -3$. Define

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad B = -A$$

and let

$$L_l(g) = \operatorname{tr}(g(A, B)) \quad \text{for } g \in \mathbb{R}\langle \underline{x}, I_l \rangle.$$

Since $L_l(f_l) = -1$, the three first constraints of problem (3.22) are easily verified and $\sum_l L_l(f_l) = -3$. For the last one, given, say $h \in \mathbb{R}\langle \underline{x}, I_1 \rangle \cap \mathbb{R}\langle \underline{x}, I_2 \rangle = \mathbb{R}\langle x_2 \rangle$, we have

$$\begin{aligned} L_1(h) &= \operatorname{tr}(h(B)), \\ L_2(h) &= \operatorname{tr}(h(A)), \end{aligned}$$

since $L_1$ (resp. $L_2$) is defined on $\mathbb{R}\langle x_1, x_2\rangle$ (resp. $\mathbb{R}\langle x_2, x_3\rangle$) and $h$ depends only on the second (resp. first) variable $x_2$ corresponding to $B$ (resp. $A$).

But matrices $A$ and $B$ are orthogonally equivalent as $UAU^T = B$ for

$$U = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix},$$

whence $h(B) = h(UAU^T) = Uh(A)U^T$ and $h(A)$ have the same trace.

## Sparse GNS construction

Next, we provide the main theoretical tools to extract solutions of sparse noncommutative optimization problems. For this purpose, we first present sparse noncommutative versions of theorems by Curto and Fialkow, derived in the context of commutative polynomials [71], eigenvalue optimization of noncommutative polynomials [212, Lemma 2.2] (see also [238], [9, Chapter 21] and [53, Theorem 1.69]), and trace optimization of noncommutative polynomials [54] We recall this theorem, which relies on a finite-dimensional GNS construction.

**Theorem 3.3.9** *Let $S \subseteq \operatorname{Sym}\mathbb{R}\langle \underline{x}\rangle$ and set $\delta := \max\{\lceil \deg(g)/2\rceil : g \in S \cup \{1\}\}$. For $r \in \mathbb{N}$, let $L : \mathbb{R}\langle \underline{x}\rangle_{2r+2\delta} \to \mathbb{R}$ be a unital linear functional satisfying $L(\mathcal{M}(S)_{r+\delta}) \subseteq \mathbb{R}_{\geq 0}$. If $L$ is $\delta$-flat, then there exist $\hat{A} \in \mathcal{D}_S^r$ for some $t \leq \boldsymbol{\sigma}(n, r)$ and a unit vector $\mathbf{v}$ such that*

$$L(g) = \langle g(\hat{A})\mathbf{v}, \mathbf{v}\rangle\,, \tag{3.23}$$

*for all $g \in \operatorname{Sym}\mathbb{R}\langle \underline{x}\rangle_{2r}$.*

We now give the sparse version of Theorem 3.3.9.

**Theorem 3.3.2** *Let $S \subseteq \operatorname{Sym}\mathbb{R}\langle \underline{x}\rangle_{2r}$, and assume $\mathcal{D}_S$ is as in (3.20) with the additional quadratic constraints (3.19). Suppose Assumption 3.3.4(i) holds. Set $\delta := \max\{\lceil \deg(g)/2\rceil : g \in S \cup \{1\}\}$. Let $L : \mathbb{R}\langle \underline{x}\rangle_{2r+2\delta} \to \mathbb{R}$ be a unital linear functional satisfying $L(\mathcal{M}(S)_r^{\mathrm{cs}}) \subseteq \mathbb{R}_{\geq 0}$. Assume that the following holds:*

*(H1) $\mathbf{M}_{r+\delta}(L, I_l)$ and $\mathbf{M}_{r+\delta}(L, I_l \cap I_j)$ are $\delta$-flat, for all $j, l \in [p]$.*

*Then, there exist finite-dimensional Hilbert spaces $\mathcal{H}(I_l)$ with dimension $t_l$, for all $l \in [p]$, Hilbert spaces $\mathcal{H}(I_j \cap I_l) \subseteq \mathcal{H}(I_j), \mathcal{H}(I_l)$ for all pairs $(j, l)$ with $I_j \cap I_l \neq 0$, and operators $\hat{A}^l$, $\hat{A}^{jl}$, acting on them, respectively. Further, there are unit vectors $\mathbf{v}^j \in \mathcal{H}(I_j)$ and $\mathbf{v}^{jl} \in \mathcal{H}(I_j \cap I_l)$ such that*

$$\begin{aligned} L(f) &= \langle f(\hat{A}^j)\mathbf{v}^j, \mathbf{v}^j\rangle \quad \text{for all } f \in \mathbb{R}\langle \underline{x}, I_j\rangle_{2r}, \\ L(g) &= \langle g(\hat{A}^{jl})\mathbf{v}^{jl}, \mathbf{v}^{jl}\rangle \quad \text{for all } g \in \mathbb{R}\langle \underline{X}(I_j \cap I_l)\rangle_{2r}. \end{aligned} \tag{3.24}$$

*Assuming that for all pairs $(j, l)$ with $I_j \cap I_l \neq \emptyset$, one has*

*(H2) the matrices $(\hat{A}_i^{jl})_{i \in I_j \cap I_l}$ have no common complex invariant subspaces,*

*then there exist $\underline{A} \in \mathcal{D}_S^t$, with $t := t_1 \cdots t_p$, and a unit vector $\mathbf{v}$ such that*

$$L(f) = \langle f(\underline{A})\mathbf{v}, \mathbf{v}\rangle\,, \tag{3.25}$$

*for all $f \in \sum_l \mathbb{R}\langle \underline{x}, I_l\rangle_{2r}$.*

$$\begin{tikzcd}
 & \mathcal{A} & \\
\mathcal{A}(I_1) \arrow[ur, dashed, "j_1"] & & \mathcal{A}(I_2) \arrow[ul, dashed, "j_2"'] \\
 & \mathcal{A}(I_1 \cap I_2) \arrow[ul, "\iota_1"] \arrow[ur, "\iota_2"'] &
\end{tikzcd}$$

Figure 3.5: Amalgamation of finite-dimensional $C^\star$-algebras

**Example 3.3.2 (Non-amalgamation in the category of finite-dimensional algebras)** *Given $I_1$ and $I_2$, suppose $\mathcal{A}(I_1 \cap I_2)$ is generated by the $2 \times 2$ diagonal matrix*

$$A^{12} = \begin{pmatrix} 1 & \\ & 2 \end{pmatrix},$$

*and assume $\mathcal{A}(I_1) = \mathcal{A}(I_2) = \mathbb{M}_3(\mathbb{R})$. (Observe that $\mathcal{A}(I_1 \cap I_2)$ is the algebra of all diagonal matrices.) For each $l \in \{1, 2\}$, let us define $\iota_l(A) := A \oplus l$, for all $A \in \mathcal{A}(I_1 \cap I_2)$. We claim that there is no finite-dimensional $C^\star$-algebra $\mathcal{A}$ amalgamating the above Figure 3.5. Indeed, by the Skolem-Noether theorem, every homomorphism $\mathbb{M}_n(\mathbb{R}) \to \mathbb{M}_m(\mathbb{R})$ is of the form $x \mapsto P^{-1}(x \otimes \mathrm{I}_{m/n})P$ for some invertible $P$; in particular, $n$ divides $m$. If a desired $\mathcal{A}$ existed, then the matrices $(A^{12} \oplus 1) \otimes \mathrm{I}_l$ and $(A^{12} \oplus 2) \otimes \mathrm{I}_l$ would be similar. But they are not as is easily seen from eigenvalue multiplicities.*

**Remark 3.3.1** *Theorem 3.3.2 can be seen as a noncommutative variant of the result by Lasserre stated in [179, Theorem 3.7], related to the minimizers extraction in the context of sparse polynomial optimization. In the sparse commutative case, Lasserre assumes flatness of each moment matrix indexed in the canonical basis of $\mathbb{R}[\underline{X}, I_l]_r$, for each $l \in [p]$, which is similar to our flatness condition (H1). The difference is that this technical flatness condition on each $I_l$ adapts to the degree of the constraints polynomials on variables in $I_l$, resulting in an adapted parameter $\delta_l$ instead of global $\delta$. We could assume the same in Theorem 3.3.2 but for the sake of simplicity, we assume that these parameters are all equal. In addition, Lasserre assumes that each moment matrix indexed in the canonical basis of $\mathbb{R}[\underline{x}, I_j \cap I_l)]_r$ is rank one, for all pairs $(j, l)$ with $I_j \cap I_l \neq \emptyset$, which is the commutative analog of our irreducibility condition (H2).*

As in the dense case, we can summarize the sparse GNS construction procedure described in the proof of Theorem 3.3.2 into an algorithm, called `SparseGNS`, stated in [J3, Algorithm 4.6], for the case $p = 2$ (the general case is similar).

## Eigenvalue optimization of nc sparse polynomials

We provide SDP relaxations allowing one to under-approximate the smallest eigenvalue that a given nc polynomial can attain on a tuple of symmetric matrices from a given semialgebraic set. We first recall the celebrated Helton-McCullough SOS theorem [123, 212] stating the equivalence between SOHS and positive semidefinite nc polynomials.

**Theorem 3.3.10** *Given $f \in \mathbb{R}\langle \underline{x} \rangle$, we have $f(\underline{A}) \succeq 0$, for all $\underline{A} \in \mathbb{S}^n$, if and only if $f \in \Sigma\langle \underline{x} \rangle$.*

In contrast with the constrained case where we obtain the analog of Putinar's Positivstellensatz in Theorem 3.3.1, there is no sparse analog of Theorem 3.3.10, as shown in the following example.

**Lemma 3.3.11** *There exist polynomials which are sparse sums of hermitian squares but are not sums of sparse hermitian squares.*

PROOF Let $v = \begin{bmatrix} x_1 & x_1x_2 & x_2 & x_3 & x_3x_2 \end{bmatrix}$,

$$\mathbf{G}_f = \begin{bmatrix} 1 & -1 & -1 & 0 & \alpha \\ -1 & 2 & 0 & -\alpha & 0 \\ -1 & 0 & 3 & -1 & 9 \\ 0 & -\alpha & -1 & 6 & -27 \\ \alpha & 0 & 9 & -27 & 142 \end{bmatrix}, \qquad \alpha \in \mathbb{R}, \tag{3.26}$$

and consider

$$\begin{aligned} f &= v\mathbf{G}_f v^\star \\ &= x_1^2 - x_1x_2 - x_2x_1 + 3x_2^2 - 2x_1x_2x_1 + 2x_1x_2^2x_1 \\ &\quad - x_2x_3 - x_3x_2 + 6x_3^2 + 9x_2^2x_3 + 9x_3x_2^2 - 54x_3x_2x_3 + 142x_3x_2^2x_3. \end{aligned} \tag{3.27}$$

The polynomial $f$ is clearly sparse w.r.t. $I_1 = \{x_1, x_2\}$ and $I_2 = \{x_2, x_3\}$. Note that the matrix $G$ is positive semidefinite if and only if $0.270615 \lesssim \alpha \lesssim 1.1075$, whence $f$ is a sparse polynomial that is an SOHS.

We claim that $f \notin \Sigma\langle \underline{x}, I_1\rangle + \Sigma\langle \underline{x}, I_2\rangle$, i.e., $f$ is not a sum of sparse hermitian squares. By the Newton chip method [53, Section 2.3] only monomials in $v$ can appear in a sum of squares decomposition of $f$. Further, every Gram matrix of $f$ (with border vector $v$) is of the form (3.26). However, the matrix $\mathbf{G}_f$ with $\alpha = 0$ is not positive semidefinite, hence $f \notin \Sigma\langle \underline{x}, I_1\rangle + \Sigma\langle \underline{x}, I_2\rangle$.

## Unconstrained eigenvalue optimization with correlative sparsity

Let I stands for the identity matrix. Given $f \in \operatorname{Sym}\mathbb{R}\langle \underline{x}\rangle$ of degree $2d$, the smallest eigenvalue of $f$ is obtained by solving the following optimization problem

$$\lambda_{\min}(f) := \inf\{\langle f(\underline{A})\mathbf{v}, \mathbf{v}\rangle : \underline{A} \in \mathbb{S}^n, \|\mathbf{v}\| = 1\}\,. \tag{3.28}$$

The optimal value $\lambda_{\min}(f)$ of Problem (3.28) is the greatest lower bound on the eigenvalues of $f(\underline{A})$ over all $n$-tuples $\underline{A}$ of real symmetric matrices. Problem (3.28) can be rewritten as follows:

$$\begin{aligned} \lambda_{\min}(f) = \sup_b \quad & b \\ \text{s.t.} \quad & f(\underline{A}) - b\,\mathrm{I} \succeq 0\,, \quad \forall \underline{A} \in \mathbb{S}^n\,, \end{aligned} \tag{3.29}$$

which is in turn equivalent to

$$\begin{aligned} \lambda^d_{\min}(f) = \sup_b \quad & b \\ \text{s.t.} \quad & f(\underline{x}) - b \in \Sigma\langle \underline{x}\rangle_d\,, \end{aligned} \tag{3.30}$$

as a consequence of Theorem 3.3.10.

The dual of SDP (3.30) is

$$\begin{aligned} L^d_{\mathrm{sohs}}(f) = \inf_L \quad & \langle \mathbf{M}_d(L), \mathbf{G}_f\rangle \\ \text{s.t.} \quad & L(1) = 1\,, \quad \mathbf{M}_d(L) \succeq 0\,, \\ & L : \mathbb{R}\langle \underline{x}\rangle_{2d} \to \mathbb{R} \text{ linear}\,, \end{aligned} \tag{3.31}$$

where $\mathbf{G}_f$ is a Gram matrix for $f$ (see Proposition 3.3.1).
One can compute $\lambda_{\min}(f)$ by solving a single SDP, either SDP (3.31) or SDP (3.30), since there is no duality gap between these two programs (see, e.g., [53, Theorem 4.1]), that is, one has $L^d_{\text{sohs}}(f) = \lambda^d_{\min}(f) = \lambda_{\min}(f)$.

Now, we address eigenvalue optimization for a given sparse nc polynomial $f = f_1 + \cdots + f_p$ of degree $2d$, with $f_l \in \operatorname{Sym}\mathbb{R}\langle \underline{x}, I_l\rangle_{2d}$, for all $l \in [p]$. For all $l \in [p]$, let $\mathbf{G}_{f_l}$ be a Gram matrix associated to $f_l$. The sparse variant of SDP (3.31) is

$$\begin{aligned} L^d_{\text{cs}}(f) = \inf_{L} \quad & \sum_{l=1}^{p} \langle \mathbf{M}_d(L, I_l), G_{f_l}\rangle \\ \text{s.t.} \quad & L(1) = 1\,, \quad \mathbf{M}_d(L, I_l) \succeq 0\,, \quad l \in [p]\,, \\ & L : \mathbb{R}\langle \underline{x}, I_1\rangle_{2d} + \cdots + \mathbb{R}\langle \underline{x}, I_p\rangle_{2d} \to \mathbb{R} \text{ linear}\,, \end{aligned} \tag{3.32}$$

whose dual is the sparse variant of SDP (3.30):

$$\begin{aligned} \lambda^d_{\text{cs}}(f) = \sup_{b} \quad & b \\ \text{s.t.} \quad & f - b \in \Sigma\langle \underline{x}, I_1\rangle_{2d} + \cdots + \Sigma\langle \underline{x}, I_p\rangle_{2d}\,, \end{aligned} \tag{3.33}$$

To prove that there is no duality gap between SDP (3.32) and SDP (3.33), we need a sparse variant of [213, Proposition 3.4], which says that $\Sigma\langle \underline{x}\rangle_d$ is closed in $\mathbb{R}\langle \underline{x}\rangle_{2d}$:

**Proposition 3.3.12** *The set $\Sigma\langle \underline{x}\rangle^{\text{cs}}_d$ is a closed convex subset of $\mathbb{R}\langle \underline{x}, I_1\rangle_{2d} + \cdots + \mathbb{R}\langle \underline{x}, I_p\rangle_{2d}$.*

From Proposition 3.3.12, we obtain the following theorem which does not require Assumption 3.3.4.

**Theorem 3.3.3** *Let $f \in \operatorname{Sym}\mathbb{R}\langle \underline{x}\rangle$ of degree $2d$, with $f = f_1 + \cdots + f_p$, $f_k \in \operatorname{Sym}\mathbb{R}\langle \underline{x}, I_k\rangle_{2d}$, for all $k \in [p]$. Then, one has $\lambda^d_{\text{cs}}(f) = L^d_{\text{cs}}(f)$, i.e., there is no duality gap between SDP (3.32) and SDP (3.33).*

**Remark 3.3.2** *By contrast with the dense case, it is not enough to compute the solution of SDP (3.32) to obtain the optimal value $\lambda_{\min}(f)$ of the unconstrained optimization problem (3.28). However, one can still compute a certified lower bound $\lambda^d_{\text{cs}}(f)$ by solving a single SDP, either in the primal form (3.32) or in the dual form (3.33). Note that the related computational cost is potentially much less expensive. Indeed, SDP (3.33) involves $\sum_{l=1}^p \sigma(n_l, 2d)$ equality constraints and $\sum_{l=1}^p \sigma(n_l, d) + 1$ variables. This is in contrast with the dense version (3.30), which involves $\sigma(n, 2d)$ equality constraints and $1 + \sigma(n, d)$ variables.*

## Constrained eigenvalue optimization with correlative sparsity

Here, we focus on providing lower bounds for the constrained eigenvalue optimization of nc polynomials. Given $f \in \operatorname{Sym}\mathbb{R}\langle \underline{x}\rangle$ and $S := \{g_1, \dots, g_m\} \subset \operatorname{Sym}\mathbb{R}\langle \underline{x}\rangle$ as in (3.16), let us define $\lambda_{\min}(f, S)$ as follows:

$$\lambda_{\min}(f, S) := \inf\{\langle f(\underline{A})\mathbf{v}, \mathbf{v}\rangle : \underline{A} \in \mathcal{D}^{\infty}_S, \|\mathbf{v}\| = 1\}\,, \tag{3.34}$$

which is, as for the unconstrained case, equivalent to

$$\begin{aligned} \lambda_{\min}(f, S) = \sup_{b} \quad & b \\ \text{s.t.} \quad & f(\underline{A}) - b\,\mathrm{I} \succeq 0\,, \quad \forall \underline{A} \in \mathcal{D}^{\infty}_S\,. \end{aligned} \tag{3.35}$$

As usual, let $r_j := \lceil \deg g_j/2 \rceil$, for each $j \in [m]$ and $r_{\min} := \max\{\lceil \deg f/2 \rceil, r_1, \dots, r_m\}$. As shown in [238, 57] (see also [126]), one can approximate $\lambda_{\min}(f, S)$ from below via the following hierarchy of SDP programs, indexed by $r \geq r_{\min}$:

$$\begin{aligned} \lambda^r(f,S) := \sup_b \quad & b \\ \text{s.t.} \quad & f - b \in \mathcal{M}(S)_r \,. \end{aligned} \tag{3.36}$$

The dual of SDP (3.36) is

$$\begin{aligned} L^r(f,S) := \inf_L \quad & \langle \mathbf{M}_r(L), \mathbf{G}_f \rangle \\ \text{s.t.} \quad & L(1) = 1 \,, \\ & \mathbf{M}_r(L) \succeq 0 \,, \quad \mathbf{M}_{r-r_j}(g_j L) \succeq 0 \,, \quad j \in [m] \,, \\ & L : \mathbb{R}\langle \underline{x} \rangle_{2r} \to \mathbb{R} \text{ linear} \,, \end{aligned} \tag{3.37}$$

Under additional assumptions, this hierarchy of primal-dual SDP (3.36)-(3.37) converges to the value of the constrained eigenvalue problem.

**Corollary 3.3.13** *Assume that $\mathcal{D}_S$ is as in (3.20) with the additional quadratic constraints (3.19) and that the quadratic module $M_S$ is archimedean. Then the following holds for each $f \in \operatorname{Sym} \mathbb{R}\langle \underline{x} \rangle$:*

$$\lim_{r \to \infty} L^r(f,S) = \lim_{r \to \infty} \lambda^r(f,S) = \lambda_{\min}(f,S) \,. \tag{3.38}$$

The main ingredient of the proof (see, e.g., [53, Corollary 4.11]) is the nc analog of Putinar's Positivstellensatz, stated in Theorem 3.3.2.
Let $S \cup \{f\} \subseteq \operatorname{Sym} \mathbb{R}\langle \underline{x} \rangle$ and let $\mathcal{D}_S$ be as in (3.20) with the additional quadratic constraints (3.19). Let $\mathcal{M}(S)^{\text{cs}}$ be as in (3.21) and let us define $\mathcal{M}(S)_r^{\text{cs}}$ in the same way as the truncated quadratic module $\mathcal{M}(S)_r$ in (3.18). Now, let us state the sparse variant of the primal-dual hierarchy (3.36)-(3.37) of lower bounds for $\lambda_{\min}(f, S)$.

For all $r \geq r_{\min}$, the sparse variant of SDP (3.37) is

$$\begin{aligned} L^r_{\text{cs}}(f,S) := \inf_L \quad & \sum_{l=1}^{p} \langle \mathbf{M}_r(L, I_l), G_{f_l} \rangle \\ \text{s.t.} \quad & L(1) = 1 \,, \\ & \mathbf{M}_r(L, I_l) \succeq 0 \,, \quad l \in [p] \,, \\ & \mathbf{M}_{r-r_j}(g_j L, I_l) \succeq 0 \,, \quad j \in J_l \,, \quad l \in [p] \,, \\ & L : \mathbb{R}\langle \underline{x}, I_1 \rangle_{2r} + \cdots + \mathbb{R}\langle \underline{x}, I_p \rangle_{2r} \to \mathbb{R} \text{ linear} \,, \end{aligned} \tag{3.39}$$

whose dual is the sparse variant of SDP (3.36):

$$\begin{aligned} \lambda^r_{\text{cs}}(f,S) := \sup_b \quad & b \\ \text{s.t.} \quad & f - b \in \mathcal{M}(S)_r \,. \end{aligned} \tag{3.40}$$

Recall that an $\varepsilon$-neighborhood of 0 is the set $\mathcal{N}_\varepsilon$ defined for a given $\varepsilon > 0$ by:

$$\mathcal{N}_\varepsilon := \bigcup_{k \in \mathbb{N}} \left\{ \underline{A} := (A_1, \dots, A_n) \in \mathbb{S}_k^n : \varepsilon^2 - \sum_{i=1}^{n} A_i^2 \succeq 0 \right\} .$$

**Lemma 3.3.14** *If $h \in \mathbb{R}\langle \underline{x} \rangle$ vanishes on an $\varepsilon$-neighborhood of 0, then $h = 0$.*

**Proposition 3.3.15** *Let $S \cup \{f\} \subseteq \operatorname{Sym} \mathbb{R}\langle \underline{x} \rangle$, assume that $\mathcal{D}_S$ contains an $\varepsilon$-neighborhood of 0 and that $\mathcal{D}_S$ is as in (3.20) with the additional quadratic constraints (3.19). Then SDP (3.39) admits strictly feasible solutions.*

**Corollary 3.3.16** *Let $S \cup \{f\} \subseteq \operatorname{Sym} \mathbb{R}\langle \underline{x} \rangle$, assume that $\mathcal{D}_S$ is as in (3.20) with the additional quadratic constraints (3.19). Let Assumption 3.3.4 hold. Then, one has*

$$\lim_{r \to \infty} L^r_{\mathrm{cs}}(f,S) = \lim_{r \to \infty} \lambda^r_{\mathrm{cs}}(f,S) = \lambda_{\min}(f,S) \,. \tag{3.41}$$

As for the unconstrained case, there is no sparse variant of the "perfect" Positivstellensatz stated in [53, §4.4] or [125], for constrained eigenvalue optimization over convex nc semialgebraic sets, such as those associated either to the sparse nc ball $\mathbb{B}^{\mathrm{cs}} := \{1 - \sum_{i \in I_1} x_i^2, \ldots, 1 - \sum_{i \in I_p} x_i^2\}$ or the nc polydisc $\mathbb{D} := \{1 - x_1^2, \ldots, 1 - x_n^2\}$. Namely, for an nc polynomial $f$ of degree $2d+1$, computing only SDP (3.32) with optimal value $\lambda^{d+1}_{\mathrm{cs}}(f,S)$ when $S = \mathbb{B}^{\mathrm{cs}}$ or $S = \mathbb{D}^{\mathrm{cs}}$ does not suffice to obtain the value of $\lambda_{\min}(f,S)$. This is explained in Example 3.3.3 below, which implies that there is no sparse variant of [53, Corollary 4.18] when $S = \mathbb{B}^{\mathrm{cs}}$.

**Example 3.3.3** *Let us consider a randomly generated cubic polynomial $f = f_1 + f_2$ with*

$$\begin{aligned}
f_1 = 4 &- x_1 + 3x_2 - 3x_3 - 3x_1^2 - 7x_1x_2 + 6x_1x_3 - x_2x_1 - 5x_3x_1 + 5x_3x_2 \\
&- 5x_1^3 - 3x_1^2x_3 + 4x_1x_2x_1 - 6x_1x_2x_3 + 7x_1x_3x_1 + 2x_1x_3x_2 - x_1x_3^2 \\
&- x_2x_1^2 + 3x_2x_1x_2 - x_2x_1x_3 - 2x_2^3 - 5x_2^2x_3 - 4x_2x_3^2 - 5x_3x_1^2 \\
&+ 7x_3x_1x_2 + 6x_3x_2x_1 - 4x_3x_2x_2 - x_3^2x_1 - 2x_3^2x_2 + 7x_3^3 \,, \\
f_2 = &- 1 + 6x_2 + 5x_3 + 3x_4 - 5x_2^2 + 2x_2x_3 + 4x_2x_4 - 4x_3x_2 + x_3^2 - x_3x_4 \\
&+ x_4x_2 - x_4x_3 + 2x_4^2 - 7x_2^3 + 4x_2x_3^2 + 5x_2x_3x_4 - 7x_2x_4x_3 - 7x_2x_4^2 \\
&+ x_3x_2^2 + 6x_3x_2x_3 - 6x_3x_2x_4 - 3x_3^2x_2 - 7x_3^2x_4 + 6x_3x_4x_2 \\
&- 3x_3x_4x_3 - 7x_3x_4^2 + 3x_4x_2^2 - 7x_4x_2x_3 - x_4x_2x_4 - 5x_4x_3^2 \\
&+ 7x_4x_3x_4 + 6x_4^2x_2 - 4x_4^3 \,,
\end{aligned}$$

*and the nc polyball $S = \mathbb{B}^{\mathrm{cs}} = \{1 - x_1^2 - x_2^2 - x_3^2, 1 - x_2^2 - x_3^2 - x_4^2\}$ corresponding to $I_1 = \{1,2,3\}$ and $I_2 = \{2,3,4\}$. Then, one has $\lambda^2_{\mathrm{cs}}(f,S) \simeq -27.536 < \lambda^3_{\mathrm{cs}}(f,S) \simeq -27.467 \simeq \lambda^2_{\min}(f,S) = \lambda_{\min}(f,S)$.*

## Extracting Optimizers

Here, we explain how to extract a pair of optimizers $(\underline{A}, \mathbf{v})$ for the eigenvalue optimization problems when the flatness and irreducibility conditions of Theorem 3.3.2 hold. We apply the `SparseGNS` procedure described earlier (and explicitly stated in [J3, Algorithm 4.6]) on the optimal solution of SDP (3.32) in the unconstrained case or SDP (3.39) in the constrained case. In the unconstrained case, we have the following sparse variant of [53, Proposition 4.4].

**Proposition 3.3.17** *Given $f$ as in Theorem 3.3.3, let us assume that SDP (3.32) yields an optimal solution $\mathbf{M}_{d+1}(L)$ associated to $L^{d+1}_{\mathrm{cs}}(f)$. If the linear functional $L$ underlying $\mathbf{M}_{d+1}(L)$ satisfies the flatness (H1) and irreducibility (H2) conditions stated in Theorem 3.3.2, then one has*

$$\lambda_{\min}(f) = L^{d+1}_{\mathrm{cs}}(f) = \sum_{l=1}^{p} \langle \mathbf{M}_{d+1}(L, I_l), \mathbf{G}_{f_l} \rangle \,.$$

We can extract optimizers for the unconstrained minimal eigenvalue problem (3.28) thanks to the following algorithm.

In the constrained case, the next result is the sparse variant of [53, Theorem 4.12] and is a direct corollary of Theorem 3.3.2.

```
Require: $f \in \operatorname{Sym}\mathbb{R}\langle \underline{x} \rangle_{2d}$ satisfying Assumption 3.3.4.
Compute $L_{\text{cs}}^{d+1}(f)$ by solving SDP (3.32)
if SDP (3.32) is unbounded or its optimum is not attained then
    Stop
end if
Let $\mathbf{M}_{d+1}(L)$ be an optimizer of SDP (3.32). Compute $\underline{A}, \mathbf{v} :=$ SparseGNS $(\mathbf{M}_{d+1}(L))$.
Ensure: $\underline{A}$ and $\mathbf{v}$.
```

SparseEigGNS

**Corollary 3.3.18** *Let $S \cup \{f\} \subseteq \operatorname{Sym}\mathbb{R}\langle \underline{x} \rangle$, assume that $\mathcal{D}_S$ is as in (3.20) with the additional quadratic constraints (3.19). Suppose Assumptions 3.3.4(i)-(ii) hold. Let $\mathbf{M}_r(L)$ be an optimal solution of SDP (3.39) with value $L^r(f,S)$, for $r \geq r_{\min} + \delta$, such that $L$ satisfies the assumptions of Theorem 3.3.2. Then, there exist $t \in \mathbb{N}$, $\underline{A} \in \mathcal{D}_S^t$ and a unit vector $\mathbf{v}$ such that*

$$\lambda_{\min}(f,S) = \langle f(\underline{A})\mathbf{v}, \mathbf{v} \rangle = L^r(f,S)\,.$$

**Remark 3.3.3** As in the dense case [53, Algorithm 4.2], one can provide a randomized algorithm to look for flat optimal solutions for the constrained eigenvalue problem (3.34). The underlying reason which motivates this randomized approach is work by Nie, who derives in [227] a hierarchy of SDP programs, with a random objective function, that converges to a flat solution (under mild assumptions).

**Example 3.3.4** *Consider the sparse polynomial $f = f_1 + f_2$ from Example 3.3.3. The Hankel matrix $\mathbf{M}_3(L)$ obtained when computing $\lambda_{\text{cs}}^3$ by solving (3.39) for $r = 3$ satisfies the flatness (H1) and irreducibility (H2) conditions of Theorem 3.3.2. We can thus apply the* `SparseGNS` *algorithm yielding*

$$A_1 = \begin{bmatrix} 0.0059 & 0.0481 & 0.1638 & 0.4570 \\ 0.0481 & -0.2583 & 0.5629 & -0.2624 \\ 0.1638 & 0.5629 & 0.3265 & -0.3734 \\ 0.4570 & -0.2624 & -0.3734 & -0.2337 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} -0.3502 & 0.0080 & 0.1411 & 0.0865 \\ 0.0080 & -0.4053 & 0.2404 & -0.1649 \\ 0.1411 & 0.2404 & -0.0959 & 0.3652 \\ 0.0865 & -0.1649 & 0.3652 & 0.4117 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} -0.7669 & -0.0074 & -0.1313 & -0.0805 \\ -0.0074 & -0.4715 & -0.2238 & 0.1535 \\ -0.1313 & -0.2238 & 0.0848 & -0.3400 \\ -0.0805 & 0.1535 & -0.3400 & -0.2126 \end{bmatrix}$$

$$A_4 = \begin{bmatrix} 0.3302 & -0.1839 & 0.1811 & -0.0404 \\ -0.1839 & -0.1069 & 0.5114 & -0.0570 \\ 0.1811 & 0.5114 & 0.1311 & -0.3664 \\ -0.0404 & -0.0570 & -0.3664 & 0.4440 \end{bmatrix}$$

*where*

$$f(\underline{A}) = \begin{bmatrix} -10.3144 & 3.9233 & -5.0836 & -7.7828 \\ 3.9233 & 1.8363 & 4.5078 & -7.5905 \\ -5.0836 & 4.5078 & -19.5827 & 13.9157 \\ -7.7828 & -7.5905 & 13.9157 & 8.3381 \end{bmatrix}$$

*has minimal eigenvalue $-27.4665$ with unit eigenvector*

$$\mathbf{v} = \begin{bmatrix} 0.1546 & -0.2507 & 0.8840 & -0.3631 \end{bmatrix}^T.$$

*In this case all the ranks involved were equal to four. So $A_2$ and $A_3$ were computed already from $\mathbf{M}_3(L, I_1 \cap I_2)$, after an appropriate basis change $A_1$ (and the same $A_2, A_3$) was obtained from $\mathbf{M}_3(L, I_1)$, and finally $A_4$ was computed from $\mathbf{M}_3(L, I_2)$.*

In [J3, § 6], the interested reader can find more details about SDP relaxations allowing one to under-approximate the smallest trace of an nc polynomial on a semialgebraic set.

## Numerical experiments

The aim of this section is to provide experimental comparison between the bounds given by the dense relaxations (using `NCeigMin` under `NCSOStools`) and the ones produced by our sparse variants. The resulting algorithm, denoted by `NCeigMinSparse`, is currently implemented in `NCSOStools` [58]. This software library is available within MATLAB and interfaced with the SDP solver MOSEK [7], which turned out to yield better performance than SeDuMi 1.3 [279]. All numerical results were obtained using a cluster available at the Faculty of mechanical engineering, University of Ljubljana, which has 30 TFlops computing performance. For our computations we used only one computing node which consisted of 2 Intel Xeon X5670 2,93GHz processors, each with 6 computing cores; 48 GB DDR3 memory; 500 GB hard drive. We ran MATLAB in a plain (sequential) mode, without imposing any paralelization.

In Table 3.1, we report results obtained for minimizing the eigenvalue of the nc variants of the following functions:

- The chained singular function [68]:

$$f_{\mathrm{csf}} := \sum_{i \in J} \left( (x_i + 10x_{i+1})^2 + 5(x_{i+2} - x_{i+3})^2 + (x_{i+1} - 2x_{i+2})^4 + 10(x_i - 10x_{i+3})^4 \right),$$

  where $J = [n-3]$ and $n$ is a multiple of 4. In this case, one can choose $I_l = \{l, l+1, l+2, l+3\}$ for all $l \in [n-3]$ so that the associated sparsity pattern satisfies (3.6).

- The generalized Rosenbrock function [222]:

$$f_{\mathrm{gRf}} := 1 + \sum_{i=1}^{n-1} \left( 100(x_{i+1} - x_i^2)^2 + (1 - x_{i+1})^2 \right).$$

  In this case, one can choose $I_l = \{l, l+1\}$ for all $l \in [n-1]$ so that the associated sparsity pattern satisfies (3.6).

We compute bounds on the minimal eigenvalues of $f = f_{\mathrm{csf}}$ for each $n \in \{4, \ldots, 24\}$ being a multiple of 4, and $f_{\mathrm{gRf}}$ for even values of $n \in \{2, \ldots, 20\}$. For both functions, the minimal eigenvalue is 0. We indicate in Table 3.1 the data related to the SDP solved by MOSEK. For each value of $n$, $m_{\mathrm{sdp}}$ stands for the total number of constraints and $n_{\mathrm{sdp}}$ stands for the total number of variables either of the SDP program (3.31) solved to compute $\lambda_{\min}(f)$ or the SDP program (3.32) solved to compute $\lambda^2_{\mathrm{cs}}(f)$. As emphasized in the columns corresponding to $m_{\mathrm{sdp}}$, the size of the SDP programs can be significantly reduced after exploiting sparsity, which is consistent with Remark 3.3.2. While the procedure `NCeigMin` does not take sparsity into account, it relies on the Newton chip method [53, §2.3] to reduce the number of variables involved in the Hankel matrix from SDP (3.31). This explains why $n_{\mathrm{sdp}}$ is smaller for some values of $n$ (e.g. $n = 8$ for $f_{\mathrm{csf}}$) when running `NCeigMin`. However, the sparse procedure `NCeigMinSparse` turns out to be very often more efficient to compute the minimal eigenvalue. So far, our `NCeigMinSparse` procedure is limited by the computational abilities of current SDP solvers (such as MOSEK) to handle matrices with more constraints and variables than the ones obtained, e.g., for the chained singular function at $n = 24$ (see the related

Table 3.1: `NCeigMin` vs `NCeigMinSparse` for unconstrained minimal eigenvalues of the chained singular and generalized Rosenbrock functions.

| $f$ | $n$ | `NCeigMin` | | | | `NCeigMinSparse` | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | $m_{\text{sdp}}$ | $n_{\text{sdp}}$ | $\lambda^2_{\min}(f)$ | time (s) | $m_{\text{sdp}}$ | $n_{\text{sdp}}$ | $\lambda^2_{\text{cs}}(f)$ | time (s) |
| $f_{\text{csf}}$ | 4 | 78 | 169 | 0 | 0.42 | 78 | 169 | 0 | 0.37 |
| | 8 | 398 | 841 | 0 | 1.33 | 165 | 1323 | 0 | 3.69 |
| | 12 | 974 | 2025 | 0 | 4.35 | 298 | 2205 | 0 | 6.28 |
| | 16 | 1806 | 3721 | 0 | 14.29 | 413 | 3087 | 0 | 9.18 |
| | 20 | 2894 | 5929 | 0 | 52.47 | 537 | 3969 | 0 | 12.78 |
| | 24 | 4238 | 8649 | 0 | 152.17 | 661 | 4851 | 0 | 17.65 |
| $f_{\text{gRf}}$ | 10 | 200 | 400 | 0 | 0.56 | 95 | 441 | 0 | 1.39 |
| | 12 | 288 | 576 | 0 | 0.81 | 117 | 539 | 0 | 1.78 |
| | 14 | 392 | 784 | 0 | 1.12 | 139 | 637 | 0 | 2.20 |
| | 16 | 512 | 1024 | 0 | 1.46 | 161 | 735 | 0 | 2.67 |
| | 18 | 648 | 1296 | 0 | 2.15 | 183 | 833 | 0 | 3.26 |
| | 20 | 800 | 1600 | 0 | 2.92 | 205 | 931 | 0 | 4.10 |

Table 3.2: `NCeigMin` vs `NCeigMinSparse` for minimal eigenvalue of the chained singular function on the nc polydisc $S_{\text{csf}}$.

| $n$ | `NCeigMin` | | | | `NCeigMinSparse` | | | |
|---|---|---|---|---|---|---|---|---|
| | $m_{\text{sdp}}$ | $n_{\text{sdp}}$ | $\lambda^2(f_{\text{csf}}, S_{\text{csf}})$ | time (s) | $m_{\text{sdp}}$ | $n_{\text{sdp}}$ | $\lambda^2_{\text{cs}}(f_{\text{csf}}, S_{\text{csf}})$ | time (s) |
| 4 | 161 | 641 | 315.21 | 3.25 | 161 | 641 | 315.21 | 2.95 |
| 8 | 1009 | 6625 | 965.48 | 146.99 | 525 | 1923 | 965.48 | 4.66 |
| 12 | 3121 | 28705 | 1615.7 | 7891.6 | 889 | 3205 | 1615.7 | 7.43 |
| 16 | | | − | | 1253 | 4487 | 2266.05 | 13.20 |
| 20 | | | − | | 1617 | 5769 | 2916.32 | 18.50 |
| 24 | | | − | | 1981 | 7051 | 3566.56 | 26.38 |

values of $m_{\text{sdp}}$ and $n_{\text{sdp}}$ in the corresponding column). It turns out that exploiting the sparsity pattern yields SDP programs with significantly fewer variables than the ones obtained after running the Newton chip method.

In the column reporting timings, we indicate the time needed to prepare *and* solve the SDP relaxation. For values of $n, d \gtrsim 8$, our current implementation in (interpreted) MATLAB happens to be rather inefficient to construct the SDP problem itself, mainly because we rely on a naive nc polynomial arithmetic. To overcome this computational burden, we plan to interface `NCSOStools` with a `C` library implementing a more sophisticated monomial arithmetic. We also emphasize that for these unconstrained problems, each function is a sum of sparse hermitian squares, thus the sparse procedure `NCeigMinSparse` always retrieves the same optimal value as the dense procedure `NCeigMin`. However, the bound computed via the sparse procedure can be a strict lower bound of the minimal eigenvalue, as shown in Lemma 3.3.11.

In Table 3.2, we report results obtained for minimizing the eigenvalue of the nc chained singular function on the semialgebraic set $S_{\text{csf}} := \{1 - x_1^2, \ldots, 1 - x_n^2, x_1 - 1/3, \ldots, x_n - 1/3\}$ for $n \in \{4, 8, 12, 16, 20, 24\}$. Since $f$ has degree 4, it follows from [53, Corollary 4.18] that it is enough to solve SDP (3.32) with optimal value $\lambda^2(f, S_{\text{csf}})$ to compute the minimal eigenvalue $\lambda_{\min}(f, S_{\text{cs}})$. For the experiments described in Table 3.2, we cannot rely on the Newton chip method as in the unconstrained case. Thus the dense procedure `NCeigMin` suffers from a severe computational bur-

Table 3.3: `NCeigMin` vs `NCeigMinSparse` for minimal eigenvalue of random cubic polynomials on the nc polyball $S = \mathbb{B}^{\text{cs}}$.

| $n$ | `NCeigMin` | | | | `NCeigMinSparse` | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | $m_{\text{sdp}}$ | $n_{\text{sdp}}$ | $\lambda_2(f_{\text{rand}}, S)$ | time (s) | $r$ | $m_{\text{sdp}}$ | $n_{\text{sdp}}$ | $\lambda^r_{\text{cs}}(f_{\text{rand}}, S)$ | time (s) |
| 4 | 71 | 491 | -53.64 | 3.31 | 2 | 79 | 370 | -53.72 | 1.18 |
| | | | | | 3 | 729 | 3538 | -53.64 | 12.64 |
| 6 | 239 | 2045 | -142.52 | 26.79 | 2 | 179 | 740 | -142.62 | 2.33 |
| | | | | | 3 | 1535 | 7076 | -142.52 | 29.52 |
| 8 | 559 | 5815 | -165.89 | 171.30 | 2 | 279 | 1110 | -166.32 | 3.73 |
| | | | | | 3 | 2341 | 10614 | -165.91 | 62.70 |
| 10 | 1079 | 13289 | -199.62 | 857.95 | 2 | 379 | 1480 | -200.51 | 5.43 |
| | | | | | 3 | 3147 | 14152 | -199.66 | 139.22 |
| 11 | 1429 | 18985 | -180.39 | 2111.26 | 2 | 429 | 1665 | -180.93 | 6.58 |
| | | | | | 3 | 3550 | 15921 | -180.40 | 209.73 |
| 12 | - | - | - | - | 2 | 479 | 1850 | -385.89 | 7.82 |
| | | | | | 3 | 3953 | 17690 | -384.87 | 289.12 |
| 16 | - | - | - | - | 2 | 679 | 2590 | -344.31 | 15.46 |
| | | | | | 3 | 5565 | 24766 | -342.15 | 975.43 |
| 20 | - | - | - | - | 2 | 879 | 3330 | -504.36 | 31.41 |
| | | | | | 3 | 7177 | 31842 | -503.02 | 2587.61 |

den for $n > 10$; the symbol "$-$" in a column entry indicates that the calculation did not finish in a couple of hours. As already observed before for the unconstrained case, the sparse procedure `NCeigMinSparse` performs much better than `NCeigMin`. Surprisingly, `NCeigMinSparse` yields the same bounds as `NCeigMin` at the minimal relaxation order $s = 2$, for all values of $n \leq 10$.

As shown in Example 3.3.3, there is no guarantee to obtain the above mentioned convergence behavior in a systematic way. We consider randomly generated cubic $n$-variate polynomials $f_{\text{rand}}$ satisfying Assumption 3.3.4 with $I_l = \{l, l+1, l+2\}$, for all $l \in [n-2]$. The corresponding nc polyball is given by $\mathbb{B}^{\text{cs}} := \{1 - x_1^2 - x_2^2 - x_3^2, \ldots, 1 - x_{n-2}^2 - x_{n-1}^2 - x_n^2\}$. In Table 3.3, we report results obtained for minimizing the eigenvalue of $f_{\text{rand}}$ on $\mathbb{B}^{\text{cs}}$, for each value of $n \in \{4, \ldots, 10\}$. Here again, the sparse procedure `NCeigMinSparse` yields better performance than `NCeigMin`. Moreover, the sparse bound obtained for each $n \leq 10$ at minimal relaxation order $r = 2$ already gives an accurate approximation of the optimal bound provided by the dense procedure. We emphasize that the value of the third order relaxation obtained with the sparse procedure is almost equal to the optimal bound. In addition, the dense procedure cannot handle to solve the minimal order relaxation for $n > 10$, while we can always obtain a lower bound of the eigenvalue with `NCeigMinSparse`.

## 3.4 Term sparsity in polynomial optimization

As emphasized earlier for distinct applications, exploiting CS arising from POP may allow to significantly reduce the computational cost of the related hierarchy of SDP relaxations. Nevertheless many POP can be fairly sparse, but they do not exhibit a non-trivial csp (i.e. the corresponding csp graph is complete). For instance, if $f$ has a term involving all variables or some constraint, e.g., $1 - \|\mathbf{x}\|^2 \geq 0$, involves all variables, then the csp is trivial. Besides, even if a POP admits a non-trivial csp, some maximal cliques of the csp graph (after a chordal extension) may still have a large size (say over 20), which makes the resulting SDP problem still hard to solve.

However, instead of exploiting sparsity from the perspective of *variables*, one can also exploit sparsity from the perspective of *terms*. The results of this section have been published in [J22, J21].

Roughly speaking, the considered sparsity can be also represented by a graph, which is called a correlative sparsity pattern (tsp) graph. But unlike the csp graph, the nodes of a tsp graph correspond to monomials (not necessarily variables) and the edges of the graph grasp the links between monomials in the SOS representation of positive polynomials. We can design an iterative procedure to enlarge the tsp graph in order to iteratively exploit term sparsity (TS) in the initial POP. Each iteration consists of two steps: (i) a support-extension operation and (ii) a block-closure operation on adjacency matrices or a chordal-extension.

In doing so we obtain a sequence

$$G_1 \subseteq G_2 \subseteq \cdots \subseteq G_k$$

of graphs where "$G_i \subseteq G_{i+1}$" means that $G_i$ is a subgraph of $G_{i+1}$. Step (ii) consists of either performing completion of the connected components for each graph or performing an approximately minimum chordal extension. Then combining this iterative procedure with the standard moment-SOS hierarchy results in a two-level moment-SOS hierarchy with quasi block-diagonal SDP matrices. When the sizes of blocks are small, then the associated SDP relaxations can be drastically much cheaper to solve.

To some extent, TS (focusing on monomials) is finer than the CS (focusing on variables). If a POP is sparse in the sense of CS (i.e. the csp graph is not complete), then it must be sparse in the sense of TS (i.e. the tsp graph is not complete), while the converse is not necessarily true. So the basic idea for solving large-scale POP is as follows: first exploit CS to obtain a coarse decomposition in terms of variables with cliques, and second exploit TS for subsystems involving each clique of variables. This idea has been successfully carried out in [R14] and will be presented later on.

## The TSSOS hierarchy for unconstrained POP

In this section, we describe an iterative procedure to exploit TS for the primal-dual moment-SOS relaxations of unconstrained POP. Let $f(\mathbf{x}) = \sum_{\alpha \in \mathbf{A}} f_\alpha \mathbf{x}^\alpha \in \mathbb{R}[\mathbf{x}]$ with $\mathrm{supp}(f) = \mathbf{A}$ (assuming $\mathbf{0} \in \mathbf{A}$) and $\mathscr{B}$ be a monomial basis with $t = |\mathscr{B}|$. Assume that $f \in \mathbb{R}[\mathbf{x}]$ is a polynomial of degree $2d$. We can for instance choose the standard monomial basis $\mathscr{B} = \mathbf{x}^{\mathbb{N}^n_d}$ or the integer points in half of the Newton polytope of $f$, i.e., by

$$\mathscr{B} = \frac{1}{2}\mathcal{C}(f) \cap \mathbb{N}^n \subseteq \mathbb{N}^n_d. \tag{3.42}$$

Recall the formulation of unconstrained POP:

$$\mathbf{P}: \quad f_{\min} := \inf_{\mathbf{x}} \{f(\mathbf{x}) : \mathbf{x} \in \mathbb{R}^n\}. \tag{3.43}$$

For convenience, we abuse notation in the sequel and denote by $\mathscr{B}$ (resp. $\beta$) instead of $\mathbf{x}^{\mathscr{B}}$ (resp. $\mathbf{x}^\beta$) a monomial basis (resp. a monomial). In the following, we will consider graphs with $V = \mathscr{B}$ as the set of nodes. Suppose that $G(V, E)$ is such a graph. We define two operations on $G$: *support-extension* and *chordal-extension*. The support-extension of $G$, denoted by $\mathrm{SE}(G)$, is the graph with nodes $\mathscr{B}$ and with edges

$$E(\mathrm{SE}(G)) := \{\{\beta, \gamma\} \mid (\beta, \gamma) \in V \times V,\, \beta \neq \gamma,\, \beta + \gamma \in \mathrm{supp}(G)\}.$$

**Example 3.4.1** *Consider the following graph $G(V, E)$ with*

$$V = \{1, x_1, x_2, x_3, x_2x_3, x_1x_3, x_1x_2\} \text{ and } E = \{\{1, x_2x_3\}, \{x_2, x_1x_3\}\}.$$

*Then $E(\mathrm{SE}(G)) = \{\{1, x_2x_3\}, \{x_2, x_1x_3\}, \{x_2, x_3\}, \{x_1, x_2x_3\}, \{x_3, x_1x_2\}\}$. See Figure 3.6 for the support-extension $\mathrm{SE}(G)$ of $G$.*

Figure 3.6: The support-extension $\mathrm{SE}(G)$ of $G$. The dashed edges are added in the process of support-extension.

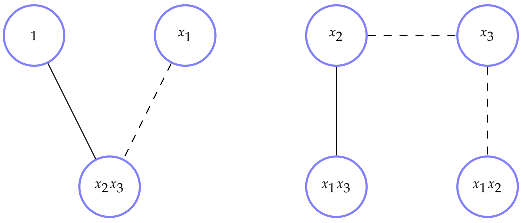
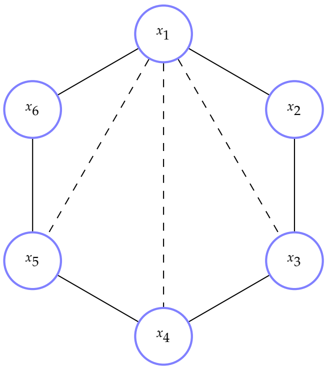


Figure 3.7: The chordal-extension $\overline{G}$ of $G$. The dashed edges are added in the process of chordal-extension.



Any specific chordal-extension of $G$ is denoted by $\overline{G}$.

**Example 3.4.2** *Consider the following graph $G(V, E)$ with $V = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ and $E = \{\{x_1, x_2\}, \{x_2, x_3\}, \{x_3, x_4\}, \{x_4, x_5\}, \{x_5, x_6\}, \{x_6, x_1\}\}$. See Figure 3.7 for the chordal-extension $\overline{G}$ of $G$.*

**Remark 3.4.1** *For a graph $G(V, E)$, the chordal-extension of $G$ is usually not unique. A chordal-extension with the least number of edges is called a* minimum chordal-extension. *Finding a minimum chordal-extension of a graph is an NP-hard problem in general. Fortunately, several heuristic algorithms, such as the minimum degree ordering, are known to efficiently produce a good approximation [5, 122].*

In the sequel, we assume that for graphs $G, H$ with the same set of nodes, if $E(G) \subseteq E(H)$, then $E(\overline{G}) \subseteq E(\overline{H})$. This assumption is reasonable since any chordal-extension of $H$ must be also a chordal-extension of $G$.

We define $G^{\mathrm{tsp}} = G_0(V, E_0)$ to be the graph with $V = \mathscr{B}$ and

$$E_0 = \{\{\beta, \gamma\} \mid (\beta, \gamma) \in V \times V, \beta \neq \gamma, \beta + \gamma \in \mathbf{A} \cup (2\mathscr{B})\}, \tag{3.44}$$

where $2\mathscr{B} = \{2\beta \mid \beta \in \mathscr{B}\}$. We call $G^{\mathrm{tsp}} = G_0$ the tsp graph associated with $f$.

For $k \geq 1$, we recursively define a sequence of graphs $(G_k(V, E_k))_{k \geq 1}$ by

$$G_k := \overline{\mathrm{SE}(G_{k-1})}. \tag{3.45}$$

Given a monomial basis $\mathscr{B}$, the moment matrix $M_{\mathscr{B}}(\mathbf{y})$ associated with $\mathscr{B}$ and $\mathbf{y}$ is the matrix with rows and columns indexed by $\mathscr{B}$. Then the moment SDP relaxation of $\mathbf{P}$ is

$$f_{\min}^d := \begin{array}{ll} \inf & L_{\mathbf{y}}(f) \\ \text{s.t.} & \mathbf{M}_{\mathscr{B}}(\mathbf{y}) \succeq 0, \\ & y_{\mathbf{0}} = 1. \end{array} \tag{3.46}$$

If $f$ is sparse, by replacing $\mathbf{M}_{\mathscr{B}}(\mathbf{y}) \succeq 0$ with the weaker condition $\mathbf{M}_{\mathscr{B}}(\mathbf{y}) \in \Pi_{G_k}(\mathbb{S}_t^+)$ in (3.46), we obtain a sparse moment SDP relaxation of (3.43) for each $k \geq 1$:

$$\mathbf{P}_{\text{ts}}^k : \quad \begin{array}{ll} \inf & L_{\mathbf{y}}(f) \\ \text{s.t.} & \mathbf{M}_{\mathscr{B}}(\mathbf{y}) \in \Pi_{G_k}(\mathbf{S}_+^r), \\ & y_{\mathbf{0}} = 1, \end{array} \tag{3.47}$$

with optimal value denoted by $f_{\text{ts}}^k$. We call $k$ the *sparse order*. By construction, one has $G_k \subseteq G_{k+1}$ for all $k \geq 1$ and therefore the sequence of graphs $(G_k(V, E_k))_{k\geq 1}$ stabilizes after a finite number of steps. The intuition behind the support-extension operation is that once one position related to $y_\alpha$ in the moment matrix $M_{\mathscr{B}}(\mathbf{y})$ is "activated" in the sparsity pattern, then all positions related to $y_\alpha$ in $M_{\mathscr{B}}(\mathbf{y})$ should be "activated". Theorem 3.1.1 and Theorem 3.1.2 provide the rationale behind the mechanism of the chordal-extension operation.

**Theorem 3.4.1** *The sequence $(f_{\text{ts}}^k)_{k\geq 1}$ is monotone nondecreasing and $f_{\text{ts}}^k \leq f_{\min}^d$ for all $k$.*

As a consequence of Theorem 3.4.1, we obtain the following hierarchy of lower bounds for the optimum of the original problem $\mathbf{P}$:

$$f_{\text{ts}}^1 \leq f_{\text{ts}}^2 \leq \cdots \leq f_{\min}^d \leq f_{\min}. \tag{3.48}$$

When we use an approximately minimum chordal-extension, we say that (3.47) (and its associated sequence (3.48)) is the *chordal-TSSOS* hierarchy for $\mathbf{P}$. The *maximal* chordal-extension of a graph is the one that completes every connected component of the graph. If we use the maximal chordal-extension in (3.45), then we say that (3.47) is the *block-TSSOS* hierarchy.

We can show (see [J22] for more details) that the sequence of optima of the block-TSSOS hierarchy always converges to the optimum of the dense moment-SOS relaxation. Unlike the block-TSSOS hierarchy, there is no guarantee that the chordal-TSSOS hierarchy of lower bounds $(f_{\text{ts}}^k)_{k\geq 1}$ converges to the value $f_{\min}^d$. The following is an example.

**Example 3.4.3** *Consider the commutative version of the polynomial from (3.27):*

$$f = x_1^2 - 2x_1x_2 + 3x_2^2 - 2x_1^2x_2 + 2x_1^2x_2^2 - 2x_2x_3 + 6x_3^2 + 18x_2^2x_3 - 54x_2x_3^2 + 142x_2^2x_3^2.$$

*The monomial basis computed from the Newton polytope is $\{1, x_1, x_2, x_3, x_1x_2, x_2x_3\}$. We have $E_0 = \{\{1, x_1x_2\}, \{1, x_2x_3\}, \{x_1, x_1x_2\}, \{x_1, x_2\}, \{x_2, x_3\}, \{x_2, x_2x_3\}, \{x_3, x_2x_3\}\}$. Figure 3.8 shows the tsp graph $G_0$ (without dashed edges) and its chordal-extension $G_1$ (with dashed edges) for $f$. The graph sequence $(G_k)_{k\geq 1}$ stabilizes at $k = 1$. Solving the SDP problem $\mathbf{P}_{\text{ts}}^1$ associated with $G_1$, we obtain $f_{\text{ts}}^1 \approx -0.00355$ while we have $f_{\min}^2 = f_{\min} = 0$.*

Even though there is no theoretical convergence guarantees for the chordal-TSSOS hierarchy, the convergence often occurs in practice (see [J21, § 6]).

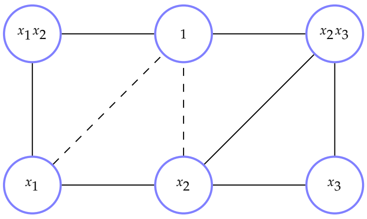
For each $k \geq 1$, the dual SDP problem of (3.47) is

$$\begin{array}{lll} \sup & b & \\ \text{s.t.} & \langle \mathbf{G}, \mathbf{B}_\alpha \rangle = f_\alpha - b1_{\alpha=\mathbf{0}}, & \forall \alpha \in \operatorname{supp}(G_k) \cup (2\mathscr{B}), \\ & \mathbf{G} \in \mathbb{S}_t^+ \cap \mathbb{S}(G_k), & \end{array} \tag{3.49}$$

where $t = |\mathscr{B}|$ and $\mathbf{B}_\alpha$ has been defined after (2.7).

**Proposition 3.4.2** *For each $k \geq 1$, there is no duality gap between $\mathbf{P}_{\text{ts}}^k$ and its dual (3.49).*

Figure 3.8: The tsp graph $G_0$ and its chordal-extension $G_1$ for Example 3.4.3.



## Comparison with SDSOS [3]

The following definition of SDSOS polynomials has been introduced and studied in [3]. A symmetric matrix $\mathbf{G} \in \mathbb{S}_t$ is *diagonally dominant* if $\mathbf{G}_{ii} \geq \sum_{j \neq i} |\mathbf{G}_{ij}|$ for $i \in [t]$ and is *scaled diagonally dominant* if there exists a positive definite $t \times t$ diagonal matrix $\mathbf{D}$ such that $\mathbf{DGD}$ is diagonally dominant. We say that a polynomial $f(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$ is a *scaled diagonally dominant sum of squares* (SDSOS) polynomial if it admits a Gram matrix representation as in (3.49) (with $b = 0$) with a scaled diagonally dominant Gram matrix $\mathbf{G}$. We denote the set of SDSOS polynomials by $SDSOS$.

Following [3], by replacing the nonnegativity condition in $\mathbf{P}$ with the SDSOS condition, one obtains the SDSOS relaxation of $\mathbf{P}$ and $\mathbf{P}^d$:

$$(\mathrm{SDSOS}): \quad f_{\mathrm{sdsos}} := \sup_b \{b : f(\mathbf{x}) - b \in \mathrm{SDSOS}\}.$$

The first TSSOS relaxation is always better than the SDSOS relaxation:

**Theorem 3.4.3** *With the above notation, one has* $f_{\mathrm{ts}}^1 \geq f_{sdsos}$.

In addition, the first TSSOS relaxation is always equivalent to the first (classical dense) moment relaxation in the quadratic case.

**Theorem 3.4.1** *Suppose that* $f \in \mathbb{R}[\mathbf{x}]$ *in* $\mathbf{P}$ *is a quadratic polynomial. Then* $f_{\mathrm{ts}}^1 = f_{\mathrm{min}}^1$.

## The TSSOS hierarchy for constrained POP

In this section, we describe an iterative procedure to exploit TS for the primal-dual moment-SOS hierarchy of a constrained POP:

$$\mathbf{P}: \quad f_{\min} = \min_{\mathbf{x} \in \mathbb{R}^n} \{f(\mathbf{x}) : \mathbf{x} \in \mathbf{X}\}. \tag{3.50}$$

where $\mathbf{X} = \{\mathbf{x} \in \mathbb{R}^n : g_1(\mathbf{x}) \geq 0, \ldots, g_m(\mathbf{x}) \geq 0\}$, is defined as in (1.1). Let

$$\mathbf{A} = \mathrm{supp}(f) \cup \bigcup_{j=1}^{m} \mathrm{supp}(g_j). \tag{3.51}$$

As usual $r_j := \lceil \deg(g_j)/2 \rceil, j \in [m]$ and $r_{\min} = \max\{\lceil \deg(f)/2 \rceil, r_1, \ldots, r_m\}$. Fix a relaxation order $r \geq r_{\min}$ in Lasserre's hierarchy. Let $g_0 = 1$, $r_0 = 0$ and $\mathscr{B}_{j,r} = \mathbb{N}_{r-r_j}^n$ be the standard monomial basis for $j = 0, \ldots, m$. We define a graph $G_{0,r}^{(0)}(V_{0,r}, E_{0,r}^{(0)})$ with $V_{0,r} = \mathscr{B}_{0,r}$ and

$$E_{0,r}^{(0)} = \{\{\beta, \gamma\} \mid (\beta, \gamma) \in V_{0,r} \times V_{0,r}, \beta \neq \gamma, \beta + \gamma \in \mathbf{A} \cup (2\mathscr{B}_{0,r})\}. \tag{3.52}$$

We call $G_0$ the tsp graph associated with $(\mathrm{Q}_0)$.

For $k \geq 1$, we recursively define a sequence of graphs $(G_{j,r}^{(k)}(V_{j,r}, E_{j,r}^{(k)}))_{k\geq 1}$ with $V_{j,r} = \mathscr{B}_{j,r}$ for $j = 0, \ldots, m$ by

$$G_{0,r}^{(k)} := \overline{\mathrm{SE}(G_{0,r}^{(k-1)})} \text{ and } G_{j,r}^{(k)} := \overline{F_{j,r}^{(k)}}, \quad j \in [m], \tag{3.53}$$

where $F_{j,r}^{(k)}$ is the graph with $V(F_{j,r}^{(k)}) = \mathscr{B}_{j,r}$ and

$$\begin{aligned} E(F_{j,r}^{(k)}) =& \{\{\beta, \gamma\} \mid (\beta, \gamma) \in \mathscr{B}_{j,r} \times \mathscr{B}_{j,r}, \, \beta \neq \gamma, \\ & (\mathrm{supp}(g_j) + \beta + \gamma) \cap \mathrm{supp}(G_{0,r}^{(k-1)}) \neq \emptyset\}, \quad j \in [m]. \end{aligned} \tag{3.54}$$

Let $t_j := \binom{n+r-r_j}{r-r_j} = |\mathscr{B}_{j,r}|$ and consider the dense moment relaxation of $\mathbf{P}$:

$$\mathbf{P}^r : \quad \inf_{\mathbf{y}} \{ L_{\mathbf{y}}(f) : y_0 = 1; \quad \mathbf{M}_{r-r_j}(g_j \, \mathbf{y}) \succeq 0, \quad j = 0, \ldots, m \}, \tag{3.55}$$

Therefore by replacing $\mathbf{M}_{r-r_j}(g_j\mathbf{y}) \succeq 0$ with the weaker condition $\mathbf{M}_{r-r_j}(g_j\mathbf{y}) \in \Pi_{G_{j,r}^{(k)}}(\mathbf{S}_{t_j}^{+})$ for $j = 0, \ldots, m$ in (3.55), we obtain the following sparse SDP relaxation of $\mathbf{P}^r$ and $\mathbf{P}$ for each $k \geq 1$:

$$\mathbf{P}_{\mathrm{ts}}^{r,k} : \quad \begin{array}{rll} f_{\mathrm{ts}}^{r,k} := & \inf & L_{\mathbf{y}}(f) \\ & \text{s.t.} & \mathbf{M}_r(\mathbf{y}) \in \Pi_{G_{0,r}^{(k)}}(\mathbf{S}_{t_0}^{+}), \\ & & \mathbf{M}_{r-r_j}(g_j\mathbf{y}) \in \Pi_{G_{j,r}^{(k)}}(\mathbf{S}_{t_j}^{+}), \quad j \in [m], \\ & & y_{\mathbf{0}} = 1. \end{array} \tag{3.56}$$

We call $k$ the *sparse order*. By construction, one has $G_{j,r}^{(k)} \subseteq G_{j,r}^{(k+1)}$ for all $j, k$. Therefore, for every $j$, the sequence of graphs $(G_{j,r}^{(k)})_{k\geq 1}$ stabilizes after a finite number of steps.

**Theorem 3.4.2** *Fixing a relaxation order $r \geq r_{\min}$, the sequence $(f_{\mathrm{ts}}^{r,k})_{k\geq 1}$ is monotone nondecreasing and $f_{\mathrm{ts}}^{r,k} \leq f_{\min}^{r}$ for all $k \geq 1$. When using the block-TSSOS hierarchy, the sequence converges to $f_{\min}^{r}$ in finitely many steps.*

**Theorem 3.4.3** *Fixing a sparse order $k \geq 1$, the sequence $(f_{\mathrm{ts}}^{r,k})_{r\geq r_{\min}}$ is monotone nondecreasing.*

Combining Theorem 3.4.2 with Theorem 3.4.3, we have the following two-level hierarchy of lower bounds for the optimum of $(\mathrm{Q}_0)$:

$$\begin{array}{ccccccc}
f_{\mathrm{ts}}^{r_{\min},1} & \leq & f_{\mathrm{ts}}^{r_{\min},2} & \leq & \cdots & \leq & f_{\min}^{r_{\min}} \\
\rotatebox{90}{$\geq$} & & \rotatebox{90}{$\geq$} & & & & \rotatebox{90}{$\geq$} \\
f_{\mathrm{ts}}^{r_{\min}+1,1} & \leq & f_{\mathrm{ts}}^{r_{\min}+1,2} & \leq & \cdots & \leq & f_{\min}^{r_{\min}+1} \\
\rotatebox{90}{$\geq$} & & \rotatebox{90}{$\geq$} & & & & \rotatebox{90}{$\geq$} \\
\vdots & & \vdots & & \vdots & & \vdots \\
\rotatebox{90}{$\geq$} & & \rotatebox{90}{$\geq$} & & & & \rotatebox{90}{$\geq$} \\
f_{\mathrm{ts}}^{r,1} & \leq & f_{\mathrm{ts}}^{r,2} & \leq & \cdots & \leq & f_{\min}^{r} \\
\rotatebox{90}{$\geq$} & & \rotatebox{90}{$\geq$} & & & & \rotatebox{90}{$\geq$} \\
\vdots & & \vdots & & \vdots & & \vdots
\end{array} \tag{3.57}$$

The array of lower bounds (3.57) (and its associated SDP relaxations (3.56)) is what we call the TSSOS moment-SOS hierarchy associated with $\mathbf{P}$.

For each $k \geq 1$, the dual of $\mathbf{P}_{\mathrm{ts}}^{r,k}$ reads as

$$\begin{array}{ll}
\sup & b \\
\text{s.t.} & \sum_{j=0}^{m} \langle \mathbf{C}_{\alpha}^{j}, \mathbf{G}_j \rangle = f_{\alpha} - b \mathbf{1}_{\alpha=\mathbf{0}}, \quad \forall \alpha \in \cup_{j=0}^{m} (\mathrm{supp}(g_j) + \mathrm{supp}(G_{j,r}^{(k)})) \cup (2\mathscr{B}_{0,r}), \\
& \mathbf{G}_j \in \mathbf{S}_{t_j}^{+} \cap \mathbf{S}(G_{j,r}^{(k)}), \quad j = 0, \ldots, m,
\end{array} \tag{3.58}$$

where $\mathbf{C}_{\alpha}^{j}$ is defined after (2.7).

**Proposition 3.4.4** *Assume that* $\mathbf{X}$ *has a nonempty interior. Then there is no duality gap between* $\mathbf{P}_{\mathrm{ts}}^{r,k}$ *and its dual* (3.58) *for any* $r \geq r_{\min}$ *and* $k \geq 1$.

As in the unconstrained case, there is no theoretical guarantee that the chordal-TSSOS hierarchy converges to the optimal value $f_{\min}^{r}$ of the dense moment-SOS relaxation but it occurs very often in practice. As in Theorem 3.4.1, for quadratically constrained quadratic programs, we always have $f_{\mathrm{ts}}^{1,1} = f_{\min}^{1}$.

Dedicated algorithms allow one to obtain a possibly smaller monomial basis $\mathscr{B}$; see [J21, § 4] for more details.

### Sign-symmetries and a sparse representation theorem for positive polynomials

Here, we consider the block-TSSOS hierarchy. As mentioned earlier, the sequence of graphs $(G_{j,r}^{(k)})_{k \geq 1}$ stabilizes after a finite number of steps to a graph, denoted by $G_{j,r}^{(*)}$. Let $B_{j,r}^{(*)}$ be the $\{0,1\}$-binary adjacency matrix associated to $G_{j,r}^{(*)}$. If $B_{0,r}^{(*)}$ is not an all-one matrix, then it induces a partition of the monomial basis $\mathbb{N}_r^n$: two vectors $\beta, \gamma \in \mathbb{N}_r^n$ belong to the same block if and only if the rows and columns indexed by $\beta, \gamma$ belong to the same block in $B_{0,r}^{(*)}$. We next provide an interpretation of this partition in terms of *sign-symmetries*, a tool introduced in [199] to characterize block-diagonal SOS decompositions of positive polynomials.

**Definition 3.4.5** *Given a finite set* $\mathbf{A} \subseteq \mathbb{N}^n$*, the* sign-symmetries *of* $\mathbf{A}$ *are defined by all vectors* $\mathbf{r} \in \mathbb{Z}_2^n$ *such that* $\mathbf{r}^T \alpha \equiv 0 \; (mod\; 2)$ *for all* $\alpha \in \mathbf{A}$.

For any $\alpha \in \mathbb{N}^n$, we define $(\alpha)_2 := (\alpha_1 (\mathrm{mod}\, 2), \ldots, \alpha_n (\mathrm{mod}\, 2)) \in \mathbb{Z}_2^n$. We also use the same notation for any subset $\mathbf{A} \subseteq \mathbb{N}^n$, i.e., $(\mathbf{A})_2 := \{(\alpha)_2 \mid \alpha \in \mathscr{A}\} \subseteq \mathbb{Z}_2^n$.

For a subset $S \subseteq \mathbb{Z}_2^n$, the *subspace* spanned by $S$ in $\mathbb{Z}_2^n$, denoted by $\overline{S}$, is the set $\{(\sum_i \mathbf{s}_i)_2 \mid \mathbf{s}_i \in S\}$ and the *orthogonal complement space* of $S$ in $\mathbb{Z}_2^n$, denoted by $S^{\perp}$, is the set $\{\alpha \in \mathbb{Z}_2^n \mid \alpha^T \mathbf{s} \equiv 0 \,(\mathrm{mod}\, 2)\, , \forall \mathbf{s} \in S\}$.

**Remark 3.4.2** *By definition, the set of sign-symmetries of* $\mathbf{A}$ *is just the orthogonal complement space* $(\mathbf{A})_2^\perp$ *in* $\mathbb{Z}_2^n$. *Hence the sign-symmetries of* $\mathbf{A}$ *can be essentially represented by a basis of the subspace* $(\mathbf{A})_2^\perp$ *in* $\mathbb{Z}_2^n$.

**Lemma 3.4.6** *Let* $S \subseteq \mathbb{Z}_2^n$. *Then* $(S^\perp)^\perp = \overline{S}$.

**Lemma 3.4.7** *Suppose* $B$ *is a* $\{0,1\}$*-binary matrix with rows and columns indexed by* $\mathscr{B} \subseteq \mathbb{N}^n$ *and that* $G$ *is its adjacency graph. Let* $\overline{G}$ *be the adjacency graph of* $\overline{B}$. *Then* $(\mathrm{supp}(\overline{G}))_2 \subseteq \overline{(\mathrm{supp}(G))_2}$.

**Theorem 3.4.4** *For a positive integer* $r$, *let* $\mathbf{A} \subseteq \mathbb{N}_{2r}^n$ *be defined as in* (3.51) *and assume that the sign-symmetries of* $\mathbf{A}$ *are given by the columns of a binary matrix denoted by* $R$. *Then* $\beta, \gamma$ *belong to the same block in the partition of* $\mathbb{N}_r^n$ *induced by* $B_{0,r}^{(*)}$ *if and only if* $R^T(\beta + \gamma) \equiv 0 \ (mod\ 2)$.

**Example 3.4.4** *Theorem 3.4.4 is applied for the standard monomial basis* $\mathbb{N}_r^n$. *If a smaller monomial basis is chosen, then we only have the "only if" part of the conclusion in Theorem 3.4.4. Let* $f = 1 + x^2y^4 + x^4y^2 + x^4y^4 - xy^2 - 3x^2y^2$ *and* $\mathbf{A} = \mathrm{supp}(f)$. *The monomial basis given by the Newton polytope is* $\mathscr{B} = \{1, xy, xy^2, x^2y, x^2y^2\}$. *The sign-symmetries of* $\mathbf{A}$ *consist of two elements:* $(0,0)$ *and* $(0,1)$. *According to the sign-symmetries,* $\mathscr{B}$ *is partitioned into* $\{1, xy^2, x^2y^2\}$ *and* $\{xy, x^2y\}$ *(recall that* $\beta, \gamma$ *belong to the same block in the partition induced by the sign-symmetries* $R$ *if and only if* $R^T(\beta+\gamma) \equiv 0 \ (mod\ 2)$*). On the other hand, we have*

$$B_{\mathbf{A}}^{(*)} = B_{\mathbf{A}}^{(1)} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

*Thus the partition of* $\mathscr{B}$ *induced by* $B_{\mathbf{A}}^{(*)}$ *is* $\{1, xy^2, x^2y^2\}$, $\{xy\}$ *and* $\{x^2y\}$, *which is a refinement of the partition determined by the sign-symmetries.*

By virtue of Theorem 3.4.4, the partition of the monomial basis $\mathbb{N}_{r-r_j}^n$ induced by $B_{j,r}^{(*)}$, $j \in [m]$, can also be characterized using sign-symmetries.

**Corollary 3.4.8** *Notations are as in Theorem 3.4.4 and assume* $\mathbf{A} = \mathrm{supp}(f) \cup \bigcup_{j=1}^m \mathrm{supp}(g_j)$. *Then* $\beta, \gamma$ *belong to the same block in the partition of* $\mathbb{N}_{r-r_j}^n$ *induced by* $B_{j,r}^{(*)}$ *if and only if* $R^T(\beta+\gamma) \equiv 0 \ (mod\ 2)$, $j \in [m]$.

Theorem 3.4.4 together with Corollary 3.4.8 implies that the block structure of the TSSOS hierarchy at each relaxation order converges to the block structure determined by the sign-symmetries related to the support of the input data, under the assumption that the standard monomial bases are used.

**Remark 3.4.3** *Though it is guaranteed that at the final step of the block-TSSOS hierarchy, an equivalent SDP (with block structure determined by sign-symmetries if the standard monomial bases are used) is retrieved, in practice it frequently happens that the same optimal value as the dense moment-SOS relaxation is achieved at an earlier step, even at the first step, but with a much cheaper computational cost.*

As a corollary of Theorem 3.4.4 and Corollary 3.4.8, we obtain the following sparse representation theorem for positive polynomials over basic compact semialgebraic sets.

**Theorem 3.4.5** *Assume that the quadratic module $\mathcal{M}(\mathbf{X})$ is Archimedean and that $f$ is positive on $\mathbf{X}$. Let $\mathbf{A} = \text{supp}(f) \cup \bigcup_{j=1}^{m} \text{supp}(g_j)$ and let the sign-symmetries of $\mathbf{A}$ be given as the columns of a binary matrix denoted by $R$. Then $f$ can be decomposed as*

$$f = \sigma_0 + \sum_{j=1}^{m} \sigma_j g_j,$$

*for some SOS polynomials $\sigma_0, \sigma_1, \ldots, \sigma_m$ satisfying $R^T\alpha \equiv 0 \ (mod\ 2)$ for any $\alpha \in \text{supp}(\sigma_j), j = 0, \ldots, m$.*

## Numerical experiments

Next, we present numerical results of the proposed TSSOS hierarchies of block SDP relaxations. Our related algorithm, named TSSOS[2], is implemented in Julia for constructing instances of the SDP problems (3.47) and (3.56), then relies on MOSEK [7] to solve them. TSSOS utilizes the Julia packages LIGHTGRAPHS [50] to handle graphs and JUMP [84] to model SDP. In the following subsections, we compare the performance of TSSOS with that of GLOPTIPOLY [132] and YALMIP [200]. As for TSSOS, GLOPTIPOLY and YALMIP also rely on MOSEK to solve SDP problems.

We first consider Lyapunov functions emerging from some networked systems. In [119], the authors propose a structured SOS decomposition for those systems, which allows them to handle structured Lyapunov function candidates up to 50 variables.

The following polynomial is from Example 2 in [119]:

$$f = \sum_{i=1}^{n} a_i(x_i^2 + x_i^4) - \sum_{i=1}^{n}\sum_{s=1}^{n} b_{is} x_i^2 x_k^2,$$

where $a_i$ are randomly chosen from $[1, 2]$ and $b_{is}$ are randomly chosen from $[\frac{0.5}{n}, \frac{1.5}{n}]$. Here, $n$ is the number of nodes in the network. The task is to determine whether $f$ is globally nonnegative.

The initial tsp graph $G_0$ (see Figure 3.9) has 1 maximal clique of size $n+1$ (involving the nodes $1, x_1^2, \ldots, x_n^2$), $\frac{n(n-1)}{2}$ maximal cliques of size 3 (involving the nodes $x_i^2, x_j^2, x_i x_j$ for each pair $\{i, j\}, i \neq j$) and $n$ maximal cliques of size 1 (involving the node $x_i$ for each $i$). Note that $G_0$ is already a chordal graph. So we have $G_1 = G_0$.

The computational cost for the TSSOS (with sparse order $k = 1$) and dense SDP relaxations is displayed in Table 3.4. In the column "#SDP blocks", $i \times j$ means $j$ SDP blocks of size $i$.

Table 3.4: Computational cost comparison for the sparse and dense SDP relaxations

| | #SDP blocks | #equality constraints |
|---|---|---|
| TSSOS | $3 \times \frac{n(n-1)}{2}, 1 \times n, (n+1) \times 1$ | $\frac{3n(n-1)}{2} + 2n + 1$ |
| Dense | $\binom{n+2}{2} \times 1$ | $\binom{n+4}{4}$ |

We solve SDP (3.47) at $k = 1$ with TSSOS for $n = 10, 20, 30, 40, 50, 60, 70, 80$. The results are listed in Table 3.5. Note that "mb" stands for the maximal size of blocks of SDP matrices.

For this example, the size of systems that can be handled in [119] is up to $n = 50$ nodes while our approach can easily handle systems with up to $n = 80$ nodes.

[2] `https://github.com/wangjie212/TSSOS`

Figure 3.9: The tsp graph $G_0$ of $f$. This is a subgraph of $G_0$. The whole graph $G_0$ is obtained by putting all such subgraphs together.

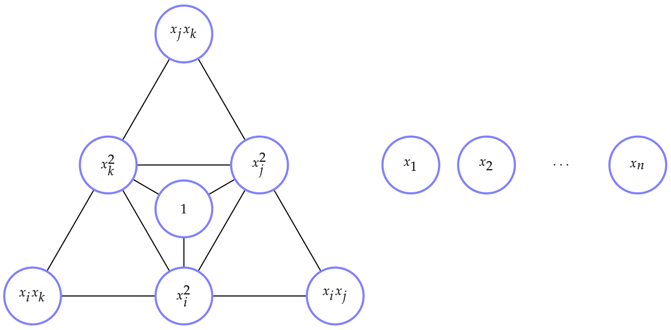


Table 3.5: Results for the first network problem

| $n$ | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 |
|---|---|---|---|---|---|---|---|---|
| mb | 11 | 31 | 31 | 41 | 51 | 61 | 71 | 81 |
| time | 0.006 | 0.03 | 0.10 | 0.34 | 0.92 | 1.9 | 4.7 | 12 |

The following polynomial is from Example 3 in [119]:

$$V = \sum_{i=1}^{n} a_i(\frac{1}{2}x_i^2 - \frac{1}{4}x_i^4) + \frac{1}{2}\sum_{i=1}^{n}\sum_{s=1}^{n} b_{is}\frac{1}{4}(x_i - x_k)^4, \tag{3.59}$$

where $a_i$ are randomly chosen from $[0.5, 1.5]$ and $b_{is}$ are randomly chosen from $[\frac{0.5}{n}, \frac{1.5}{n}]$. The task is to analyze the domain on which the Hamiltonian function $V$ for a network of Duffing oscillators is positive definite. We use the following condition to establish an inner approximation of the domain on which $V$ is positive definite:

$$f = V - \sum_{i=1}^{n} \lambda_i x_i^2(g - x_i^2) \geq 0, \tag{3.60}$$

where $\lambda_i > 0$ are scalar decision variables and $g$ is a fixed positive scalar. Clearly, the condition (3.60) ensures that $V$ is positive definite when $x_i^2 < g$. Here we solve SDP (3.47) at $k = 1$ with TSSOS for $n = 10, 20, 30, 40, 50$. For this example, graphs arising in the TSSOS hierarchy are naturally chordal, so we simply exploit chordal decompositions. This example was also examined in [209] to demonstrate the advantage of SDSOS programming compared to dense SOS programming. The method based on SDSOS programming was executed in SPOT [214] with MOSEK as a second-order cone programming solver. The results are listed in Table 3.6.

For this example, TSSOS uses much less decision variables than SDSOS programming, and hence spends less time compared to SDSOS programming. On the other hand, TSSOS computes a positive definite form $V$ after selecting a value for $g$ up to 2 (which is the same as the maximal value obtained by the dense SOS) while the method in [119] can select $g$ up to 1.8 and the one based on SDSOS programming only works out for a maximal value of $g$ up to around 1.5.

Table 3.6: Results for the second network problem

| $n$ | | 10 | 20 | 30 | 40 | 50 |
|---|---|---|---|---|---|---|
| #SDP blocks | TSSOS | $3 \times 45$,<br>$1 \times 10$,<br>$11 \times 1$ | $3 \times 190$,<br>$1 \times 20$,<br>$21 \times 1$ | $3 \times 435$,<br>$1 \times 30$,<br>$31 \times 1$ | $3 \times 780$,<br>$1 \times 40$,<br>$41 \times 1$ | $3 \times 1225$,<br>$1 \times 50$,<br>$51 \times 1$ |
| | SDSOS | $2 \times 2145$ | $2 \times 26565$ | $2 \times 122760$ | $2 \times 370230$ | $2 \times 878475$ |
| #SDP vars | TSSOS | 346 | 1391 | 3136 | 5581 | 8726 |
| | SDSOS | 6435 | 79695 | 368280 | 1110690 | 2635425 |
| time | TSSOS | 0.01 | 0.06 | 0.17 | 0.50 | 0.89 |
| | SDSOS | 0.47 | 1.14 | 5.47 | 20 | 70 |

## 3.5 Combining correlative and term sparsity

For large-scale POP, it is natural to ask whether one can combine CS and TS to further exploit both sparsity features possessed by the problem. As we shall see next, the answer is affirmative. The material from this section is issued from [R14].

A first natural idea to combine CS and TS would be to apply the TSSOS hierarchy for each clique *separately*, once the cliques of variables have been obtained from the csp graph of (3.50). However, with this naive approach convergence may be lost and below we describe the extra care needed to avoid this annoying consequence.

### The CS-TSSOS hierarchy

Let $G^{\text{csp}}$ be the csp graph associated to (3.50), $\overline{G}^{\text{csp}}$ a chordal extension of $G^{\text{csp}}$ and $I_l, l \in [p]$ be the maximal cliques of $\overline{G}^{\text{csp}}$ with cardinal denoted by $n_l$. As in Section 3.1, the set of variables $\mathbf{x}$ is partitioned into $\mathbf{x}(I_1), \mathbf{x}(I_2), \ldots, \mathbf{x}(I_p)$. Let $J_1, \ldots, J_p$ be defined as in Section 3.1.

Now we apply TS to each subsystem involving variables $\mathbf{x}(I_l)$, $l \in [p]$ respectively as follows. Let

$$\mathbf{A} := \text{supp}(f) \cup \bigcup_{j=1}^{m} \text{supp}(g_j) \text{ and } \mathbf{A}_l := \{\alpha \in \mathbf{A} \mid \text{supp}(\alpha) \subseteq I_l\} \tag{3.61}$$

for $l \in [p]$. As before, $r_{\min} := \max\{\lceil \deg(f)/2 \rceil, r_1, \ldots, r_m\}$, $r_0 := 0$ and $g_0 := 1$. Fix a relaxation order $r \geq r_{\min}$ and let $\mathbb{N}^{n_l}_{r-r_j}$ be the standard monomial basis for $j \in \{0\} \cup J_l, l \in [p]$. Let $G^{\text{tsp}}_{r,l}$ be the tsp graph with nodes $\mathbb{N}^{n_l}_{r-r_j}$ associated to $\mathbf{A}_l$ defined as in Section 3.4. Note that we embed $\mathbb{N}^{n_l}_{r-r_j}$ into $\mathbb{N}^{n}_{r-r_j}$ via the map $\alpha = (\alpha_i) \in \mathbb{N}^{n_l}_{r-r_j} \mapsto \alpha' = (\alpha'_i) \in \mathbb{N}^{n}_{r-r_j}$ which satisfies

$$\alpha'_i = \begin{cases} \alpha_i, & \text{if } i \in I_l, \\ 0, & \text{otherwise.} \end{cases}$$

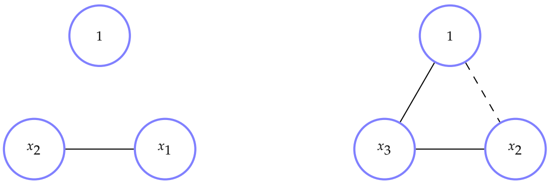
Assume that $G^{(0)}_{r,l,0} = G^{\text{tsp}}_{r,l}$ and $G^{(0)}_{r,l,j}, j \in J_l, l \in [p]$ are empty graphs. Letting

$$\mathcal{C}^{(k-1)}_r := \cup_{l=1}^{p} \cup_{j \in \{0\} \cup J_l} (\text{supp}(g_j) + \text{supp}(G^{(k-1)}_{r,l,j})), \quad k \geq 1, \tag{3.62}$$

we recursively define a sequence of graphs $(G^{(k)}_{r,l,j}(V_{r,l,j}, E^{(k)}_{r,l,j}))_{k \geq 1}$ with $V_{r,l,j} = \mathbb{N}^{n_l}_{r-r_j}$ for $j \in \{0\} \cup J_l, l \in [p]$ by

$$G^{(k)}_{r,l,j} := \overline{F^{(k)}_{r,l,j}}, \tag{3.63}$$

Figure 3.10: The tsp graphs of Example 3.5.1. Each node has a self-loop which is not displayed for simplicity. The dashed edge is added after the maximal chordal extension.



where $F_{r,l,j}^{(k)}$ is the graph with $V(F_{r,l,j}^{(k)}) = \mathbb{N}_{r-r_j}^{n_l}$ and

$$E(F_{r,l,j}^{(k)}) = \{\{\beta, \gamma\} \mid (\mathrm{supp}(g_j) + \beta + \gamma) \cap C_r^{(k-1)} \neq \emptyset\}. \tag{3.64}$$

**Example 3.5.1** *Let* $f = 1 + x_1^2 + x_2^2 + x_3^2 + x_1x_2 + x_2x_3 + x_3$ *and consider the unconstrained POP:* $\min\{f(\mathbf{x}) : \mathbf{x} \in \mathbb{R}^n\}$*. The variables is then partitioned into two cliques:* $\{x_1, x_2\}$ *and* $\{x_2, x_3\}$*. The tsp graphs for these two cliques are illustrated in Figure 3.10 (the left (resp. right) graph corresponds to the first (resp. second) clique). If we apply the TSSOS hierarchy (using the maximal chordal extension in (3.63)) separately for each clique, then the graph sequences* $(G_{1,l}^{(k)})_{k\geq 1}, l = 1, 2$ *(the subscript j is omitted since there are no constraints) stabilize at* $k = 1$*. However, the added (dashed) edge in the right graph corresponds to the monomial* $x_2$*, which only involves the variable* $x_2$ *belonging to the first clique. Hence we need to add the edge connecting* 1 *and* $x_2$ *to the left graph in order to get convergence guarantees. Consequently, the graph sequences* $(G_{1,l}^{(k)})_{k\geq 1}, l = 1, 2$ *stabilize at* $k = 2$*.*

Let $t_{l,j} := \binom{n_l + r - r_j}{r - r_j}$ for all $l, j$. Then for each $k \geq 1$, the SDP hierarchy based on combined correlative and term sparsities, abbreviated as CS-TSSOS hierarchy, for (3.50) is defined as:

$$\mathbf{P}_{\text{cs-ts}}^{r,k} : \begin{cases} \inf & L_{\mathbf{y}}(f) \\ \text{s.t.} & \mathbf{M}_r(\mathbf{y}, I_l) \in \Pi_{G_{r,l,0}^{(k)}}(\mathbb{S}_{t_{l,0}}^+), \quad l \in [p], \\ & \mathbf{M}_{r-r_j}(g_j\mathbf{y}, I_l) \in \Pi_{G_{r,l,j}^{(k)}}(\mathbb{S}_{t_{l,j}}^+), \quad j \in J_l, \quad l \in [p], \\ & y_{\mathbf{0}} = 1, \end{cases} \tag{3.65}$$

with optimal value denoted by $f_{\text{cs-ts}}^{r,k}$.

## Convergence guarantees

**Theorem 3.5.1** *For any* $r \geq r_{\min}$*, the sequence* $(f_{\text{cs-ts}}^{r,k})_{k\geq 1}$ *is monotone non-decreasing and* $f_{\text{cs-ts}}^{r,k} \leq f_{\min}^r$ *for all k. For any* $k \geq 1$*, the sequence* $(f_{\text{cs-ts}}^{r,k})_{r\geq r_{\min}}$ *is monotone non-decreasing. If we use the maximal chordal extension in (3.63), then for any* $r \geq r_{\min}$*, the sequence* $(f_{\text{cs-ts}}^{r,k})_{k\geq 1}$ *converges to* $f_{\min}^r$ *in finitely many steps.*

As in Section 3.4, we obtain a two-level hierarchy of lower bounds for the optimum of **P** (3.50).

Figure 3.11: The csp graph of Example 3.5.2

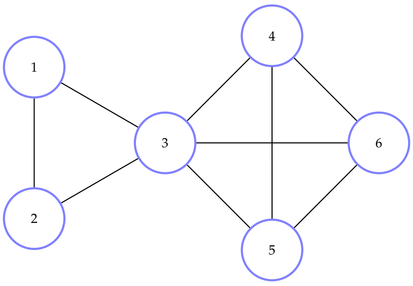
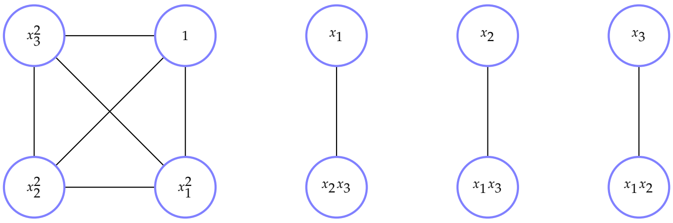


Figure 3.12: The tsp graph for the first clique of Example 3.5.2. Each node has a self-loop which is not displayed for simplicity.



**Example 3.5.2** *Let $f = 1 + \sum_{i=1}^{6} x_i^4 + x_1x_2x_3 + x_3x_4x_5 + x_3x_4x_6 + x_3x_5x_6 + x_4x_5x_6$, and consider the unconstrained POP: $\min\{f(\mathbf{x}) : \mathbf{x} \in \mathbb{R}^n\}$. Apply the CS-TSSOS hierarchy (using the maximal chordal extension in (3.53)) to this problem. First, using the csp graph (see Figure 3.11), partition variables into the two cliques $\{x_1, x_2, x_3\}$ and $\{x_3, x_4, x_5, x_6\}$. Figure 3.12 and Figure 3.13 illustrate the tsp graphs for the first clique and the second clique respectively. For the first clique one obtains four blocks of SDP matrices with respective sizes $4, 2, 2, 2$. For the second clique one obtains two blocks of SDP matrices with respective sizes $5, 10$. Thus the original size $28$ of the SDP matrix has been reduced to a maximal size of $10$.*

*If one applies the TSSOS hierarchy (using the maximal chordal extension in (3.53)) directly to the problem (i.e. without partitioning variables), then the tsp graph is illustrated in Figure 3.14. One obtains five blocks of SDP matrices with respective size $7, 2, 2, 2, 10$. Compared to the CS-TSSOS case, the two blocks of SDP matrices with respective sizes $4, 5$ are replaced by a single block SDP matrix with size $7$.*

The CS-TSSOS hierarchy entails a trade-off. One has the freedom to choose a specific chordal extension for any graph involved in (3.50). This choice affects the resulting size of (submatrix) blocks and the quality of optimal values of the corresponding CS-TSSOS hierarchy. Intuitively, chordal extensions with less edges should lead to (submatrix) blocks of smaller size and optimal values of (possibly) lower quality while chordal extensions with more edges should lead to (submatrix) blocks with larger size and optimal values of (possibly) higher quality.

As in Section 3.4, we obtain a sparse representation for polynomials positive on a basic compact semialgebraic set.

**Theorem 3.5.2** *Let $f \in \mathbb{R}[\mathbf{x}]$ and $\mathbf{X}$ be as in Assumption (3.1.3), with $I_l, J_l$ be defined as in Section 3.1 and $\mathbf{A} = \operatorname{supp}(f) \cup \bigcup_{j=1}^{m} \operatorname{supp}(g_j)$. Assume that the sign-symmetries of $\mathbf{A}$ are represented by the*

Figure 3.13: The tsp graph for the second clique of Example 3.5.2

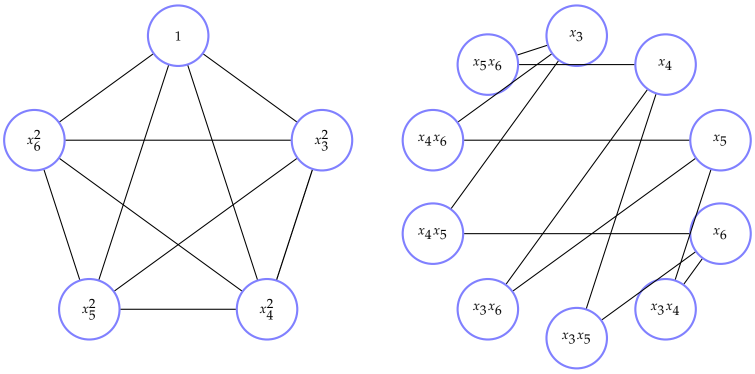
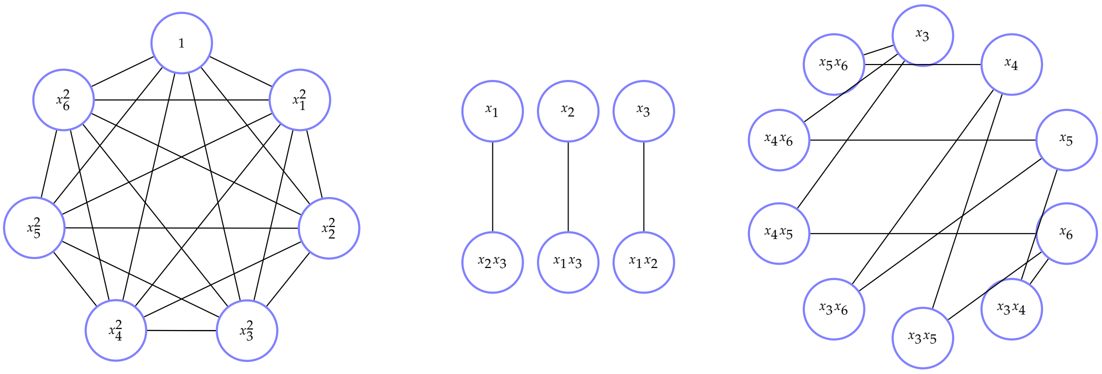


Figure 3.14: The tsp graph without partitioning variables of Example 3.5.2



*columns of the binary matrix $R$. If $f$ is positive on $\mathbf{X}$, then*

$$f = \sum_{l=1}^{p} (\sigma_{l,0} + \sum_{j \in J_l} \sigma_{l,j} g_j), \tag{3.66}$$

*for some polynomials $\sigma_{l,j} \in \Sigma[\mathbf{x}(I_l)], j \in \{0\} \cup J_l, l \in [p]$, satisfying $R^T\alpha \equiv 0 \ (mod\ 2)$ for any $\alpha \in \operatorname{supp}(\sigma_{l,j})$, i.e., $\operatorname{supp}(\sigma_{l,j})_2 \subseteq R^\perp$.*

### Extracting a solution

In the case of dense moment-SOS relaxations, there is a standard procedure described in [131] to extract globally optimal solutions when the so-called flatness condition is satisfied and this procedure is also generalized to the correlative sparse setting in [179, § 3.3]. However, in our combined sparsity setting, the corresponding procedure cannot be directly applied because the moment matrix associated to each clique does not involve enough moment variables. In order to extract a solution, we may add an order-one (dense) moment matrix for each clique in (3.54):

$$\begin{cases}\inf & L_{\mathbf{y}}(f)\\ \text{s.t.} & \mathbf{M}_r(\mathbf{y}, I_l) \in \Pi_{G_{r,l,0}^{(k)}}(\mathbb{S}_{t_{l,0}}^+), \quad l \in [p],\\ & \mathbf{M}_1(\mathbf{y}, I_l) \succeq 0, \quad l \in [p],\\ & \mathbf{M}_{r-r_j}(g_j\mathbf{y}, I_l) \in \Pi_{G_{r,l,j}^{(k)}}(\mathbb{S}_{t_{l,j}}^+), \quad j \in J_l, \quad l \in [p],\\ & y_{\mathbf{0}} = 1.\end{cases} \tag{3.67}$$

Let $\mathbf{y}^{\text{opt}}$ be an optimal solution of (3.67). Typically, $\mathbf{M}_1(\mathbf{y}^{\text{opt}}, I_l)$ (after identifying sufficiently small entries with zero) is a block diagonal matrix (up to permutation). If for all $l$, every block of $\mathbf{M}_1(\mathbf{y}^{\text{opt}}, I_l))$ (see [179, Theorem 3.7]) has rank one, then a globally optimal solution $\mathbf{x}^{\text{opt}}$ to $\mathbf{P}$ (3.50) can be extracted. At the same time, the global optimality is certified. Otherwise, the relaxation might be not exact or yield multiple global solutions. In the latter case, adding a small perturbation to the objective function, as in [300], may yield a unique global solution.

**Remark 3.5.1** *Note that* (3.67) *is a stronger relaxation of* $\mathbf{P}$ *than* $\mathbf{P}^{r,k}_{cs\text{-}ts}$. *Therefore, even if the globally optimal value is not achieved,* (3.67) *still provides a better lower bound for* $\mathbf{P}$ *than* $\mathbf{P}^{r,k}_{cs\text{-}ts}$. *If* $\mathbf{P}$ *is a quadratically constrained quadratic problem, then* (3.67) *is always stronger than the first-order dense relaxation of* $\mathbf{P}$.

## Applications to AC-OPF problems

The AC optimal power flow (AC-OPF) is a central problem in power systems. It can be formulated as the following POP:

$$\begin{cases}\inf\limits_{V_i, S_s^g, S_{ij}} & \sum_{s\in G}(\mathbf{c}_{2s}(\Re(S_s^g))^2 + \mathbf{c}_{1s}\Re(S_s^g) + \mathbf{c}_{0s})\\ \text{s.t.} & \angle V_{\text{ref}} = 0,\\ & \mathbf{S}_s^{gl} \le S_s^g \le \mathbf{S}_s^{gu}, \quad \forall s \in G,\\ & \boldsymbol{v}_i^l \le |V_i| \le \boldsymbol{v}_i^u, \quad \forall i \in N,\\ & \sum_{s\in G_i} S_s^g - \mathbf{S}_i^d - \mathbf{Y}_i^s|V_i|^2 = \sum_{(i,j)\in E_i\cup E_i^R} S_{ij}, \quad \forall i \in N,\\ & S_{ij} = (\mathbf{Y}_{ij}^* - \mathbf{i}\frac{\mathbf{b}_{ij}^c}{2})\frac{|V_i|^2}{|\mathbf{T}_{ij}|^2} - \mathbf{Y}_{ij}^*\frac{V_iV_j^*}{\mathbf{T}_{ij}}, \quad \forall (i,j) \in E,\\ & S_{ji} = (\mathbf{Y}_{ij}^* - \mathbf{i}\frac{\mathbf{b}_{ij}^c}{2})|V_j|^2 - \mathbf{Y}_{ij}^*\frac{V_i^*V_j}{\mathbf{T}_{ij}^*}, \quad \forall (i,j) \in E,\\ & |S_{ij}| \le \mathbf{s}_{ij}^u, \quad \forall (i,j) \in E \cup E^R,\\ & \boldsymbol{\theta}_{ij}^{\Delta l} \le \angle(V_iV_j^*) \le \boldsymbol{\theta}_{ij}^{\Delta u}, \quad \forall (i,j) \in E,\end{cases} \tag{3.68}$$

where $V_i$ is the voltage, $S_s^g$ is the power generation, $S_{ij}$ is the power flow (all are complex variables; $\Re(\cdot)$ and $\angle\cdot$ stand for the real part and the angle of a complex number, respectively) and all symbols in boldface are constants. For a full description on AC-OPF problems, the reader may refer to [18]. By introducing real variables for both real and imaginary part of each complex variable, we can convert an AC-OPF problem to a POP involving only real variables.

To tackle an AC-OPF problem, we first compute a locally optimal solution with a local solver and then rely on an SDP relaxation to certify the global optimality. Suppose that the optimal value reported by the local solver is AC and the optimal value of the SDP relaxation is opt. The *optimality gap* between the locally optimal solution and the SDP relaxation is defined by

$$\text{gap} := \frac{\text{AC} - \text{opt}}{\text{AC}}.$$

Table 3.7: The data for AC-OPF problems

| case name | var | cons | mc | AC | Shor | |
|---|---|---|---|---|---|---|
| | | | | | opt | gap |
| 3_lmbd_api | 12 | 28 | 6 | 1.1242e4 | 1.0417e4 | 7.34% |
| 5_pjm | 20 | 55 | 6 | 1.7552e4 | 1.6634e4 | 5.22% |
| 24_ieee_rts_api | 114 | 315 | 10 | 1.3495e5 | 1.3216e5 | 2.06% |
| 24_ieee_rts_sad | 114 | 315 | 14 | 7.6943e4 | 7.3592e4 | 4.36% |
| 30_as_api | 72 | 297 | 8 | 4.9962e3 | 4.9256e3 | 1.41% |
| 73_ieee_rts_api | 344 | 971 | 16 | 4.2273e5 | 4.1041e5 | 2.91% |
| 73_ieee_rts_sad | 344 | 971 | 16 | 2.2775e5 | 2.2148e5 | 2.75% |
| 118_ieee_api | 344 | 1325 | 21 | 2.4205e5 | 2.1504e5 | 11.16% |
| 118_ieee_sad | 344 | 1325 | 21 | 1.0522e5 | 1.0181e5 | 3.24% |
| 162_ieee_dtc | 348 | 1809 | 21 | 1.0808e5 | 1.0616e5 | 1.78% |
| 162_ieee_dtc_api | 348 | 1809 | 21 | 1.2100e5 | 1.1928e5 | 1.42% |
| 240_pserc | 766 | 3322 | 16 | 3.3297e6 | 3.2818e6 | 1.44% |
| 500_tamu_api | 1112 | 4613 | 20 | 4.2776e4 | 4.2286e4 | 1.14% |
| 500_tamu | 1112 | 4613 | 30 | 7.2578e4 | 7.1034e4 | 2.12% |
| 1888_rte | 4356 | 18257 | 26 | 1.4025e6 | 1.3748e6 | 1.97% |

If the optimality gap is less than 1.00%, then we accept the locally optimal solution as globally optimal. For many AC-OPF problems, the first-order moment-SOS relaxation (Shor relaxation) is already able to certify the global optimality (with an optimality gap less than 1.00%). We focus on more challenging AC-OPF problems, for which the gap is greater than 1.00%. We select such benchmarks from the AC-OPF library *pglib* [18]. Since we shall go to the second-order moment-SOS relaxation, we can replace the variables $S_{ij}$ and $S_{ji}$ by their right-hand side values in (3.68) and then convert the resulting problem to a real POP. The data for these selected AC-OPF benchmarks are displayed in Table 3.7, where the AC values are from *pglib*. Note that "cons" stands for the number of polynomial constraints and "mc" stands for the maximal size of maximal cliques in the csp graph from (3.68).

We execute the (sparse) Shor's relaxation, the second-order CSSOS hierarchy and the second-order CS-TSSOS hierarchy of sparse order $k = 1$. The result for these AC-OPF benchmarks is displayed in Table 3.8. For each instance, the CS-TSSOS hierarchy succeeds to reduce the optimality gap to less than 1.00%. Again, one can still reduce the optimality gap further by increasing the sparse order $k$. We also observe that even if the bound obtained by CSSOS should be theoretically better than the one obtained by CS-TSSOS, CS-TSSOS practically provides slightly more accurate bounds than CSSOS for the tested instances (when CSSOS can be executed), due to numerical uncertainties arising when solving the SDP relaxations related to CSSOS.

Table 3.8: The results for AC-OPF problems

| case name | CSSOS (2nd) | | | | CS-TSSOS (2nd) | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | mb | opt | time | gap | mb | opt | time | gap | CE |
| 3_lmbd_api | 28 | 1.1242e4 | 0.21 | 0.00% | 22 | 1.1242e4 | 0.09 | 0.00% | max |
| 5_pjm | 28 | 1.7437e4 | 0.46 | 0.66% | 22 | 1.7543e4 | 0.30 | 0.05% | max |
| 24_ieee_rts_api | 66 | 1.3339e5 | 4.75 | 1.16% | 31 | 1.3396e5 | 2.01 | 0.73% | max |
| 24_ieee_rts_sad | 120 | 7.5108e4 | 98.3 | 2.38% | 39 | 7.6942e4 | 14.8 | 0.00% | max |
| 30_as_api | 45 | 4.9485e3 | 3.40 | 0.95% | 22 | 4.9833e3 | 2.66 | 0.26% | max |
| 73_ieee_rts_api | 153 | 4.1523e5 | 502 | 1.77% | 44 | 4.1942e5 | 72.8 | 0.78% | max |
| 73_ieee_rts_sad | 153 | 2.2383e5 | 445 | 1.72% | 44 | 2.2755e5 | 79.1 | 0.09% | max |
| 118_ieee_api | 253 | – | – | – | 31 | 2.4180e5 | 82.7 | 0.11% | min |
| 118_ieee_sad | 253 | – | – | – | 73 | 1.0470e5 | 169 | 0.50% | max |
| 162_ieee_dtc | 253 | – | – | – | 34 | 1.0802e5 | 278 | 0.05% | min |
| 162_ieee_dtc_api | 253 | – | – | – | 34 | 1.2096e5 | 201 | 0.03% | min |
| 240_pserc | 153 | 3.2883e6 | 300 | 1.24% | 44 | 3.3042e6 | 33.9 | 0.77% | max |
| 500_tamu_api | 231 | 4.2321e4 | 893 | 1.06% | 43 | 4.2412e4 | 50.3 | 0.85% | max |
| 500_tamu | 496 | – | – | – | 31 | 7.2396e4 | 410 | 0.25% | min |
| 1888_rte | 378 | – | – | – | 27 | 1.3953e6 | 934 | 0.51% | min |

# CHAPTER 4
# Research perspectives

My long-term research goal is to embed polynomial optimization techniques in certain academic and industrial frameworks, in particular the fields of quantum information theory and free probability, energy networks (optimal power flow) and deep learning, where several hard, but highly promising real-world challenges remain to be tackled.

To reach this ultimate aim, I will consider the three main challenging goals:

(G1) Applications to quantum information and free probability;

(G2) Applications to optimal power flow;

(G3) Applications to deep learning.

## G1: quantum information and free probabilities

The directions mentioned here relate to my bilateral French-Slovenian research project QUANTPOP (acronym standing for "quantum information with noncommutative polynomial optimization"), funded by the PHC Proteus (Partenariat Hubert Curien). We intend to conduct this research, jointly with experts from real algebraic geometry from University of Ljubljana and physics researchers from ICFO Barcelona.

The starting point of this project has been the collaboration outlined in Section 3.3 to design appropriate schemes to take into account sparsity involved in noncommutative optimization problems. Apart from sparsity, we intend to pursue research to take into account other properties of structured noncommutative problems, such as symmetry. The underlying motivation is to tackle important applications, including the computation of quantum graph parameters or maximum violation bounds of Bell inequalities in quantum information theory. Here is a non-exhaustive list of potential problems involving sparse and symmetric noncommutative functions in quantum information:

- **Quantum games**, in particular the problem of finding the number of mutually unbiased bases in dimension 6, open for several decades [2] (symmetric)

- Lower bounding the ground state **energy of many body Hamiltonians**, which is a fundamental problem with potential applications to chemistry and quantum simulators [26] (sparse, some are symmetric)

- **Inflation for quantum correlations** in networks [241] (sparse and symmetric)

- **Device independent quantum key distribution** [285] (symmetric)

**Trace polynomials.** So far, prior research in quantum information theory focused intensively on reformulating problems as eigenvalue optimization of noncommutative polynomials. One famous application is to characterize the set of quantum correlations. Our goal is to derive approximation schemes for other problems of interest arising in quantum information problems. An important one is to investigate asymptotic limits of output states of tensor products of random quantum channels. A random quantum channel is a communication channel which can transmit quantum information, as well as classical information (an example of quantum information is the state of a qubit). In [93], the authors consider output state limits by computing bounds of generalized traces of tensors. Another recent work [241], focuses on the set of quantum correlations in networks, and formulate the problem in the framework of constrained trace optimization. They obtain a hierarchy of convex relaxations of then incorporate the initial constraints into the semidefinite matrix defined in the Navascués, Pironio and Acín (NPA) hierarchy. The resulting extended NPA hierarchy allows to successfully identify correlations not attainable in the entanglement-swapping scenario. Other applications include optimal entanglement witness for multi-partite Werner states [142]. With the framework considered in Section 1.5 one can optimize over trace polynomials when the objective cost only involves sums of trace products (pure trace polynomials). A topic of future research is to derive a hierarchy of primal-dual SDP programs converging to the minimal eigenvalue of a trace polynomial under trace polynomial inequality constraints. Another practical goal relates to the implementation of the optimization algorithms from Section 1.5 in a modeling toolbox and incorporate it within the NCTSSOS software, dedicated to the modeling of sparse noncommutative polynomial optimization problems.

**Exploiting the structure of noncommutative optimization problems.** Sharing the same drawbacks as the classical moment-SOS hierarchy, our tracial framework will be limited to optimization problems involving a modest number of variables. To overcome this scalability issue, we intend to focus on exploiting structural properties of the input data. One possibility is to extend both frameworks exploiting correlative sparsity [J3] and term sparsity from [R13] to optimization problems involving sparse trace polynomials, which could lead to save even more computational cost as the number of involved terms happens to be larger than in the commutative setting. An alternative relaxation scheme allows us to compute lower bounds of nonnegative polynomials, which are sums of nonnegative circuits (SONC), as seen in Section 2.4. In the context of J. Wang's postdoctoral supervision, we have recently proved that finding lower bounds of nonnegative polynomials via SONC boils down to solving second-order conic programs [C14, R7], a special type of convex optimization problem. This approach allow significantly faster computation of lower bounds than via semidefinite relaxations. One promising track of research would be to define an analogous notion of SONC polynomials in the noncommutative setting and to find a way to check their positivity via efficient relaxation schemes, such as second-order conic programming. Another way to reduce the size limitation is to exploit symmetries when present in the problem definition [251]. Classical invariant theory studies polynomials that are preserved under linear group actions. In the context of polynomial optimization, if one assumes that that all input polynomial data are invariant under a linear group action, then [251] shows that one can obtain a block diagonalization of the related SDP. Assuming invariance under the full symmetric group, the so-called degree principle can be used to transform the initial problem into a set of lower dimensional problems such that the resulting relaxations have *finite convergence*. Recent work [156] focused on the noncommutative analogue of this invariant theory in the context of rational functions, yielding positivity certificates for invariant rational functions in terms of SOS of invariants. One of the goals of this task is to benefit from these results to derive the noncommutative analogue of [251] for invariant POP. The subtle difference with the commutative case is that it shall be required to rely on rational functions to index the block diagonal semidefinite matrices involved in the resulting relaxations.

**Noncommutative Christoffel-Darboux kernels.** Algorithms similar to the one from [131] allow one to extract optimizers of eigenvalue or trace minimization problems; see, e.g., [238], [9, Chapter 21], [53, Theorem 1.69] and Section 1.5. A framework based on noncommutative Christoffel-

Darboux kernels could be an alternative to approximate minimizers by relying on the levelsets of such kernels. This shall lead us to study the noncommutative analog of Christoffel-Darboux kernel associated to a certain class of distributions occurring in free probability, and to investigate the related asymptotic properties. One of the equivalent definitions of the Christoffel-Darboux kernel is via a sum of orthonormal polynomials, which has been done in [69] in the noncommutative context. Further explorations of systems of multivariate orthogonal and orthonormal noncommutative polynomials have been undertaken by several authors, for instance [12, 11, 10, 281]. We wish to bring as novelty to this study the structure of operator spaces [234] and the application of the classical analysis of plurisubharmonic functions [107], which we believe has never been considered before in the noncommutative context.

**Upper bounds hierarchies.** A second SOS-based hierarchy proposed in [175] yields a monotone sequence of *upper bounds* which converges to the minimum and therefore can be seen as complementary to the first moment-SOS hierarchy of lower bounds. In addition, and in contrast to the hierarchy of lower bounds, the function to be minimized may not be either a polynomial or a semialgebraic function. At each step of the hierarchy, an upper bound on the minimum of a given polynomial is computed by solving a so-called *generalized eigenvalue* problem. Several efforts have been made to provide convergence rates for the hierarchy of upper bounds. In [158], the authors obtain convergence rates which are no worse than $O(1/\sqrt{r})$ and often match practical experiments. On some specific sets this convergence rate has been improved. For instance, for the box $[-1, 1]^n$ and the sphere, an $O(1/r^2)$ rate of convergence has been obtained in [73] and [157] respectively. For some other cases, in particular convex bodies, an $O(\log^2 r/r^2)$ rate of convergence rates has been recently obtained in [271] and in [190]. All these research efforts show that the asymptotic behavior of the upper bounds hierarchy is better understood than for the lower bounds hierarchy. As for the lower bounds hierarchy, the size of the resulting matrices is critical and restricts its application to small size problems. In practice, there exist very efficient algorithms based on first-order methods to obtain upper bounds for the minimum of polynomials or rational functions. So far the main interest of the upper bound hierarchy has been its theoretical rate of convergence to the global minimum as such guarantees are rather rare. A first attempt to break the curse of dimensionality in the upper bounds hierarchy for polynomial optimization has been done in [178]. The idea is to use the pushforward measure of the Lebesgue measure by the polynomial to minimize. In doing so one reduces the initial problem to a related univariate problem and as a result one obtains a hierarchy of upper bounds (again generalized eigenvalue problems) which involves univariate SOS polynomials of increasing degree. In [R4], we extended this framework to the case of sums of rational functions. By contrast with commutative optimization problems, it is even more challenging to obtain upper bounds for problems in noncommutative variables, such as the ones arising in quantum information. Existing methods include the density matrix renormalization group (DMRG) [304], which is a numerical variational technique devised to obtain the low-energy physics of quantum many-body systems, or quantum variants of Monte-Carlo methods [228]. Our goal here is to propose an alternative of these two families of methods with applications in quantum information. A first attempt has been done in [250] to compute minimal eigenvalues of pure quartic oscillators, but without any convergence guarantees and lack of scalability. We intend to rely on free probabilities to derive a converging hierarchy of upper bounds for eigenvalue and trace optimization problems and apply it to the above-mentioned quantum information problems. Challenges to tackle include the choice of the linear functional (with a priori available moments) in relation with the target application, the noncommutative extension of the framework based on pushforward measures, as well as the analysis of the convergence rate.

## G2: energy networks

Some aforementioned topics of investigation are related to my project FastOPF (acronym standing for "Fast polynomial optimization techniques for optimal power flow"), jointly funded by AMIES (French Agency for mathematics in interaction with industry and society) and the industrial company RTE (France's energy transmission system operator).

**Optimal power flow.** As seen in Section 3.5, alternative current-optimal power flow (AC-OPF) problems can be modeled with polynomial optimization. Several convex relaxations have been recently provided with the goal of solving AC-OPF instances to global optimality. These efforts led to efficient solution algorithms that can solve many instances found in the literature, which model real-world networks. The concurrent methods usually perform costly domain partitioning and spatial branching on continuous variables. Our framework shall overcome these issues by providing fast yet accurate bounds. In the case of OPF, equality constraints involve the voltage, power generation and power flow variables. For many AC-OPF problems, the first-order moment-SOS relaxation (Shor relaxation) is already able to certify global optimality with an optimality gap less than 1.00%. In the recent contribution [R14], outlined in Section 3.5, we have developed a new CS-TSSOS hierarchy to solve large-scale OPF with sparse data. We focused on challenging AC-OPF problems, for which the gap is greater than 1.00%, from the AC-OPF library **pglib** [18]. We simultaneously exploit the few correlations between variables and between terms to solve OPF instances with up to 1 888 buses, yielding POP with more than 4 000 variables and 18 000 constraints. They are solved with a 0.5% optimality gap. On the other hand, we have shown in [R10, R11] that the SDP relaxations in the moment-SOS hierarchy have a nice constant trace property, which can be exploited to avoid solving the relaxations via interior point methods and rather use ad-hoc spectral methods that minimize the largest eigenvalue of a matrix pencil. As a result we obtain a hierarchy of "spectral relaxations". The resulting algorithm is much more efficient than the usual interior-point methods and can handle matrices of size up to 2 000 with more than 1.5 million constraints in less than a hour on a standard laptop. Further perspectives include combining both frameworks to solve AC-OPF instances with up to tens of thousand buses with optimality gap close to 0. We will exploit the sparsity of the input data of POP and the sparsity of the associated spectral relaxations. For this, we will combine the two associated solvers recently developed at LAAS together with J. Wang during his postdoc and N. H. A. Mai during his PhD, namely `TSSOS` and `SpectralPOP`, and execute the resulting software library on large-scale instances [86] with the "pfcalcul" LAAS HPC cluster.

**FIR filters and certified complex polynomial optimization.** Finite impulse response (FIR) filters are now widely used for implementation of smart grid abilities after noise reduction/distortion, in order to obtain improved power quality along with power transfer capability of grid connected energy systems, e.g., solar photovoltaic systems [268]. Such filters can be designed by encoding certain positivity constraints of trigonometric polynomials [83] with linear matrix inequalities, and solving them numerically with semidefinite programming. However, small numerical errors may compromise the input specifications of the implemented filter. To overcome these issues, we intend to extend our current certification framework, presented in Section 2.2 for univariate polynomials and Section 2.3 for multivariate polynomials, to the case of complex polynomials with coefficients having rational real and imaginary parts. We plan to investigate these research directions during the PhD of V. T. Hieu in the context of the POEMA project.

**Stability of large-scale power systems.** One usual way to model power networks is to rely on an interconnection of weakly coupled nodes, while the systems dynamic is driven by generators, which are modeled by closed-loop controlled ordinary differential equations. One way to ensure the stability of large-scale power systems is to approximate from outside backward reachable sets of polynomial ordinary differential equations, with sparse dynamics. Such outer approximations contain the set of all initial conditions ensuring that the system can operate in a safe way. Most of the technical literature on stability analysis for power networks focuses relies on Lyapunov functions computed by nonconvex optimization, e.g., a bilinear variant of polynomial SOS optimization as in [8]. Recent advances have been done in the MAC team to derive SDP hierarchies, including [137] but without convergence guarantees, and [262] with rather strong assumptions on the dynamics. One challenging perspective of research is to broaden the range of analyzable dynamical problems, including large-scale power systems, while maintaining satisfactory convergence guarantees. To reach this goal, we have recently started to develop a tool, called SparseDynamicSystem, to compute forward/backward reachable sets, maximum positively invariant sets, global attractors for polynomial dynamic systems based on the term sparsity adapted moment-SOS hierarchies outlined in Section 3.4 and Section 3.5. Benchmarks of interest include multi-mode dimensional models obtained by projecting the Burgers equation with ordinary diffusion (see [311]) as well as high dimensional powergrid problems from [167].

**Time delay systems.** An additional challenge is that time delays can deteriorate both stability and performance of controllers used for networked power systems [19]. To analyze systems governed by delay differential equation, existing approaches consider Lyapunov-Krasovskii certificates [92], and conversion into transport partial differential equations with an additional time variable, see [91, § 1.4.1]. However, such methods can suffer either from conservatism and lack of convergence guarantees, or numerical issues arising from the underlying discretization schemes. Our idea is to propose an occupation-measure based method, as seen previously in Section 1.2 for analysis and control of continuous-time systems with time delays, by extending the framework dedicated either to polynomial optimal control [189] or to partial differential equations [J15, 211].

## G3: deep learning

Most research tracks mentioned below are related to my participation as a junior member of the ANITI (Artificial and Natural Intelligence Toulouse Institute) chaire "Polynomial optimization for Machine Learning", led by J.-B. Lasserre.

**Lipschitz Constants of ReLU Networks.** In the context of the PhD of T. Chen [C4, R1], we have introduced a sublevel moment-SOS hierarchy where each SDP relaxation can be viewed as an intermediate between the $r$-th and $(r+1)$-th order SDP relaxations of the moment-SOS hierarchy (dense or sparse version). With the flexible choice of determining the size (level) and number (depth) of subsets in the SDP relaxation, one is able to obtain different improvements compared to the $r$-th order relaxation, based on the machine memory capacity. In particular, we obtained promising results for $r = 1$ and various types of problems in deep learning, including robustness certification and Lipschitz constant of neural networks, where the standard moment-SOS hierarchy (or its sparse variant) is computationally intractable. Our sublevel strategy has been designed for quadratically constrained quadratic problems modeling single hidden layer networks. As the number of layers increases, the degree of the objective function also increases and the approach must be combined with lifting in order to deal with higher-degree objective polynomials. Efficient derivation of approximate sparse certificates for high degree polynomials should allow to enlarge the spectrum of applicability of such techniques to larger size networks and broader classes of activation functions.

**Data-driven techniques.** Recent research [187] investigated the ability of Christoffel-Darboux kernels to capture information about the support of an unknown probability measure. A distinguishing feature of this approach is to allow one to infer support characteristics, based on the knowledge of finitely many moments of the underlying measure. In Section 1.3, we relied on such kernels to approximate the support of singular continuous invariant measures. A major open question remains whether this approach can be used in a data-driven setting, where the underlying model is unknown and only observed data are available. We will investigate this direction, building on the recent work [163]. Progress in this direction would be an enabling factor in bringing the elegant and powerful tools of the moment-SOS hierarchy to the realm of the present-day big-data applications, which are currently typically tackled using ad-hoc heuristic techniques with limited mathematical foundation. We will develop new methods furnished with a theoretical analysis, including convergence rate and non-asymptotic out-of-sample error. One first possible investigation track will consist of studying Christoffel-Darboux kernels to extend the approach from [235] for measures supported on specific classes of mathematical varieties. We intend to apply this framework to deep learning network models, for which latent representation correspond to such low-dimensional varieties, including MNIST, CIFAR10 or fashion MNIST. Other main steps will include the investigation of adaptive sampling techniques and basis choice for the approach developed in [163] as well as extension of the proposed methodology beyond the considered class of problems, e.g., to data-driven optimal control.

**Stability Analysis of Recurrent Neural Networks.** To prove the stability of recurrent neural networks with ReLU activation functions, we started to focus in [C5] on the positive $\ell_2$ induced norm of discrete-time linear time-invariant systems, where the input signals are restricted to be nonnegative. We have provided tractable methods based on copositive programming to get enclosure bounds of this norm. However, the treatment was primitive and hence conservative. In this respect, [183] has already shown how to construct a hierarchy of SDP to solve copositive programs in an asymptotically exact fashion. Nevertheless, this approach does not allow us to handle practical size problems since the size of SDP grows very rapidly. We need further effort to reduce computational burden for instance by finding out sparsity structure. We plan to rely on efficient first-order methods to solve the specific conic relaxations arising from POP with sphere constraints [R10]. The used small-gain type treatment for the stability analysis of RNN might be too shallow in view of advanced integral quadratic constraint (IQC) theory [215]. Namely, for the stability analysis of feedback systems constructed from a linear time invariant system and nonlinear elements (i.e., Lurye systems), the effectiveness of the IQC approach with Zames-Falb multipliers [312] is widely recognized, see, e.g., [88, 89]. Therefore it would be strongly preferable to build a new copositive programming-based approach relying on the powerful IQC framework. To this end, we need to explore sound ways to capture the properties of nonlinear elements exhibiting positivity, such as ReLU, by introducing copositive multipliers and incorporate them into existing IQC conditions. It is also important to seek for possible ways to introduce copositive multipliers to deal with saturated systems on the basis of the techniques developed for their analysis and synthesis [286].

**Computer-assisted stability proofs.** Another important certification goal is to provide computer-assisted proofs for the stability of systems whose controllers have been designed with deep neural networks. A classical way to show stability of a given continuous/discrete-time polynomial system is by proving that a given function is a Lyapunov function. We intend to obtain formal proofs of polynomial nonnegativity, which shall be handled with SOS certificates. Since proof assistants have computational limitation, we can rely on external tools that produce certificates, as seen in Chapter 2, whose checking is reasonably easier from a computational point of view. In [R2], we have provided such certificates with the RealCertify [C9] tool available outside of the proof assistant MinLog [36] and verified inside. An interesting track of further research is to derive formally certified outer/inner approximations of sets of interest arising in the context of dynamical systems, such as backward/forward reachable sets or maximal invariants.

**Robust and stochastic polynomial optimization for deep learning.** Deep learning algorithms are now embedded in automatic procedures for smart sensing and networks (grids). Robust polynomial optimization and stochastic optimization can be used for air quality monitoring where the optimization problems have uncertainties due to grid parametrization, model physics and boundary conditions. With some loss of precision in the optimizer, one could in principle reduce the NP-hard POP into a stochastic algorithm that computes in polynomial time. Robust polynomial optimization [173, 171, 216] and stochastic optimization shall be used to improve the efficiency of multi-energy smart grids and mass rapid transit (MRT). Two types of optimizations could be solved: (1) the optimal integration of microgrids, energy hubs, and decentralized energy resources into the core grid, (2) the optimal coordination among renewable energy technologies such as wind, solar and hydropower. In MRT systems, the components to be optimized are rolling stock, power management, signaling, and stations. The optimization algorithms will also serve to coordinate the power grid and MRT system for robust planning of power supply to MRT.

# Exhaustive list of publications

## International peer-reviewed journals

[J1] A. Adjé, P.-L. Garoche, and V. Magron. A Sums-of-Squares extension of policy iterations. *Nonlinear Analysis: Hybrid Systems* 25 (2017), pp. 60–78.

[J2] T. Hales et al. "A formal proof of the Kepler conjecture". *Forum of mathematics, Pi.* Vol. 5. Cambridge University Press. 2017.

[J3] I. Klep, V. Magron, and J. Povh. Sparse noncommutative polynomial optimization. *Mathematical Programming* (2021), pp. 1–41.

[J4] J. B. Lasserre and V. Magron. Computing the Hausdorff Boundary Measure of Semialgebraic Sets. *SIAM Journal on Applied Algebra and Geometry* 4.3 (2020), pp. 441–469.

[J5] J.-B. Lasserre and V. Magron. In SDP relaxations, inaccurate solvers do robust optimization. *SIAM Journal on Optimization* 29.3 (2019), pp. 2128–2145.

[J6] J.-B. Lasserre and V. Magron. Optimal Data Fitting: A Moment Approach. *SIAM Journal on Optimization* 28.4 (2018), pp. 3127–3144.

[J7] V. Magron. Error bounds for polynomial optimization over the hypercube using Putinar type representations. English. *Optimization Letters* 8.7 (2014), pp. 1–9.

[J8] V. Magron, M. Forets, and D. Henrion. Semidefinite Approximations of Invariant Measures for Polynomial Systems. *Discrete & Continuous Dynamical Systems - B* 24.1531-3492_2019_12_6745 (2019), p. 6745.

[J9] V. Magron, D. Henrion, and J.-B. Lasserre. Approximating Pareto curves using semidefinite relaxations. *Operations Research Letters* 42.6–7 (2014), pp. 432–437.

[J10] V. Magron, D. Henrion, and J.-B. Lasserre. Semidefinite Approximations of Projections and Polynomial Images of SemiAlgebraic Sets. *SIAM Journal on Optimization* 25.4 (2015), pp. 2143–2164.

[J11] V. Magron, A. Rocca, and T. Dang. Certified Roundoff Error Bounds Using Bernstein Expansions and Sparse Krivine-Stengle Representations. *IEEE Transactions on Computers* 68.7 (2019), pp. 953–966.

[J12] V. Magron et al. Formal proofs for Nonlinear Optimization. *Journal of Formalized Reasoning* 8.1 (2015), pp. 1–24.

[J13] V. Magron. Interval Enclosures of Upper Bounds of Roundoff Errors Using Semidefinite Programming. *ACM Trans. Math. Softw.* 44.4 (June 2018), 41:1–41:18.

[J14] V. Magron, G. Constantinides, and A. Donaldson. Certified roundoff error bounds using semidefinite programming. *ACM Trans. Math. Software* 43.4 (2017), Art. 34, 31.

[J15] V. Magron and C. Prieur. Optimal control of linear PDEs using occupation measures and SDP relaxations. *IMA Journal of Mathematical Control and Information* 37.1 (2020), pp. 159–174.

[J16] V. Magron and M. Safey El Din. On Exact Reznick, Hilbert-Artin and Putinar's Representations. *Journal of Symbolic Computation* (2021). Accepted for publication.

[J17] V. Magron, M. Safey El Din, and M. Schweighofer. Algorithms for weighted sum of squares decomposition of non-negative univariate polynomials. *Journal of Symbolic Computation* 93 (2019), pp. 200–220.

[J18] V. Magron et al. Certification of real inequalities: templates and sums of squares. English. *Mathematical Programming* 151.2 (2015), pp. 477–506.

[J19] V. Magron et al. Semidefinite approximations of reachable sets for discrete-time polynomial systems. *SIAM Journal on Control and Optimization* 57.4 (2019), pp. 2799–2820.

[J20] N. H. A. Mai, J.-B. Lasserre, and V. Magron. Positivity certificates and polynomial optimization on non-compact semialgebraic sets. *Mathematical Programming* (2021). Accepted for publication.

[J21] J. Wang, V. Magron, and J.-B. Lasserre. Chordal-TSSOS: a moment-SOS hierarchy that exploits term sparsity with chordal extension. *SIAM Journal on Optimization* 31.1 (2021), pp. 114–141.

[J22] J. Wang, V. Magron, and J.-B. Lasserre. TSSOS: A Moment-SOS hierarchy that exploits term sparsity. *SIAM Journal on Optimization* 31.1 (2021), pp. 30–58.

## Publications in the peer-reviewed proceedings of international conferences

[C1] A. Adjé, P.-L. Garoche, and V. Magron. “Property-based Polynomial Invariant Generation using Sums-of-Squares Optimization”. English. *Static Analysis Symposium (SAS)*. Lecture Notes in Computer Science. Springer International Publishing, 2015.

[C2] X. Allamigeon et al. “Certification of Bounds of Non-linear Functions: the Templates Method”. English. *Intelligent Computer Mathematics*. Ed. by J. Carette et al. Vol. 7961. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013, pp. 51–65.

[C3] X. Allamigeon et al. “Certification of Inequalities involving Transcendental Functions: combining SDP and Max-plus Approximation”. *Proceedings of the European Control Conference (ECC) Zurich*. 2013, 2244–2250.

[C4] T. Chen et al. “Semialgebraic Optimization for Bounding Lipschitz Constants of ReLU Networks”. Accepted for publication. 2020.

[C5] Y. Ebihara et al. “*l*_2 Induced Norm Analysis of Discrete-Time LTI Systems for Nonnegative Input Signals and Its Application to Stability Analysis of Recurrent Neural Networks”. *Proceedings of the 2021 European Control Conference*. Accepted for publication. 2021.

[C6] V. Magron. “NLCertify: A Tool for Formal Nonlinear Optimization”. English. *Mathematical Software – ICMS 2014*. Ed. by H. Hong and C. Yap. Vol. 8592. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2014, pp. 315–320.

[C7] V. Magron, S. Sugimoto, and S. Yoshimura. “Time Dependent Magnetic Structural Coupled Analysis of MRI Model with PC Cluster”. *Proceedings of the Conference on Computational Engineering and Science*. Vol. 15. CCES, 2010, 757–760.

[C8] V. Magron and M. Safey El Din. “On Exact Polya and Putinar’s Representations”. *ISSAC’18: Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation*. New-York, US: ACM, 2018.

[C9] V. Magron and M. Safey El Din. “RealCertify: a Maple package for certifying nonnegativity”. *ISSAC’18: Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation*. New York, NY, USA, 2018.

[C10] V. Magron, H. Seidler, and T. de Wolff. “Exact Optimization via Sums of Nonnegative Circuits and Arithmetic-geometric-mean-exponentials”. *Proceedings of the 2019 on International Symposium on Symbolic and Algebraic Computation*. 2019, pp. 291–298.

[C11] A. Rocca, V. Magron, and T. Dang. "Certified Roundoff Error Bounds using Bernstein Expansions and Sparse Krivine-Stengle Representations". *24th IEEE Symposium on Computer Arithmetic, **Best Paper Award***. IEEE, 2017.

[C12] A. Rocca et al. "Occupation measure methods for modelling and analysis of biological hybrid systems". Vol. 51. 16. 6th IFAC Conference on Analysis and Design of Hybrid Systems ADHS 2018. 2018, pp. 181–186.

[C13] J. Wang, M. Maggio, and V. Magron. "SparseJSR: A Fast Algorithm to Compute Joint Spectral Radius via Sparse SOS Decompositions". Accepted for publication. 2021.

[C14] J. Wang and V. Magron. "A second order cone characterization for sums of nonnegative circuits". *Proceedings of the 45th International Symposium on Symbolic and Algebraic Computation*. 2020, pp. 450–457.

## Articles under submission, research reports

[R1] T. Chen et al. A Sublevel Moment-SOS Hierarchy for Polynomial Optimization. *arXiv preprint arXiv:2101.05167* (2021).

[R2] G. Devadze, V. Magron, and S. Streif. Computer-assisted proofs for Lyapunov stability via Sums of Squares certificates and Constructive Analysis. *arXiv preprint arXiv:2006.09884* (2020).

[R3] I. Klep, V. Magron, and J. Volčič. Optimization over trace polynomials. *arXiv preprint arXiv:2006.12510* (2020).

[R4] J.-B. Lasserre et al. Minimizing rational functions: a hierarchy of approximations via pushforward measures. *arXiv preprint arXiv:2012.05793* (2020).

[R5] V. Magron and J. Wang. TSSOS: a Julia library to exploit sparsity for large-scale polynomial optimization. *arXiv preprint arXiv:2103.00915* (2021).

[R6] V. Magron. "Formal Proofs for Global Optimization–Templates and Sums of Squares". PhD thesis. Ecole Polytechnique X, 2013.

[R7] V. Magron and J. Wang. SONC Optimization and Exact Nonnegativity Certificates via Second-Order Cone Programming. *arXiv preprint arXiv:2012.07903* (2020).

[R8] N. H. A. Mai, A. Bhardwaj, and V. Magron. The Constant Trace Property in Noncommutative Optimization. *arXiv preprint arXiv:2102.02162* (2021).

[R9] N. H. A. Mai, V. Magron, and J.-B. Lasserre. A sparse version of Reznick's Positivstellensatz. *arXiv preprint arXiv:2002.05101* (2020).

[R10] N. H. A. Mai, V. Magron, and J.-B. Lasserre. A hierarchy of spectral relaxations for polynomial optimization. *arXiv preprint arXiv:2007.09027* (2020).

[R11] N. H. A. Mai et al. Exploiting constant trace property in large-scale polynomial optimization. *arXiv preprint arXiv:2012.08873* (2020).

[R12] N. Vreman et al. Stability of Control Systems under Extended Weakly-Hard Constraints. *arXiv preprint arXiv:2101.11312* (2021).

[R13] J. Wang and V. Magron. Exploiting term sparsity in noncommutative polynomial optimization. *arXiv preprint arXiv:2010.06956* (2020).

[R14] J. Wang et al. CS-TSSOS: Correlative and term sparsity for large-scale polynomial optimization. *arXiv preprint arXiv:2005.02828* (2020).

# Bibliography

[1] M. R. Abdalmoaty, D. Henrion, and L. Rodrigues. “Measures and LMIs for optimal control of piecewise-affine systems”. *2013 European Control Conference (ECC)*. 2013, pp. 3173–3178.

[2] E. A. Aguilar et al. Connections between Mutually Unbiased Bases and Quantum Random Access Codes. *Phys. Rev. Lett.* 121 (5 2018), p. 050501.

[3] A. A. Ahmadi and A. Majumdar. “DSOS and SDSOS optimization: LP and SOCP-based alternatives to sum of squares optimization”. *2014 48th annual conference on information sciences and systems (CISS)*. IEEE. 2014, pp. 1–5.

[4] L. Ambrosio, N. Fusco, and D. Pallara. *Functions of bounded variation and free discontinuity problems*. Oxford mathematical monographs. Clarendon Press, 2000.

[5] P. R. Amestoy, T. A. Davis, and I. S. Duff. An approximate minimum degree ordering algorithm. *SIAM Journal on Matrix Analysis and Applications* 17.4 (1996), pp. 886–905.

[6] H. Ammari et al. Identification of an algebraic domain in two dimensions from a finite number of its generalized polarization tensors. *Mathematische Annalen* 375.3-4 (2019), pp. 1337–1354.

[7] E. D. Andersen and K. D. Andersen. “The Mosek Interior Point Optimizer for Linear Programming: An Implementation of the Homogeneous Algorithm”. English. *High Performance Optimization*. Ed. by H. Frenk et al. Vol. 33. Applied Optimization. Springer US, 2000, pp. 197–232.

[8] M. Anghel, F. Milano, and A. Papachristodoulou. Algorithmic construction of Lyapunov functions for power system stability analysis. *IEEE Transactions on Circuits and Systems I: Regular Papers* 60.9 (2013), pp. 2533–2546.

[9] M. F. Anjos and J.-B. Lasserre, eds. *Handbook on semidefinite, conic and polynomial optimization*. Vol. 166. International Series in Operations Research & Management Science. Springer, New York, 2012, pp. xii+960.

[10] M. Anshelevich. Monic non-commutative orthogonal polynomials. *Proceedings of the American Mathematical Society* 136.7 (2008), pp. 2395–2405.

[11] M. Anshelevich. Orthogonal polynomials with a resolvent-type generating function. *Transactions of the American Mathematical Society* 360.8 (2008), pp. 4125–4143.

[12] M. Anshelevich. “Product-type non-commutative polynomial states”. *Noncommutative harmonic analysis with applications to probability. II*. Vol. 89. Banach Center Publications. Institute of Mathematics, Polish Academy of Sciences, WARSZAWA, 2010.

[13] R. B. Ash. *Real Analysis and Probability*. New York: Academic Press, 1972.

[14] P. Aston and O. Junge. Computing the invariant measure and the Lyapunov exponent for one-dimensional maps using a measure-preserving polynomial basis. *Mathematics of Computation* 83.288 (2014), pp. 1869–1902.

[15] S. Ayupov, A. Rakhimov, and S. Usmanov. *Jordan, real and Lie structures in operator algebras*. Vol. 418. Mathematics and its Applications. Kluwer Academic Publishers Group, Dordrecht, 1997, pp. x+225.

[16] B. Reznick. Extremal PSD forms with few terms. *Duke Mathematical Journal* 45.2 (1978), pp. 363–374.

[17] B. Reznick. Uniform denominators in Hilbert's seventeenth problem. *Mathematische Zeitschrift* 220.1 (1995), pp. 75–97.

[18] S. Babaeinejadsarookolaee et al. The power grid library for benchmarking ac optimal power flow algorithms. *arXiv preprint arXiv:1908.02788* (2019).

[19] B. Bamieh and D. F. Gayme. "The price of synchrony: Resistive losses due to phase synchronization in power networks". *2013 American Control Conference*. IEEE. 2013, pp. 5815–5820.

[20] B. Bank et al. Generalized polar varieties: Geometry and algorithms. *Journal of complexity* (2005).

[21] B. Bank et al. Intrinsic complexity estimates in polynomial optimization. *Journal of Complexity* 30.4 (2014), pp. 430–443.

[22] B. Bank et al. On the geometry of polar varieties. *Applicable Algebra in Engineering, Communication and Computing* 21.1 (2010), pp. 33–83.

[23] B. Bank et al. On the intrinsic complexity of point finding in real singular hypersurfaces. *Information Processing Letters* 109.19 (2009), pp. 1141–1144.

[24] B. Bank et al. Polar varieties and efficient real elimination. *Mathematische Zeitschrift* 238.1 (2001), pp. 115–144.

[25] B. Bank et al. Polar varieties and efficient real equation solving: the hypersurface case. *Journal of Complexity* 13.1 (1997), pp. 5–27.

[26] T. Barthel and R. Hübener. Solving Condensed-Matter Ground-State Problems by Semidefinite Relaxations. *Phys. Rev. Lett.* 108 (20 2012), p. 200404.

[27] A. Barvinok. *A Course in Convexity*. Graduate studies in mathematics. American Mathematical Society, 2002.

[28] S. Basu, R. Pollack, and M.-F. Roy. "A new algorithm to find a point in every cell defined by a family of polynomials". *Quantifier elimination and cylindrical algebraic decomposition*. Springer-Verlag, 1998.

[29] S. Basu, R. Pollack, and M.-F. Roy. *Algorithms in Real Algebraic Geometry (Algorithms and Computation in Mathematics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[30] S. Basu, R. Pollack, and M.-F. Roy. On the combinatorial and algebraic complexity of quantifier elimination. *Journal of the ACM (JACM)* 43.6 (1996), pp. 1002–1045.

[31] J. S. Bell. On the Einstein Podolsky Rosen paradox. *Physics Physique Fizika* 1.3 (1964), p. 195.

[32] M. A. Ben Sassi et al. Linear relaxations of polynomial positivity for polynomial Lyapunov function synthesis. *IMA Journal of Mathematical Control and Information* (2015).

[33] M. A. Ben Sassi et al. Reachability Analysis of Polynomial Systems Using Linear Programming Relaxations. *ATVA 2012* (2012). Ed. by S. Chakraborty and M. Mukund, pp. 137–151.

[34] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust Optimization*. Princeton Series in Applied Mathematics. Princeton University Press, 2009.

[35] A. Ben-Tal and A. S. Nemirovski. *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2001.

[36] U. Berger et al. "Minlog-a tool for program extraction supporting algebras and coalgebras". *International Conference on Algebra and Coalgebra in Computer Science*. Springer. 2011, pp. 393–399.

[37] O. Bernard and J.-L. Gouzé. Global qualitative description of a class of nonlinear dynamical systems. *Artificial Intelligence* 136.1 (2002), pp. 29–59.

[38] D. Bertsekas. Infinite time reachability of state-space regions by using feedback control. *IEEE Transactions on Automatic Control* 17.5 (1972), pp. 604–613.

[39] M. Berz and K. Makino. "Rigorous global search using Taylor models". *Proceedings of the 2009 conference on Symbolic numeric computation*. SNC '09. Kyoto, Japan: ACM, 2009, pp. 11–20.

[40] D. Bessis, P. Moussa, and M. Villani. Monotonic converging variational approximations to the functional integrals in quantum statistical mechanics. *J. Math. Phys.* 16.11 (1975), pp. 2318–2325.

[41] B. E. Blackadar. Weak expectations and nuclear $C^*$-algebras. *Indiana Univ. Math. J.* 27.6 (1978), pp. 1021–1026.

[42] J. R. Blair and B. Peyton. "An introduction to chordal graphs and clique trees". *Graph theory and sparse matrix computation*. Springer, 1993, pp. 1–29.

[43] F. Blanchini. Set invariance in control. *Automatica* 35.11 (1999), pp. 1747–1767.

[44] F. Blanchini and S. Miani. *Set-Theoretic Methods in Control*. Systems & Control: Foundations & Applications. Birkhäuser Boston, 2007.

[45] M. Borges et al. "Symbolic Execution with Interval Solving and Meta-heuristic Search". *Proceedings of the 2012 IEEE Fifth International Conference on Software Testing, Verification and Validation*. ICST '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 111–120.

[46] F. Boudaoud, F. Caruso, and M.-F. Roy. Certificates of Positivity in the Bernstein Basis. *Discrete & Computational Geometry* 39.4 (2008), pp. 639–655.

[47] S. Boyd et al. *Linear Matrix Inequalities in System and Control Theory*. Vol. 15. Studies in Applied Mathematics. Philadelphia, PA: SIAM, June 1994.

[48] F. Bréhard, M. Joldes, and J.-B. Lasserre. "On moment problems with holonomic functions". *Proceedings of the 2019 on International Symposium on Symbolic and Algebraic Computation*. 2019, pp. 66–73.

[49] P. Breiding and O. Marigliano. Random points on an algebraic manifold. *SIAM Journal on Mathematics of Data Science* 2.3 (2020), pp. 683–704.

[50] S. Bromberger, J. Fairbanks, et al. *JuliaGraphs/LightGraphs. jl: an optimized graphs package for the Julia programming language*. 2017.

[51] M. Brubaker, M. Salzmann, and R. Urtasun. "A family of MCMC methods on implicitly defined manifolds". *Artificial intelligence and statistics*. 2012, pp. 161–172.

[52] B. Büeler, A. Enge, and K. Fukuda. "Exact volume computation for polytopes: a practical study". *Polytopes—combinatorics and computation*. Springer. 2000, pp. 131–154.

[53] S. Burgdorf, I. Klep, and J. Povh. *Optimization of polynomials in non-commuting variables*. SpringerBriefs in Mathematics. Springer, [Cham], 2016, pp. xv+104.

[54] S. Burgdorf et al. The tracial moment problem and trace-optimization of polynomials. *Math. Program.* 137.1-2, Ser. A (2013), pp. 557–578.

[55] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa. The Quickhull Algorithm for Convex Hulls. *ACM Trans. Math. Softw.* 22.4 (1996), pp. 469–483.

[56] C. Muńoz and A. Narkawicz. Formalization of Bernstein Polynomials and Applications to Global Optimization. *J. Aut. Reasoning* 51.2 (2013), pp. 151–196.

[57] K. Cafuta, I. Klep, and J. Povh. Constrained polynomial optimization problems with noncommuting variables. *SIAM J. Optim.* 22.2 (2012), pp. 363–383.

[58] K. Cafuta, I. Klep, and J. Povh. NCSOStools: a computer algebra system for symbolic and numerical computation with noncommutative polynomials. *Optim. Methods Softw.* 26.3 (2011), pp. 363–380.

[59] J. Canny. *The complexity of robot motion planning*. MIT press, 1988.

[60] G. Chesi. *Domain of attraction: analysis and control via SOS programming*. Vol. 415. Springer Science & Business Media, 2011.

[61] S. Chevillard et al. Efficient and accurate computation of upper bounds of approximation errors. *Theoretical Computer Science* 412.16 (2011). https://hal.archives-ouvertes.fr/ensl-00445343v2, pp. 1523–1543.

[62] W.-F. Chiang et al. "Efficient Search for Inputs Causing High Floating-point Errors". *Proceedings of the 19th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*. PPoPP '14. Orlando, Florida, USA: ACM, 2014, pp. 43–52.

[63] E. B. Chin, J.-B. Lasserre, and N. Sukumar. Numerical integration of homogeneous functions on convex and nonconvex polygons and polyhedra. *Comput. Mech.* 56 (2015), pp. 967–981.

[64] M.-D. Choi, T. Y. Lam, and B. Reznick. "Sums of squares of real polynomials". *Proceedings of Symposia in Pure mathematics*. Vol. 58. American Mathematical Society. 1995, pp. 103–126.

[65] J. F. Clauser et al. Proposed experiment to test local hidden-variable theories. *Phys. rev. lett.* 23.15 (1969), p. 880.

[66] J. Cohen and T. Hickey. Two algorithms for determining volumes of convex polyhedra. *Journal of the ACM (JACM)* 26.3 (1979), pp. 401–414.

[67] G. E. Collins. "Quantifier elimination for real closed fields by cylindrical algebraic decompostion". *Automata Theory and Formal Languages 2nd GI Conference Kaiserslautern, May 20–23, 1975*. Springer. 1975, pp. 134–183.

[68] A. R. Conn, N. I. M. Gould, and P. L. Toint. Testing a class of methods for solving minimization problems with simple bounds on the variables. *Math. Comp.* 50.182 (1988), pp. 399–430.

[69] T. Constantinescu. Orthogonal polynomials in several variables. I. *arXiv:math/0205333v1 [math.FA]* (2002).

[70] R. Cools. Constructing cubature formulae: the science behind the art. *Acta numerica* 6 (1997), pp. 1–54.

[71] R. E. Curto and L. A. Fialkow. Flat extensions of positive moment matrices: recursively generated relations. *Mem. Amer. Math. Soc.* 136.648 (1998), pp. x+56.

[72] E. Darulova and V. Kuncak. "Sound Compilation of Reals". *Proceedings of the 41st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*. POPL '14. San Diego, California, USA: ACM, 2014, pp. 235–248.

[73] E. De Klerk, R. Hess, and M. Laurent. Improved convergence rates for Lasserre-type hierarchies of upper bounds for box-constrained polynomial optimization. *SIAM Journal on Optimization* 27.1 (2017), pp. 347–367.

[74] M. Dellnitz, G. Froyland, and O. Junge. "The algorithms behind GAIO—Set oriented numerical methods for dynamical systems". *Ergodic theory, analysis, and efficient simulation of dynamical systems*. Springer, 2001, pp. 145–174.

[75] M. Dellnitz, S. Klus, and A. Ziessler. A Set-Oriented Numerical Approach for Dynamical Systems with Parameter Uncertainty. *SIAM Journal on Applied Dynamical Systems* 16.1 (2017), pp. 120–138.

[76] M. Dellnitz et al. Exploring invariant sets and invariant measures. *CHAOS: An Interdisciplinary Journal of Nonlinear Science* 7.2 (1997), pp. 221–228.

[77] D. Delmas et al. "Towards an Industrial Use of FLUCTUAT on Safety-Critical Avionics Software". English. *Formal Methods for Industrial Critical Systems*. Ed. by M. Alpuente, B. Cook, and C. Joubert. Vol. 5825. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2009, pp. 53–69.

[78] J. Demmel. *On floating point errors in Cholesky*. University of Tennessee. Computer Science Department, 1989.

[79] S. Diamond and S. Boyd. CVXPY: A Python-Embedded Modeling Language for Convex Optimization. *Journal of Machine Learning Research* (2016).

[80] P. J. di Dio. The multidimensional truncated Moment Problem: Shape and Gaussian Mixture Reconstruction from Derivatives of Moments. *arXiv preprint arXiv:1907.00790* (2019).

[81] A. C. Doherty et al. "The quantum moment problem and bounds on entangled multiprover games". *2008 23rd Annual IEEE Conference on Computational Complexity*. IEEE. 2008, pp. 199–210.

[82] A. Domahidi, E. Chu, and S. Boyd. "ECOS: An SOCP solver for embedded systems". *European Control Conference (ECC)*. 2013, pp. 3071–3076.

[83] B. Dumitrescu. *Positive trigonometric polynomials and signal processing applications*. Vol. 103. Springer, 2007.

[84] I. Dunning, J. Huchette, and M. Lubin. JuMP: A modeling language for mathematical optimization. *SIAM review* 59.2 (2017), pp. 295–320.

[85] K. J. Dykema. Factoriality and Connes' invariant $T(\mathcal{M})$ for free products of von Neumann algebras. *J. Reine Angew. Math.* 450 (1994), pp. 159–180.

[86] A. Eltved, J. Dahl, and M. S. Andersen. On the robustness and scalability of semidefinite relaxation for optimal power flow problems. *Optimization and Engineering* 21.2 (2020), pp. 375–392.

[87] H. Everett et al. The Voronoi diagram of three lines. *Discrete & Computational Geometry* 42.1 (2009), pp. 94–130.

[88] M. Fetzer and C. W. Scherer. Absolute Stability Analysis of Discrete Time Feedback Interconnections. *IFAC PapersOnline* 50.1 (2017), pp. 8447–8453.

[89] M. Fetzer and C. W. Scherer. Full-block Multipliers for Repeated, Slope-Restricted Scalar Nonlinearities. *International Journal of Robust and Nonlinear Control* 27.17 (2017), pp. 3376–3411.

[90] R. FitzHugh. Impulses and physiological states in theoretical models of nerve membrane. *j-BIOPHYS-J* 1 (1961), pp. 445–466.

[91] E. Fridman. *Introduction to time-delay systems: Analysis and control*. Springer, 2014.

[92] E. Fridman. New Lyapunov–Krasovskii functionals for stability of linear retarded and neutral type systems. *Systems & control letters* 43.4 (2001), pp. 309–319.

[93] M. Fukuda and I. Nechita. Asymptotically well-behaved input states do not violate additivity for conjugate pairs of random quantum channels. *Comm. Math. Phys.* 328.3 (2014), pp. 995–1021.

[94] D. Fulkerson and O. Gross. Incidence matrices and interval graphs. *Pacific journal of mathematics* 15.3 (1965), pp. 835–855.

[95] S. Gao, S. Kong, and E. Clarke. "dReal: An SMT Solver for Nonlinear Theories over the Reals". English. *Automated Deduction – CADE-24*. Ed. by M. Bonacina. Vol. 7898. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013, pp. 208–214.

[96] L. E. Ghaoui, F. Oustry, and H. Lebert. Robust Solutions to uncertain semidefinite programs. *SIAM J. Opt* 9.1 (1998), pp. 33–52.

[97] G. H. Golub and C. F. V. Loan. *Matrix Computations (3rd Ed.)* Baltimore, MD, USA: Johns Hopkins University Press, 1996.

[98] M. C. Golumbic. *Algorithmic graph theory and perfect graphs*. Elsevier, 2004.

[99] N. Gravin et al. The inverse moment problem for convex polytopes. *Discrete & Computational Geometry* 48.3 (2012), pp. 596–621.

[100] A. Greuet and M. Safey El Din. Probabilistic Algorithm for Polynomial Optimization over a Real Algebraic Set. *SIAM Journal on Optimization* 24.3 (2014), pp. 1313–1343.

[101] A. Greuet and M. Safey El Din. Probabilistic Algorithm for Polynomial Optimization over a Real Algebraic Set. *SIAM Journal on Optimization* 24.3 (2014), pp. 1313–1343.

[102] A. Greuet et al. Global optimization of polynomials restricted to a smooth variety using sums of squares. *Journal of Symbolic Computation* 47.5 (2012), pp. 503–518.

[103] S. Gribling, D. De Laat, and M. Laurent. Lower bounds on matrix factorization ranks via noncommutative polynomial optimization. *Foundations of Computational Mathematics* 19.5 (2019), pp. 1013–1070.

[104] S. Gribling, D. de Laat, and M. Laurent. Bounds on entanglement dimensions and quantum graph parameters via noncommutative polynomial optimization. *Math. Program.* 170.1, Ser. B (2018), pp. 5–42.

[105] D. Grigoriev and N. Vorobjov. Solving systems of polynomials inequalities in subexponential time. *Journal of Symbolic Computation* 5 (1988), pp. 37–64.

[106] D. Grimm, T. Netzer, and M. Schweighofer. A note on the representation of positive polynomials with structured sparsity. *Archiv der Mathematik* 89.5 (2007), pp. 399–403.

[107] V. Guedj and A. Zeriahi. *Degenerate Complex Monge-Ampère Equations*. Tracts in Mathematics 26. European Mathematical Society, 2017.

[108] F. Guo, E. L. Kaltofen, and L. Zhi. "Certificates of Impossibility of Hilbert-Artin Representations of a Given Degree for Definite Polynomials and Functions". *Proceedings of the 37th International Symposium on Symbolic and Algebraic Computation*. ISSAC '12. Grenoble, France: ACM, 2012, pp. 195–202.

[109] F. Guo, M. Safey El Din, and L. Zhi. "Global optimization of polynomials using generalized critical values and sums of squares". *Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation*. ISSAC '10. Munich, Germany: ACM, 2010, pp. 107–114.

[110] Q. Guo, M. Safey El Din, and L. Zhi. "Computing Rational Solutions of Linear Matrix Inequalities". *Proceedings of the 38th International Symposium on Symbolic and Algebraic Computation*. ISSAC '13. Boston, Maine, USA: ACM, 2013, pp. 197–204.

[111] Q. Guo, M. Safey El Din, and L. Zhi. "Computing rational solutions of linear matrix inequalities". *Proceedings of the 38th International Symposium on Symbolic and Algebraic Computation*. ISSAC '13. ACM. New York, NY, USA, 2013, pp. 197–204.

[112] N. Gvozdenovic, M. Laurent, and F. Vallentin. Block-diagonal semidefinite programming hierarchies for 0/1 programming. *Oper. Res. Lett.* 37 (2009), pp. 27–31.

[113] D. Hadwin. A noncommutative moment problem. *Proc. Amer. Math. Soc.* 129.6 (2001), pp. 1785–1791.

[114] J. Haglund, K. Ono, and D. G. Wagner. "Theorems and conjectures involving rook polynomials with only real zeros". *Topics in number theory*. Springer, 1999, pp. 207–221.

[115] L. Hajdu and R. Tijdeman. Algebraic aspects of discrete tomography. *Journal fur die Reine und Angewandte Mathematik* 534 (2001), pp. 119–128.

[116] T. C. Hales. "Introduction to the Flyspeck Project". *Mathematics, Algorithms, Proofs*. Ed. by T. Coquand, H. Lombardi, and M.-F. Roy. Dagstuhl Seminar Proceedings 05021. Dagstuhl, Germany, 2006.

[117] T. C. Hales. *The Flyspeck Project*. 2013.

[118] L. Haller et al. "Deciding Floating-Point Logic with Systematic Abstraction". *Formal Methods in Computer-Aided Design (FMCAD)*. 2012, pp. 131–140.

[119] E. J. Hancock and A. Papachristodoulou. "Structured sum of squares for networked systems analysis". *2011 50th IEEE Conference on Decision and Control and European Control Conference*. IEEE. 2011, pp. 7236–7241.

[120] J. Harrison. "HOL Light: A Tutorial Introduction". *FMCAD*. Ed. by M. K. Srivas and A. J. Camilleri. Vol. 1166. Lecture Notes in Computer Science. Springer, 1996, pp. 265–269.

[121] S. M. Harwood and P. I. Barton. Efficient polyhedral enclosures for the reachable set of nonlinear control systems. *Mathematics of Control, Signals, and Systems* 28.1 (2016), pp. 1–33.

[122] P. Heggernes. Minimal triangulations of graphs: A survey. *Discrete Mathematics* 306.3 (2006), pp. 297–317.

[123] J. W. Helton. "Positive" noncommutative polynomials are sums of squares. *Ann. of Math. (2)* 156.2 (2002), pp. 675–694.

[124] J. W. Helton, I. Klep, and S. McCullough. Proper analytic free maps. *J. Funct. Anal.* 260.5 (2011), pp. 1476–1490.

[125] J. W. Helton, I. Klep, and S. McCullough. The convex Positivstellensatz in a free algebra. *Adv. Math.* 231.1 (2012), pp. 516–534.

[126] J. W. Helton and S. A. McCullough. A Positivstellensatz for non-commutative polynomials. *Trans. Amer. Math. Soc.* 356.9 (2004), pp. 3721–3737.

[127] M. Hénon. A two-dimensional mapping with a strange attractor. *Communications in Mathematical Physics* 50.1 (1976), pp. 69–77.

[128] D. Henrion. Semidefinite characterisation of invariant measures for one-dimensional discrete dynamical systems. eng. *Kybernetika* 48.6 (2012), pp. 1089–1099.

[129] D. Henrion and M. Korda. Convex Computation of the Region of Attraction of Polynomial Control Systems. *Automatic Control, IEEE Transactions on* 59.2 (2014), pp. 297–312.

[130] D. Henrion, J. Lasserre, and C. Savorgnan. Approximate Volume and Integration for Basic Semialgebraic Sets. *SIAM Review* 51.4 (2009), pp. 722–743.

[131] D. Henrion and J.-B. Lasserre. Detecting Global Optimality and Extracting Solutions in GloptiPoly. *Positive Polynomials in Control*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 293–310.

[132] D. Henrion, J.-B. Lasserre, and J. Löfberg. GloptiPoly 3: moments, optimization and semidefinite programming. Anglais. *Optimization Methods and Software* 24.4-5 (Aug. 2009), pp. 761–779.

[133] D. Henrion, J.-B. Lasserre, and M. Mevissen. Mean Squared Error Minimization for Inverse Moment Problems. *Applied Mathematics & Optimization* 70.1 (2014), pp. 83–110.

[134] D. Henrion, S. Naldi, and M. Safey El Din. Exact Algorithms for Linear Matrix Inequalities. *SIAM Journal on Optimization* 26.4 (2016), pp. 2512–2539.

[135] D. Henrion and J.-B. Lasserre. Inner approximations for polynomial matrix inequalities and robust stability regions. *IEEE Transactions on Automatic Control* 57.6 (2011), pp. 1456–1467.

[136] D. Henrion, S. Naldi, and M. Safey El Din. SPECTRA–a Maple library for solving linear matrix inequalities in exact arithmetic. *Optimization Methods and Software* 34.1 (2019), pp. 62–78.

[137] D. Henrion et al. Approximating regions of attraction of a sparse polynomial differential system. *arXiv preprint arXiv:1911.09500* (2019).

[138] F. Hiai, R. König, and M. Tomamichel. Generalized log-majorization and multivariate trace inequalities. *Ann. Henri Poincaré* 18.7 (2017), pp. 2499–2521.

[139] N. Higham. *Accuracy and Stability of Numerical Algorithms: Second Edition*. Society for Industrial and Applied Mathematics, 2002.

[140] C. Hillar. Sums of squares over totally real fields are rational sums of squares. *Proceedings of the American Mathematical Society* 137.3 (2009), pp. 921–930.

[141] H. Hong and M. Safey El Din. Variant quantifier elimination. *Journal of Symbolic Computation* 47.7 (2012), pp. 883–901.

[142] F. Huber. Positive Maps and Matrix Contractions from the Symmetric Group. *arXiv preprint arXiv:2002.12887* (2020).

[143] IEEE. IEEE Standard for Floating-Point Arithmetic. *IEEE Std 754-2008* (2008), pp. 1–70.

[144] J. Harrison. "Verifying Nonlinear Real Formulas via Sums of Squares". *Proceedings of the 20th International Conference on Theorem Proving in Higher Order Logics*. TPHOLs'07. Kaiserslautern, Germany: Springer-Verlag, 2007, pp. 102–118.

[145] J. Oxley. *Matroid theory*. Second. Vol. 21. Oxford Graduate Texts in Mathematics. Oxford University Press, Oxford, 2011.

[146] J. Jaśkowiec and N. Sukumar. High-order cubature rules for tetrahedra. *Int. J. Num. Methods Eng.* (2020). to appear.

[147] Z. Ji et al. MIP* = RE. *arXiv preprint arXiv:2001.04383* (2020).

[148] C. Josz and D. Henrion. Strong duality in Lasserre's hierarchy for polynomial optimization. *Optimization Letters* 10.1 (2016), pp. 3–10.

[149] D. Joyner et al. Open Source Computer Algebra Systems: SymPy. *ACM Commun. Comput. Algebra* 45.3/4 (Jan. 2012), pp. 225–234.

[150] E. Kaltofen, Z. Yang, and L. Zhi. "A Proof of the Monotone Column Permanent (MCP) Conjecture for Dimension 4 via Sums-of-squares of Rational Functions". *Proceedings of the 2009 Conference on Symbolic Numeric Computation*. SNC '09. Kyoto, Japan: ACM, 2009, pp. 65–70.

[151] E. L. Kaltofen et al. Exact certification in global polynomial optimization via sums-of-squares of rational functions with rational coefficients. *Journal of Symbolic Computation* 47.1 (2012), pp. 1–15.

[152] E. L. Kaltofen et al. "Exact certification of global optimality of approximate factorizations via rationalizing sums-of-squares with floating point scalars". *Proceedings of the 21st International Symposium on Symbolic and Algebraic computation*. ISSAC '08. ACM. New York, NY, USA, 2008, pp. 155–164.

[153] I. Klep and M. Schweighofer. Connes' embedding conjecture and sums of Hermitian squares. *Adv. Math.* 217.4 (2008), pp. 1816–1837.

[154] I. Klep and Š. Špenko. Free function theory through matrix invariants. *Canad. J. Math.* 69.2 (2017), pp. 408–433.

[155] I. Klep, Š. Špenko, and J. Volčič. Positive trace polynomials and the universal Procesi–Schacher conjecture. *Proc. Lond. Math. Soc.* 117.6 (2018), pp. 1101–1134.

[156] I. Klep et al. Noncommutative rational functions invariant under the action of a finite solvable group. *Journal of Mathematical Analysis and Applications* 490.2 (2020), p. 124341.

[157] E. de Klerk and M. Laurent. Convergence analysis of a Lasserre hierarchy of upper bounds for polynomial minimization on the sphere. *Mathematical Programming* (2020), pp. 1–21.

[158] E. de Klerk, M. Laurent, and Z. Sun. Convergence analysis for Lasserre's measure-based hierarchy of upper bounds for polynomial optimization. *Mathematical Programming A* (2016), pp. 1–30.

[159] E. d. Klerk and F. Vallentin. On the Turing Model Complexity of Interior Point Methods for Semidefinite Programming. *SIAM Journal on Optimization* 26.3 (2016), pp. 1944–1961.

[160] A. W. Knapp. *Basic Algebra*. 1st ed. Birkh&#228;user Basel, 2006.

[161] M. Kontsevich and D. Zagier. "Periods". *Mathematics unlimited-2001 and beyond*. Springer, 2001, pp. 771–808.

[162] M. Korda, D. Henrion, and C. N. Jones. "Convex computation of the maximum controlled invariant set for discrete-time polynomial control systems". *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*. 2013, pp. 7107–7112.

[163] M. Korda. Computing controlled invariant sets from data using convex optimization. *SIAM Journal on Control and Optimization* 58.5 (2020), pp. 2871–2899.

[164] M. Korda, D. Henrion, and C. N. Jones. Inner approximations of the region of attraction for polynomial dynamical systems. *IFAC Proceedings Volumes* 46.23 (2013), pp. 534–539.

[165] M. Korda, D. Henrion, and I. Mezić. Convex computation of extremal invariant measures of nonlinear dynamical systems and Markov processes. *Journal of Nonlinear Science* 31.1 (2021), pp. 1–26.

[166] N. Kryloff and N. Bogoliouboff. La Theorie Generale De La Mesure Dans Son Application A L'Etude Des Systemes Dynamiques De la Mecanique Non Lineaire. *Annals of Mathematics* 38.1 (1937), pp. 65–113.

[167] P. Kundur. Power system stability. *Power system stability and control* (2007), pp. 7–1.

[168] P. Lairez, M. Mezzarobba, and M. Safey El Din. "Computing the volume of compact semialgebraic sets". *ISSAC'19: Proceedings of the 2019 ACM International Symposium on Symbolic and Algebraic Computation*. Beijing, China: ACM, 2019.

[169] E. Landau. Über die Darstellung definiter Funktionen durch Quadrate. ger. *Mathematische Annalen* 62 (1906), pp. 272–285.

[170] S. Laplagne. Facial reduction for exact polynomial sum of squares decomposition. *Mathematics of Computation* 89.322 (2020), pp. 859–877.

[171] R. Laraki and J.-B. Lasserre. Semidefinite programming for min–max problems and games. *Mathematical programming* 131.1 (2012), pp. 305–332.

[172] A. Lasota and M. C. Mackey. *Chaos, Fractals, and Noise : Stochastic Aspects of Dynamics*. Applied Mathematical Sciences. New York: Springer-Verlag, 1994.

[173] J.-B. Lasserre. A "joint+ marginal" approach to parametric polynomial optimization. *SIAM Journal on Optimization* 20.4 (2010), pp. 1995–2022.

[174] J.-B. Lasserre. A New Look at Nonnegativity on Closed Sets and Polynomial Optimization. *SIAM journal on Optimization* 21.3 (2011), pp. 864–885.

[175] J.-B. Lasserre. A new look at nonnegativity on closed sets and polynomial optimization. *SIAM Journal on Optimization* 21.3 (2011), pp. 864–885.

[176] J.-B. Lasserre. A Sum of Squares Approximation of Nonnegative Polynomials. *SIAM Review* 49.4 (2007), pp. 651–669.

[177] J.-B. Lasserre. Borel measures with a density on a compact semi-algebraic set. *Archiv der Mathematik* 101.4 (2013), pp. 361–371.

[178] J.-B. Lasserre. Connecting optimization with spectral analysis of tri-diagonal matrices. *Mathematical Programming* (2020), pp. 1–15.

[179] J.-B. Lasserre. Convergent SDP-Relaxations in Polynomial Optimization with Sparsity. *SIAM Journal on Optimization* 17.3 (2006), pp. 822–843.

[180] J.-B. Lasserre. Global Optimization with Polynomials and the Problem of Moments. *SIAM Journal on Optimization* 11.3 (2001), pp. 796–817.

[181] J.-B. Lasserre. Lebesgue decomposition in action via semidefinite relaxations. *Advances in Computational Mathematics* 42.5 (2016), pp. 1129–1148.

[182] J.-B. Lasserre. *Moments, Positive Polynomials and Their Applications*. Imperial College Press optimization series. Imperial College Press, 2009.

[183] J.-B. Lasserre. New approximations for the cone of copositive matrices and its dual. *Mathematical Programming* 144.1 (2014), pp. 265–276.

[184] J.-B. Lasserre. "The Moment-SOS Hierarchy". *Proc. of the International Congress of Mathematicians (ICM 2018)*. Ed. by M. V. B. Sirakov F. Ney de Sousa. Vol. 4. Rio de Janeiro, Brasil: World Scientific, 2019, pp. 3773–3794.

[185] J.-B. Lasserre. Tractable approximations of sets defined with quantifiers. English. *Mathematical Programming* 151.2 (2015), pp. 507–527.

[186] J.-B. Lasserre. Volume of Sublevel Sets of Homogeneous Polynomials. *SIAM Journal on Applied Algebra and Geometry* 3.2 (2019), pp. 372–389.

[187] J.-B. Lasserre and E. Pauwels. The empirical Christoffel function with applications in data analysis. *Advances in Computational Mathematics* 45.3 (2019), pp. 1439–1468.

[188] J.-B. Lasserre and E. S. Zeron. Solving a class of multivariate integration problems via Laplace techniques. *Applicationes Mathematicae* 28 (2001), pp. 391–405.

[189] J.-B. Lasserre et al. Nonlinear optimal control via occupation measures and LMI-relaxations. *SIAM journal on control and optimization* 47.4 (2008), pp. 1643–1666.

[190] M. Laurent and L. Slot. Near-optimal analysis of Lasserre's univariate measure-based bounds for multivariate polynomial optimization. *Math. Program.* (2020). preprint arXiv:2001.11289.

[191] M. Laurent. Matrix Completion Problems. *Encyclopedia of Optimization* 3 (2009), pp. 221–229.

[192] J. Lawrence. Polytope volume computation. *mathematics of computation* 57.195 (1991), pp. 259–271.

[193] P. D. Lax. Differential equations, difference equations and matrix theory. *Comm. Pure Appl. Math.* 11 (1958), pp. 175–194.

[194] T. Lelièvre, M. Rousset, and G. Stoltz. Hybrid Monte Carlo methods for sampling probability measures on submanifolds. *Numerische Mathematik* 143.2 (2019), pp. 379–421.

[195] F. Lemaire, M. M. Maza, and Y. Xie. The RegularChains Library in MAPLE. *SIGSAM Bull.* 39.3 (Sept. 2005), pp. 96–97.

[196] E. H. Lieb and M. Loss. "Analysis, Graduate Stud. Math., vol. 14". *Amer. Math. Soc.* 2001.

[197] E. H. Lieb and R. Seiringer. Equivalent forms of the Bessis-Moussa-Villani conjecture. *J. Statist. Phys.* 115.1-2 (2004), pp. 185–190.

[198] M. Liu et al. $H_\infty$ State Estimation for Discrete-Time Chaotic Systems Based on a Unified Model. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 42.4 (2012), pp. 1053–1063.

[199] J. Lofberg. Pre-and post-processing sum-of-squares programs in practice. *IEEE transactions on automatic control* 54.5 (2009), pp. 1007–1011.

[200] J. Löfberg. "YALMIP : A Toolbox for Modeling and Optimization in MATLAB". *Proceedings of the CACSD Conference*. Taipei, Taiwan, 2004.

[201] D. G. Luenberger. *Optimization by Vector Space Methods*. 1st. New York, NY, USA: John Wiley & Sons, Inc., 1997.

[202] M. Dressler, S. Iliman, and T. de Wolff. A Positivstellensatz for Sums of Nonnegative Circuit Polynomials. *SIAM J. Appl. Algebra Geom.* 1.1 (2017), pp. 536–555.

[203] M. Dressler, S. Iliman, and T. de Wolff. *An Approach to Constrained Polynomial Optimization via Nonnegative Circuit Polynomials and Geometric Programming*. To appear in the Journal of Symbolic Computation (MEGA 2017 special issue); see also arXiv:1602.06180. 2016.

[204] M. Ghasemi and M. Marshall. *Lower Bounds for a Polynomial on a basic closed semialgebraic set using geometric programming*. Preprint, arxiv:1311.3726. 2013.

[205] M. Ghasemi and M. Marshall. Lower bounds for polynomials using geometric programming. *SIAM J. Optim.* 22.2 (2012), pp. 460–473.

[206] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Second corrected edition. Vol. 2. Algorithms and Combinatorics. Springer, 1993.

[207] M. H. Stone. The Generalized Weierstrass Approximation Theorem. *Mathematics Magazine* 21.4 (1948), pp. 167–184.

[208] M. Trnovská. Strong duality conditions in semidefinite programming. *Journal of Electrical Engineering* 56.12/s (2005), pp. 1–5.

[209] A. Majumdar, A. A. Ahmadi, and R. Tedrake. "Control and verification of high-dimensional systems with DSOS and SDSOS programming". *53rd IEEE Conference on Decision and Control*. IEEE. 2014, pp. 394–401.

[210] M. Marshall. *Positive polynomials and sums of squares*. Vol. 146. Mathematical Surveys and Monographs. American Mathematical Society, Providence, RI, 2008, pp. xii+187.

[211] S. Marx et al. A moment approach for entropy solutions to nonlinear hyperbolic PDEs. *Mathematical Control and Related Fields* 10.2156-8472_2020_1_113 (2020), p. 113.

[212] S. McCullough. Factorization of operator-valued polynomials in several non-commuting variables. *Linear Algebra Appl.* 326.1-3 (2001), pp. 193–203.

[213] S. McCullough and M. Putinar. Noncommutative sums of squares. *Pacific J. Math.* 218.1 (2005), pp. 167–171.

[214] A. Megretski. Systems polynomial optimization tools (SPOT). *Massachusetts Inst. Technol., Cambridge, MA, USA* (2010).

[215] A. Megretski and A. Rantzer. System Analysis via integral Quadratic Constraints. *IEEE Transactions on Automatic Control* 42.6 (1997), pp. 819–830.

[216] M. Mevissen, E. Ragnoli, and J. Y. Yu. "Data-driven distributionally robust polynomial optimization". *Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 1*. 2013, pp. 37–45.

[217] M. Mignotte. *Mathematics for Computer Algebra*. New York, NY, USA: Springer-Verlag New York, Inc., 1992.

[218] C. Mira et al. *Chaotic dynamics in two-dimensional noninvertible maps*. World Scientific Series on Nonlinear Science Series A. Singapore: World Scientific, 1996.

[219] K. Murota et al. A numerical algorithm for block-diagonal decomposition of matrix $*$-algebras with application to semidefinite programming. *Jpn. J. Ind. Appl. Math.* 27.1 (2010), pp. 125–160.

[220] R. Murray, V. Chandrasekaran, and A. Wierman. Newton Polytopes and Relative Entropy Optimization. *arXiv preprint arXiv:1810.01614* (2018).

[221] M. Nakata. "A numerical evaluation of highly accurate multiple-precision arithmetic version of semidefinite programming solver: SDPA-GMP, -QD and -DD." *CACSD*. 2010, pp. 29–34.

[222] S. G. Nash. Newton-type minimization via the Lánczos method. *SIAM J. Numer. Anal.* 21.4 (1984), pp. 770–788.

[223] M. Navascués et al. A paradox in bosonic energy computations via semidefinite programming relaxations. *New Journal of Physics* 15.2 (2013), p. 023026.

[224] M. Navascués, S. Pironio, and A. Acín. A convergent hierarchy of semidefinite programs characterizing the set of quantum correlations. *New J. Phys.* 10.7 (2008), p. 073013.

[225] T. Netzer and A. Thom. Hyperbolic polynomials and generalized Clifford algebras. *Discrete Comput. Geom.* 51.4 (2014), pp. 802–814.

[226] J. Nie and M. Schweighofer. On the complexity of Putinar's Positivstellensatz. *Journal of Complexity* 23.1 (2007), pp. 135–150.

[227] J. Nie. The $\mathcal{A}$-truncated $K$-moment problem. *Found. Comput. Math.* 14.6 (2014), pp. 1243–1276.

[228] M. P. Nightingale and C. J. Umrigar. *Quantum Monte Carlo methods in physics and chemistry*. 525. Springer Science & Business Media, 1998.

[229] O. Nikodym. Sur une généralisation des intégrales de M. J. Radon. fre. *Fundamenta Mathematicae* 15.1 (1930), pp. 131–179.

[230] M. C. de Oliveira et al. "Engineering systems and free semi-algebraic geometry". *Emerging applications of algebraic geometry*. Vol. 149. IMA Vol. Math. Appl. Springer, New York, 2009, pp. 17–61.

[231] A. Oustry, M. Tacchi, and D. Henrion. Inner approximations of the maximal positively invariant set for polynomial dynamical systems. *IEEE Control Systems Letters* 3.3 (2019), pp. 733–738.

[232] K. F. Pál and T. Vértesi. Quantum bounds on Bell inequalities. *Phys. Rev. A (3)* 79.2 (2009), pp. 022120, 12.

[233] G. Pataki. Bad Semidefinite Programs: They All Look the Same. *SIAM Journal on Optimization* 27.1 (2017), pp. 146–172.

[234] V. Paulsen. *Completely bounded maps and operator algebras.* Vol. 78. Cambridge studies in advanced mathematics. Cambridge University Press, 2002, pp. xii+300.

[235] E. Pauwels, M. Putinar, and J.-B. Lasserre. Data analysis from empirical moments and the Christoffel function. *Foundations of Computational Mathematics* (2020), pp. 1–31.

[236] H. Peyrl and P. Parrilo. Computing sum of squares decompositions with rational coefficients. *Theor. Comput. Sci.* 409.2 (2008), pp. 269–281.

[237] H. Peyrl and P. Parrilo. Computing sum of squares decompositions with rational coefficients. *Theoretical Computer Science* 409.2 (2008), pp. 269–281.

[238] S. Pironio, M. Navascués, and A. Acín. Convergent relaxations of polynomial optimization problems with noncommuting variables. *SIAM J. Optim.* 20.5 (2010), pp. 2157–2180.

[239] G. Pólya and G. Szegő. *Problems and theorems in analysis. II.* Classics in Mathematics. Theory of functions, zeros, polynomials, determinants, number theory, geometry, Translated from the German by C. E. Billigheimer, Reprint of the 1976 English translation. Springer-Verlag, Berlin, 1998, pp. xii+392.

[240] Y. Pourchet. Sur la représentation en somme de carrés des polynômes à une indéterminée sur un corps de nombres algébriques. fre. *Acta Arithmetica* 19.1 (1971), pp. 89–104.

[241] A. Pozas-Kerstjens et al. Bounding the sets of classical and quantum correlations in networks. *Phys. Rev. Lett.* 123.14 (2019), pp. 140503, 6.

[242] S. Prajna and A. Jadbabaie. "Safety verification of hybrid systems using barrier certificates". *International Workshop on Hybrid Systems: Computation and Control*. Springer. 2004, pp. 477–492.

[243] A. Prestel and C. Delzell. *Positive Polynomials: From Hilbert's 17th Problem to Real Algebra*. Springer Monographs in Mathematics. Springer Berlin Heidelberg, 2001.

[244] C. Procesi. The invariant theory of $n \times n$ matrices. *Adv. Math.* 19.3 (1976), pp. 306–381.

[245] M. Putinar. Positive polynomials on compact semi-algebraic sets. *Indiana University Mathematics Journal* 42.3 (1993), pp. 969–984.

[246] R. Quarez. Tight bounds for rational sums of squares over totally real fields. *Rendiconti del Circolo Matematico di Palermo* 59.3 (2010), pp. 377–388.

[247] R.J. Duffin, E.L. Peterson, and C. Zener. *Geometric programming: Theory and application*. John Wiley & Sons, Inc., New York-London-Sydney, 1967.

[248] A. Rantzer and P. A. Parrilo. "On convexity in stabilization of nonlinear systems". *Proceedings of the 39th IEEE Conference on Decision and Control*. Vol. 3. 2000, 2942–2945 vol.3.

[249] J. Renegar. On the computational complexity and geometry of the first order theory of the reals. *Journal of Symbolic Computation* 13.3 (1992), pp. 255–352.

[250] A. Ricou. "Necessary conditions for nonnegativity on *-algebras and ground state problem". MA thesis. National University of Singapore, 2019.

[251] C. Riener et al. Exploiting Symmetries in SDP-Relaxations for Polynomial Optimization. *Math. Oper. Res.* 38.1 (Feb. 2013), pp. 122–141.

[252] H. Royden and P. Fitzpatrick. *Real Analysis*. Featured Titles for Real Analysis Series. Prentice Hall, 2010.

[253] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge: Cambridge University Press, 2004, pp. xiv+716.

[254] S. Boyd et al. A tutorial on geometric programming. *Optim. Eng.* 8.1 (2007), pp. 67–127.

[255] S. Iliman and T. de Wolff. Lower Bounds for Polynomials with Simplex Newton Polytopes Based on Geometric Programming. *SIAM J. Optim.* 26.2 (2016), pp. 1128–1146.

[256] S. Iliman and T. d. Wolff. Amoebas, nonnegative polynomials and sums of squares supported on circuits. *Res. Math. Sci.* 3 (2016), 3:9.

[257] M. Safey El Din. Testing sign conditions on a multivariate polynomial and applications. *Mathematics in Computer Science* 1.1 (2007), pp. 177–207.

[258] M. Safey El Din and É. Schost. "Polar varieties and computation of one point in each connected component of a smooth real algebraic set". *ISSAC'03*. ACM, 2003, pp. 224–231.

[259] M. Safey El Din and L. Zhi. Computing Rational Points in Convex Semialgebraic Sets and Sum of Squares Decompositions. *SIAM J. on Optimization* 20.6 (Sept. 2010), pp. 2876–2889.

[260] M. Safey El Din and L. Zhi. Computing rational points in convex semialgebraic sets and sum of squares decompositions. *SIAM Journal on Optimization* 20.6 (2010), pp. 2876–2889.

[261] M. Safey El Din. *RAGlib – A library for real solving polynomial systems of equations and inequalities*. `http://www-polsys.lip6.fr/~safey/RAGLib/distrib.html`. 2007.

[262] C. Schlosser and M. Korda. Sparse moment-sum-of-squares relaxations for nonlinear dynamical systems with guaranteed convergence. *arXiv preprint arXiv:2012.05572* (2020).

[263] M. Schweighofer. "Algorithmische Beweise für Nichtnegativ- und Positivstellensätze". MA thesis. Diplomarbeit an der Universität Passau, 1999.

[264] H. Seidler and T. de Wolff. An Experimental Comparison of SONC and SOS Certificates for Unconstrained Optimization. *arXiv preprint arXiv:1808.08431* (2018).

[265] H. Seidler and T. de Wolff. *POEM: Effective Methods in Polynomial Optimization, version 0.1.1.0(a)*. `https://www3.math.tu-berlin.de/combi/RAAGConOpt/poem.html`. 2018.

[266] E. Sertöz. Computing periods of hypersurfaces. *Mathematics of Computation* 88.320 (2019), pp. 2987–3022.

[267] V. Shia et al. "Convex computation of the reachable set for controlled polynomial hybrid systems". *53rd IEEE Conference on Decision and Control*. IEEE. 2014, pp. 1499–1506.

[268] P. Shukl and B. Singh. Combined IIR and FIR filter for improved power quality of PV interfaced utility grid. *IEEE Transactions on Industry Applications* 57.1 (2020), pp. 774–783.

[269] M. Sion. On general minimax theorems. *Pacific J. Math.* 8.1 (1958), pp. 171–176.

[270] R. E. Skelton, T. Iwasaki, and K. M. Grigoriadis. *A unified algebraic approach to linear control design*. The Taylor & Francis Systems and Control Book Series. Taylor & Francis, Ltd., London, 1998, pp. xviii+285.

[271] L. Slot and M. Laurent. Improved convergence analysis of Lasserre's measure-based upper bounds for polynomial minimization on compact sets. *Mathematical Programming* (2020), pp. 1–41.

[272] A. Solovyev and T. C. Hales. "Formal Verification of Nonlinear Inequalities with Taylor Interval Approximations". *NASA Formal Methods, 5th International Symposium, NFM 2013, Moffett Field, CA, USA, May 14-16, 2013. Proceedings*. 2013, pp. 383–397.

[273] A. Solovyev et al. "Rigorous Estimation of Floating-Point Round-off Errors with Symbolic Taylor Expansions". *Proceedings of the 20th International Symposium on Formal Methods (FM)*. Ed. by N. Bjorner and F. de Boer. Vol. 9109. Lecture Notes in Computer Science. Springer, 2015, pp. 532–550.

[274] E. D. Sontag. *Mathematical control theory: deterministic finite dimensional systems*. Vol. 6. Springer Science & Business Media, 2013.

[275] F. Sottile. Real Schubert Calculus: Polynomial Systems and a Conjecture of Shapiro and Shapiro. *Experimental Mathematics* 9.2 (2000), pp. 161–182.

[276] M. K. Srivas and A. J. Camilleri, eds. *Formal Methods in Computer-Aided Design, First International Conference, FMCAD '96, Palo Alto, California, USA, November 6-8, 1996, Proceedings*. Vol. 1166. Lecture Notes in Computer Science. Springer, 1996.

[277] H. R. Stahl. Proof of the BMV conjecture. *Acta Math.* 211.2 (2013), pp. 255–290.

[278] A. Strzebonski and E. Tsigaridas. "Univariate Real Root Isolation in an Extension Field". *Proceedings of the 36th International Symposium on Symbolic and Algebraic Computation*. ISSAC '11. San Jose, California, USA: ACM, 2011, pp. 321–328.

[279] J. F. Sturm. *Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones*. 1998.

[280] D. Sutter, M. Berta, and M. Tomamichel. Multivariate trace inequalities. *Comm. Math. Phys.* 352.1 (2017), pp. 37–58.

[281] T. Banks and T. Constantinescu. Orthogonal polynomials in several non-commuting variables. II. *arXiv:math/0412528v1 [math.FA]* (2004).

[282] T. de Wolff. Amoebas, Nonnegative Polynomials and Sums of Squares Supported on Circuits. *Oberwolfach Rep.* 23 (2015). Ed. by H. Markwig, G. Mikhalkin, and E. Shustin, pp. 1308–1311.

[283] M. Tacchi et al. Exploiting Sparsity for Semi-Algebraic Set Volume Computation. *preprint arXiv:1902.02976* (2019).

[284] M. Takesaki. *Theory of operator algebras. I.* Vol. 124. Encyclopaedia of Mathematical Sciences. Reprint of the first (1979) edition, Operator Algebras and Non-commutative Geometry, 5. Springer-Verlag, Berlin, 2002, pp. xx+415.

[285] E. Y.-Z. Tan et al. Computing secure key rates for quantum key distribution with untrusted devices. *arXiv preprint arXiv:1908.11372* (2019).

[286] S. Tarbouriech et al. *Stability and Stabilization of Linear Systems with Saturating Actuators*. Springer, 2011.

[287] M. E. Taylor. *Partial Differential Equations: Basic Theory*. New York: Springer-Verlag, New Yor, Inc., Springer Texts in Mathematics, 1996.

[288] T. C. D. Team. *The Coq Proof Assistant, version 8.12.0*. 2020.

[289] *The MOSEK optimization software*. http://www.mosek.com/.

[290] V. Chandrasekaran and P. Shah. Relative Entropy Relaxations for Signomial Optimization. *SIAM J. Optim.* 26.2 (2016), pp. 1147–1173.

[291] B. Van der Pol. On relaxation oscillations. *The London, Edinburgh and Dublin Phil. Mag. & J. of Sci.* 2 (1926), pp. 978–992.

[292] M. Van Kreveld et al. *Computational geometry algorithms and applications*. Springer, 2000.

[293] L. Vandenberghe, M. S. Andersen, et al. Chordal graphs and semidefinite optimization. *Foundations and Trends® in Optimization* 1.4 (2015), pp. 241–433.

[294] A. Vannelli and M. Vidyasagar. Maximal Lyapunov functions and domains of attraction for autonomous nonlinear systems. *Automatica* 21.1 (1985), pp. 69–80.

[295] D.-V. Voiculescu. "Symmetries of some reduced free product $C^*$-algebras". *Operator algebras and their connections with topology and ergodic theory (Busteni, 1983)*. Vol. 1132. Lecture Notes in Math. Springer, Berlin, 1985, pp. 556–588.

[296] A. Volkova, C. Lauter, and T. Hilaire. "Reliable verification of digital implemented filters against frequency specifications". *24th IEEE Symposium on Computer Arithmetic*. IEEE, 2017.

[297] H. Waki. How to generate weakly infeasible semidefinite programs via Lasserre's relaxations for polynomial optimization. *Optimization Letters* 6.8 (2012), pp. 1883–1896.

[298] H. Waki, M. Nakata, and M. Muramatsu. Strange behaviors of interior-point methods for solving semidefinite programming problems in polynomial optimization. *Computational Optimization and Applications* 53.3 (2012), pp. 823–844.

[299] H. Waki et al. Algorithm 883: sparsePOP—a sparse semidefinite programming relaxation of polynomial optimization problems. *ACM Trans. Math. Software* 35.2 (2009), Art. 15, 13.

[300] H. Waki et al. Sums of Squares and Semidefinite Programming Relaxations for Polynomial Optimization Problems with Structured Sparsity. *SIAM Journal on Optimization* 17.1 (2006), pp. 218–242.

[301] J. Wang, H. Li, and B. Xia. "A new sparse SOS decomposition algorithm based on term sparsity". *Proceedings of the 2019 on International Symposium on Symbolic and Algebraic Computation*. 2019, pp. 347–354.

[302] T. Weisser, J.-B. Lasserre, and K.-C. Toh. Sparse-BSOS: a bounded degree SOS hierarchy for large scale polynomial optimization with sparsity. *Mathematical Programming Computation* 10.1 (2018), pp. 1–32.

[303] H. Weyl. Über die Gibbs' sche Erscheinung und verwandte Konvergenzphänomene. *Rendiconti del Circolo Matematico di Palermo (1884-1940)* 30.1 (1910), pp. 377–407.

[304] S. R. White. Density matrix formulation for quantum renormalization groups. *Physical review letters* 69.19 (1992), p. 2863.

[305] H. Whitney. *Geometric Integration Theory*. Princeton, NJ: Princeton University Press, 1957.

[306] P. Wittek. Algorithm 950: Ncpol2sdpa-sparse semidefinite programming relaxations for polynomial optimization problems of noncommuting variables. *ACM Trans. Math. Software* 41.3 (2015), Art. 21, 12.

[307] H. R. Wütrich. "Ein Entschiedungsverfahren für die Theorie der reell-abgeschlossenen Körper". *Lecture Notes in Computer Science*. Vol. 43. 1976, pp. 138–162.

[308] Y. Xu. Cubature formulae and polynomial ideals. *Advances in Applied Mathematics* 23.3 (1999), pp. 211–233.

[309] V. A. Yakubovich. S-procedure in nonlinear control theory. *Vestnik Leningrad University: Mathematics* 4 (1977), pp. 73–93.

[310] M. Yamashita et al. *A high-performance software package for semidefinite programs : SDPA7*. Tech. rep. Tokyo Inst. Tech.: Dept. of Information Sciences, 2010.

[311] D. Yun and B. Protas. Maximum rate of growth of enstrophy in solutions of the fractional Burgers equation. *Journal of Nonlinear Science* 28.1 (2018), pp. 395–422.

[312] G. Zames and P. Falb. Stability Conditions for Systems with Monotone and Slope-Restricted Nonlinearities. *SIAM Journal on Control* 6.1 (1968), pp. 89–108.

[313] E. Zappa, M. Holmes-Cerfon, and J. Goodman. Monte Carlo on manifolds: sampling densities and integrating functions. *Communications on Pure and Applied Mathematics* 71.12 (2018), pp. 2609–2647.

# Curriculum Vitae

Victor Magron

Born the 27/12/1985 in Toulouse, France Nationality: French

Address: CNRS LAAS, 7 avenue du Colonel Roche, Bureau E48, F-31031 Toulouse, France

Phone: +33 (0)5 61 33 64 81

Email: victor.magron@laas.fr Website: https://homepages.laas.fr/vmagron/

**Doctoral School**: ED 475, Mathématiques, Informatique, Télécommunications de Toulouse (Mathematics, Computer Science and Telecomunications, Toulouse)

## I - Education

2010-2013 **Ph. D. in computer science**, Ecole Polytechnique. *Formal Proofs for Global Optimization – Templates and Sums of Squares*, with highest honor. Under the supervision of B. Werner (Computer science Dept., École Polytechnique) and S. Gaubert (Applied Math. Dept., École Polytechnique) at INRIA Saclay in Palaiseau.

2008-2010 **Ms. Thesis in computer science** (rank A), Tokyo University, Dept. of Systems Innovation. *Time Dependent Magnetic Structural Coupled Analysis of MRI Model with Hierarchical Domain Decomposition Methods*, obtained with rank A, under the supervision of Shinobu Yoshimura.

2006-2010 École Centrale de Paris. Double diploma (Ms. Eng. degree) with Tokyo University. Training in Mathematics, Mechanics, Computer Science, Economics and Automatic Control.

2003 French HS diploma, Science major, with highest honors.

### Training Schools

2016 Three days Course on *Diffusion of Scientific Knowledge*, CNRS, Paris, France.

2015 Summer school on *Automatic Control, Invariant sets*, CNRS GIPSA-lab, Grenoble, France.

2013 Summer school on *Polynomial Optimization*, Isaac Netwton Institute, Cambridge, UK.

2012 Summer school on *Semidefinite Optimization*, Kirchberg/Hunsrück, Haus Karrenberg, Germany.

2012 Spring school on *Formalization of Mathematics*, INRIA Sophia-Antipolis, France.

## II - Positions and fellowships

2019-now CNRS (national center for scientific research): chargé de recherche de classe normale (junior researcher) at LAAS, MAC team, Toulouse.

2020-now Associate member of the Institute of Mathematics in Toulouse (IMT).

### Previous positions

2018 Associate member at CNRS LIP6, PolSys Team, Paris.

2015-2018 CNRS: chargé de recherche de seconde classe , VERIMAG, Grenoble, France (laboratory hosted by Université Grenoble Alpes).

2014-2015 Imperial College, Dept. of Electrical & Electronic Eng., research assistant in the Circuits and Systems group, with G. Constantinides and A. Donaldson.

2014 CNRS LAAS, postdoc in the Methods & Algorithms for Control group, with J.-B. Lasserre and D. Henrion.

### Teaching

2019 Mini-course on Semidefinite Programming and the Moment/Sums-of-squares Hierarchy, TU Cheminitz Fakultät für Elektrotechnik und Informationstechnik, Systems and Control seminar (6 hr of classes).

2018 ENS Rennes SPA Solvers, Principles and Architectures (3 hr of classes).

2016 Ensimag, Grenoble, teaching assistant in a software engineering course (compiler project) for Master students (60 hr of classes).

2010–2013 École Polytechnique: teaching assistant in the Department of Computer science for Undergraduate and Master students. Fundamentals of Programming and Algorithms, Principles of Programming Language, Algorithms and Programming. (64 hr of classes per year).

### Fellowships and awards

2018 *RealCertify: a Maple package for certifying non-negativity*. In ISSAC, **Best Software Demo Award**.

2017 *Certified Roundoff Error Bounds using Bernstein Expansions and Sparse Krivine-Stengle Representations*. In 24th IEEE Symposium on Computer Arithmetic, **Best Paper Award**.

2014-2015 Postdoc position funded by EPSRC grant EP/I020457/1 (Challenging Engineering Project).

2014 Postdoc position funded by fellowship of Simone and Cino del Duca foundation of the France Institute.

2010-2013 Ph. D funded by INRIA Grant (Formath EU Project).

2008-2010 Ms. funded by Monbukagakusho (Japanese Government Fellowship).

## III - Supervision

### Postdocoral fellows

2021 Abhishek Bhardwaj, Positivity certificates for noncommutative polynomials, funded by the Tremplin-COPS project.

2019-2021 Jie Wang, Certificates for sparse nonnegative polynomials, 2019-2021, funded by the Tremplin-COPS project.

2015-2017 Involvement in the (non-official) supervision of the postdoc of M. Forets in CNRS VERIMAG.

### PhD students

2019-2022 Hoang Ngoc Anh Mai, Extraction and certification of solutions from moments: generalized Christoffel functions for semialgebraic optimization, with J.-B. Lasserre.

2019-2022 Tong Chen, Polynomial optimization for certification of deep networks, with J.-B. Lasserre and E. Pauwels.

2019-2022 Vu Trung Hieu, Algebraic tools for exact SDP and its variants, with M. Safey El Din.

2015-2018 Involvement in the (non-official) supervision of the Ph. D thesis of A. Rocca in CNRS VERIMAG, with T. Dang and E. Fanchon.

### Graduate students

2019 Two Master 2 students from University of Limoges, respectively co-supervised with J.-B. Lasserre (LAAS) and K. Ghorbal (INRIA Rennes). A Master 2 student from University of Versailles, co-supervised at LIP6 with M. Safey El Din (INRIA/LIP6 PolSys). A Master 2 student from University Paris-Sud, co-supervised at IRT with E. Pauwels (IRIT) and J.-B. Lasserre (LAAS).

2016 Two Master 1 students from Ensimag Grenoble France, supervised at CNRS VERIMAG, Grenoble.

2015 One Master 2 student from Konstanz University, supervised at É. Polytechnique with B. Werner (LIX), Computer science Dept.

## IV - Organization, collaborations, research projects, evaluation, coordination

### Organization of scientific meetings

2020-2021 Co-organizer with I. Klep of the mini-symposium *Computational aspects of commutative and noncommutative positive polynomials* at the European Congress of Mathematicians

2020-2021 Co-organizer with M. Laurent and B. Mourrain of the mini-symposium *Positive Polynomials, Moments, and Applications* at the SIAM conference on Applied Algebraic Geometry

2020-2021 Co-organizer with M. Korda of the mini-symposium *Polynomial optimization: algorithms and applications* at the SIAM conference on Optimization

2021 Co-organizer with M. Korda and D. Henrion of the mini-symposium *Optimization over measures and positive polynomials* at the SMAI Congress

2020-now Organizer of the BrainPOP group: workshops and discussions about polynomial optimization.

2020-now Member of the local organizing committee scientific of SPOT: Multidisciplinary optimization seminar in Toulouse.

2017-now Local organization committee of the SMAI-MODE optimization conference.

2016-2017 Co-organizer of the Reading group in Optimization and Control of Univ. Grenoble Alpes focusing on the interplay between convex optimization and optimal control.

2015-2019 Organizer of five sessions at optimization conference (BFG'15, FGI'17, PGMO'17-18', SMAI), dedicated to the application of polynomial optimization to the fields of static analysis, control and computer arithmetic.

## Scientific vulgarization

2014 Techniques de preuve formelle en science : le défi, Semaines Sociales de France. Session: L'Homme et les Technosciences, le défi. Université Catholique de Lille. Text

## Commissions of trust

2020-now Member of the scientific committee of DO: the decision and optimization department at LAAS CNRS.

2017-now Member of the committee of the french scientific society on the mathematics of optimization and decision SMAI-MODE.

2015-2017 Reviewer for Mathematical Reviews (AMS).

2012-2017 I have reviewed around hundred articles for the following international peer-reviewed international conferences and journals.
Optimization and control: CoDIT, ECC, HSCC, CAMSAP, IEEE Trans. on Automatic Control, Circuits Systems & Signal Proc., SIAM Rev., Journal of the Franklin Inst., SIAM J. on Optim, Optim. Letters, Math. Prog., Math. Comp.
Computer science: MEGA, SYNASC, POPL, LATA, TACAS, CAV, CADE, ISSAC, Journal of Formalized Reasoning, Optimization Methods & Software, Formal Methods in System Design.

## Project grants

2021-2022 PEPS2 research collaboration between LAAS and RTE **FastOPF** (FAST polynomial optimization techniques for Optimal Power Flow). Funded by the French Agency for mathematics in interaction with industry and society (AMIES). Project leader: Victor Magron. Other participants from LAAS: Jean-Bernard Lasserre, Hoang Ngoc Anh Mai, Jie Wang. Participants from RTE: Jean Maeght, Patrick Panciatici and Manuel Ruiz.

2021-2022 Bilateral research collaboration between France and Slovenia **QUANTPOP** (QUANTum information with noncommutative Polynomial OPtimization). Funded by Partenariat Hubert Curien (PHC) Proteus. Project leader: Victor Magron. Other participants from France: Ion Nechita (CNRS IRSAMC). Participants from Slovenia: Igor Klep and Janez Povh (University of Ljubljana).

2021 International mobility grant at LTH, Sweden **POPSIC** (Polynomial OPtimization for Scalability In Control). Funded by the International Centre for Mathematics and Computer Science in Toulouse (CIMI). Project leader: Victor Magron. Participants from LTH: Anders Rantzer, Martina Maggio, Richard Pates, Pauline Kergus.

2019-2023 Polynomial optimization for Machine Learning **ANITI** (Artificial and Natural Intelligence Toulouse Institute). Topic: Optimization for Machine Learning and the Christoffel function for data analysis. Chair: Jean-Bernard Lasserre (LAAS CNRS).

2019-2022 European Commission Marie Sklodowska-Curie Innovative Training Network **POEMA** (Polynomial Optimization, Efficiency through Moments and Algebra). Période: 2019-2022. Project leader: Bernard Mourrain (INRIA Sophia, Nice). Partners: Mohab Safey El Din (Sorbonne Univ, Paris), Monique Laurent (CWI-NWO Amsterdam), Etienne de Klerk (Tilburg Univ), Markus Schweighofer (Univ Konstanz), Giorgio Ottaviani (Univ. Florence), Michael Stingl (Univ. Erlangen Nurnberg), Cordian Riener (Univ. Tromsoe), Arnaud Renaud (Artelys France), Martin Mevissen (IBM Research Ireland), Mike Dewar (Numerical Algorithms Group), Jean Maeght (RTE France). I will co-supervise a Ph. D candidate at Sorbonne Université.

2019-2020 Tremplin-ERC Starting Grant **Tremplin-COPS**. Funded by ANR. Project leader: Victor Magron. Topic: "Certification and Modeling of Polynomial Optimization Problems".

2018-2021 PGMO (Projet Gaspard Monge for Optimization) **EPICS**. Funded by fondation Jacques Hadamard. Project leader: Victor Magron. Other participants: T. Dang (CNRS VERIMAG, Grenoble) and J.-C. Faugère (INRIA Polsys, Paris). Topic: "Efficient Exact Polynomial optimization with Innovative Certifed Schemes".

2016-2017 Exploratory Project Persyval-Lab **AEPS**. Funded by the French program Investissement d'avenir (ANR-11-LABX-0025-01). Project leader: Victor Magron. Participants: Victor Magron (CNRS Verimag, Grenoble), Bruno Gaujal (INRIA Mescal/CNRS Lig, Grenoble) and Panayotis Mertikopoulos (CNRS Lig, Grenoble). Topic: Algorithmes efficaces de programmation semidéfinie pour l'optimisation stochastique.

2016 PEPS-JCJC (Pluridisciplinary Exploratory Project Young researchers) **ACE**. Funded by the French CNRS Institute of Information Sciences (INS2I). Project leader: Delphine Bresch-Pietri (CNRS Gipsa-Lab, Grenoble). Topic: Analysis and Control of Partial Differential Equations.

# V - Research talks and software

## Selection of invited talks and seminars

- Invited talk on *Optimization over trace polynomials*, 2021 February 3, Journal Club Information Quantique, Toulouse.
- Invited talk on *Sparse (Non)commutative Polynomial Optimization*, 2020 March 6, Real Algebraic Geometry with a View Toward Hyperbolic Programming and Free Probability, Oberwolfach.
- Invited talk on *Polynomial Optimization for Bounding Lipschitz Constants of Deep Networks*, 2020 February 28, Intersections between Control, Learning and Optimization IPAM, Los Angeles.
- Seminar on *Certified and efficient polynomial optimization via conic programming*, 2020 January 28, Department of Automatic Control LTH, Lund.
- Seminar on *Two-player games between polynomial optimizers and semidefinite solvers*, 2020 January 21, Mosek Aps Science Park, Copenhagen.
- Invited talk on *The quest of efficiency and certification in polynomial optimization*, 2019 November 4, Séminaire Pluridisciplinaire d'Optimisation de Toulouse (SPOT) Enseeiht, Toulouse.
- Invited talk on *Exact polynomial optimization via SOS, SONC and SAGE decompositions*, 2019 June 25, EURO, UCD Dublin.

- *Two-player games between polynomial optimizers and semidefinite solvers*, 2019 July 11, SIAM Conference on Applied Algebraic Geometry Bern.
- Invited talk on *Certified Semidefinite Approximations of Reachable Sets*, 2019 June 19, FEANICSES Workshop, ISAE, Toulouse
- *RealCertify: a Maple package for certifying non-negativity*, 2018 July 17, ISSAC 2018 (Software Demonstration) New-York.
- *On Exact Polya and Putinar's Representations*, 2018 July 17, ISSAC 2018 New-York.
- *Enclosures of Roundoff Errors using SDP*, 2017 September 27, 18th French-German-Italian Conference on Optimization, Paderborn.
- *Semidefinite Approximations of Reachability Sets for Discrete-time Polynomial Systems*, 2017 August 3, SIAM Conference on Applied Algebraic Geometry, Atlanta, Georgia Tech.
- *Convergent Robust SDP Approximations for Semialgebraic Optimization*, 2016 August 10, ICCOPT 2016 GRIPS, Tokyo.
- *Automated Precision Tuning using Semidefinite Programming*, 2015 June 15, 17th British-French-German Conference on Optimization, Imperial College, London.
- *Semidefinite approximations of projections and polynomial images of semialgebraic sets*, 2014 November 5, JNCF 2014 CIRM, Marseille.
- *Formal Nonlinear Optimization via Templates and Sum-of-Squares*, 2013 April 23, TYPES 2013 Toulouse, France.
- Invited talk on *Certification of Inequalities involving Transcendental Functions using Semidefinite Programming*, 2012 August 24, ISMP 2012 Berlin.

## Software developments

- **ctpPOP**: A Julia library to exploiting constant trace property in large-scale polynomial optimization.
- **TSSOS**: A Julia library for sparse polynomial optimization tool based on block moment-SOS hierarchies. See also the noncommutative module **NCTSSOS**.
- **SparseJSR**: A julia library to compute joint spetral radii based on sparse SOS decompositions.
- **SONCSOCP**: A Julia library unconstrained sparse polynomial optimization tool based on SONC decompositions.
- **RealCertify**: a Maple package for certifying non-negativity.
- **NLCertify**: a tool for formal nonlinear optimization.
- **Real2Float**: a tool for certified roundoff error bounds using SDP, built in top of NLCertify.
- **FPBern**: a tool for certified roundoff error bounds using Bernstein expansions.
- **FPKriSten**: a tool for certified roundoff error bounds using sparse Krivine-Stengle representations.

## The quest of modeling, certification and efficiency in polynomial optimization

**Abstract:** Certified optimization techniques have successfully tackled challenging verification problems in various fundamental and industrial applications. The formal verification of thousands of nonlinear inequalities arising in the famous proof of Kepler conjecture was achieved in August 2014. In energy networks, it is now possible to compute the solution of large-scale power flow problems with up to thousand variables. This success follows from growing research efforts in polynomial optimization, an emerging field extensively developed in the last two decades. One key advantage of these techniques is the ability to model a wide range of problems using optimization formulations, which can be in turn solved with efficient numerical tools. My methodology heavily relies on such methods, including the moment-sums of squares (moment-SOS) hierarchy by Lasserre which provides numerical certificates for positive polynomials as well as recently developed alternative methods. However, such optimization methods still encompass many major issues on both practical and theoretical sides: scalability, unknown complexity bounds, ill-conditioning of numerical solvers, lack of exact certification, convergence guarantees. This manuscript presents results along these research tracks with the long-term perspective of obtaining scientific breakthroughs to handle certification of nonlinear systems arising in real-world applications.

In the first part, I focus on modeling aspects. One relies on the moment-SOS hierarchy to analyze dynamical polynomial systems, either in the discrete-time or continuous-time setting, and problems involving noncommuting variables, for example matrices of finite or infinite size, to model quantum physics operators. In the second part, I describe how to design and analyze algorithms which output exact positivity certificates for either unconstrained or constrained optimization problems. In the last part, I explain how to improve the scalability of the hierarchy by exploiting the specific sparsity structure of the polynomial data coming from real-world problems. Important applications arise from various fields, including computer arithmetic (round-off error bounds), quantum information (noncommutative optimization), and optimal power-flow.

**Keywords:** Polynomial optimization, moments, sums of squares, eigenvalue & trace optimization, hybrid numeric-symbolic algorithms, certified optimization, sparsity pattern