ALGORITHMIC ASPECTS OF SUMS OF HERMITIAN SQUARES OF NONCOMMUTATIVE POLYNOMIALS

SABINE BURGDORF[1,3], KRISTIJAN CAFUTA, IGOR KLEP[2,3], AND JANEZ POVH[4]

ABSTRACT. This paper presents an algorithm and its implementation in the software package NCSOStools for finding sums of hermitian squares and commutators decompositions for polynomials in noncommuting variables. The algorithm is based on noncommutative analogs of the classical Gram matrix method and the Newton polytope method, which allows us to use semidefinite programming. Throughout the paper several examples are given illustrating the results.

1. INTRODUCTION

The main problem studied in this paper is whether a given real polynomial in free noncommuting variables (nc polynomial) can be decomposed as a sum of hermitian squares and commutators. This question can be reformulated as a semidefinite programming problem. As the size of these semidefinite programs tends to be very large, we present a method to reduce it significantly.

1.1. **Motivation.** The interest in finding decompositions of an nc polynomial as a sum of hermitian squares and commutators is based on the following simple fact. If such a decomposition exists, the given nc polynomial is necessarily trace-positive, i.e., all of its evaluations at tuples of matrices have nonnegative trace. Following Helton's seminal paper [Hel02], this belongs to *free real algebraic geometry* (including *free positivity*) where one is interested in positivity of nc polynomials. Much of today's interest in (free) real algebraic geometry is due to its powerful applications. For instance, the use of sums of squares and the truncated moment problem for polynomial optimization on $\mathbb{R}^n$ established by Lasserre and Parrilo [Las01, Las09, Par03, PS03, Sch05] is nowadays a common fact in real algebraic geometry with applications to control theory, mathematical finance and operations research. In the free context there are many facets of applications as well. A nice survey on connections to control theory, systems engineering and optimization is given by de Oliveira, Helton, McCullough, Putinar [dOHMP08]. Applications of the free case to quantum physics are explained e.g. by Pironio, Navascués, Acín [PNA10] who also consider computational aspects related to sums of hermitian squares (without commutators). Trace-positive nc polynomials fill a gap between these two cases, so we expect a considerable development of their applications in the future.

On the theoretical level, trace-positive nc polynomials occur naturally in von Neumann algebras and functional analysis. For instance, Connes' embedding problem [Con76] on finite

Date: 15/05/2012 at 07:40:13.

2010 *Mathematics Subject Classification.* Primary 90C22, 13J30; Secondary 47A57, 08B20.

Key words and phrases. sum of squares, semidefinite programming, noncommutative polynomial, Matlab toolbox, Newton polytope, free positivity.

[1]Partially supported by the Zukunftskolleg Konstanz.

[2]Supported by the Faculty Research Development Fund (FRDF) of The University of Auckland (project no. 3701119). Partially supported by the Slovenian Research Agency (program no. P1-0222).

[3]Partially supported by the French-Slovene partnership project Proteus 20208ZM.

[4]Supported by the Slovenian Research Agency (program no. P1-0297).

II_1-factors is a question about the existence of a certain type of sum of hermitian squares (sohs) certificates for complex trace-positive nc polynomials [KS08a]. It is widely believed that Connes' conjecture is false and our results will enable us to look for a counterexample using a computer algebra system. In addition, trace-positive nc polynomials arise in the Lieb-Seiringer reformulation of the recently solved [Sta] Bessis-Moussa-Villani (BMV) conjecture [BMV75] from statistical quantum mechanics. Many results on this problem have been obtained with the aid of computer programs – using sums of hermitian squares and commutators decompositions – written in an ad-hoc manner.

As a consequence of this surge of interest in free real algebraic geometry and sums of (hermitian) squares of nc polynomials we have developed `NCSOStools` [CKP11] – an open source Matlab toolbox for solving such problems using *semidefinite programming*. As a side product our toolbox implements symbolic computation with noncommuting variables in Matlab.

1.2. Related work and contribution.

We will denote the convex cone of sums of hermitian squares and commutators by Θ^2. Sum of hermitian squares decompositions were intensively studied by several authors. An outstanding result is due to Helton [Hel02], who has proved that for an nc polynomial $f \in \mathbb{R}\langle \underline{X} \rangle$, we have $f(A_1, \ldots, A_n) \succeq 0$ for *all* symmetric matrices A_i of the same size if and only if f is a sum of hermitian squares. We also refer the reader to [McC01, MP05] for nice alternative proofs. In [KP10] the third and the fourth author presented an algorithm for finding sums of hermitian squares decompositions (without commutators) using a variant of the Gram matrix method. The key ingredient of the method was semidefinite programming together with the Newton chip method to reduce the size of the semidefinite programming problems, which eventually turned out to be linear in the length and in the degree of the nc polynomial. Extending this method we proposed in [BCKP] another variant of the Gram matrix method to answer the question whether $f \in \Theta^2$ holds. However, an important topic that remained open in [BCKP] was how to provide *efficiently* numerical or exact certificates for either $f \in \Theta^2$ or $f \notin \Theta^2$.

Therefore the main contribution of this paper is the following: we present the *tracial Gram matrix method*, tailored for sums of hermitian squares and commutators, to resolve the separability question for Θ^2, using a *cyclic* extension of the *Newton chip method* from [KP10] which reduces the dimensions of the underlying semidefinite programs to a more manageable level. Our method can be understood as a noncommutative generalization of the classical [Rez78] Newton polytope method.

2. Preliminaries

2.1. Words, nc polynomials and involution.

Fix $n \in \mathbb{N}$ and let $\langle \underline{X} \rangle$ be the set of *words* in the n noncommuting letters $X_1, \ldots, X_n$ (including the empty word denoted by 1), i.e., $\langle \underline{X} \rangle$ is the monoid freely generated by $\underline{X} := (X_1, \ldots, X_n)$. We consider linear combinations $\sum_w a_w w$ with $a_w \in \mathbb{R}$, $w \in \langle \underline{X} \rangle$ of words in the n letters $\underline{X}$ which we call *nc polynomials*. The set of all nc polynomials is actually a *free algebra*, which we denote by $\mathbb{R}\langle \underline{X} \rangle$. An element of the form aw where $a \in \mathbb{R} \setminus \{0\}$ and $w \in \langle \underline{X} \rangle$ is called a *monomial* and a its *coefficient*. The length of the longest word in an nc polynomial $f \in \mathbb{R}\langle \underline{X} \rangle$ is the *degree* of f and is denoted by $\deg f$. The set of all nc polynomials of degree $\leq d$ will be denoted by $\mathbb{R}\langle \underline{X} \rangle_{\leq d}$. If an nc polynomial f involves only two variables, we use $\mathbb{R}\langle X, Y \rangle$ instead of $\mathbb{R}\langle X_1, X_2 \rangle$.

We equip $\mathbb{R}\langle \underline{X} \rangle$ with the *involution* $*$ that fixes $\mathbb{R} \cup \{\underline{X}\}$ pointwise and thus reverses words, e.g. $(X_1 X_2^2 X_3 - 2X_3^3)^* = X_3 X_2^2 X_1 - 2X_3^3$. Hence $\mathbb{R}\langle \underline{X} \rangle$ is the $*$-algebra freely generated by n symmetric letters. The involution extends naturally to matrices (in particular, to vectors) over $\mathbb{R}\langle \underline{X} \rangle$. For instance, if $V = (v_i)$ is a (column) vector of nc polynomials $v_i \in \mathbb{R}\langle \underline{X} \rangle$, then V^* is the row vector with components v_i^*. We use V^t to denote the row vector with components v_i.

2.2. **Sums of hermitian squares and commutators.** Let $\operatorname{Sym}\mathbb{R}\langle\underline{X}\rangle$ denote the set of all *symmetric elements*, that is,

$$\operatorname{Sym}\mathbb{R}\langle\underline{X}\rangle := \{f \in \mathbb{R}\langle\underline{X}\rangle \mid f = f^*\}.$$

An nc polynomial of the form g^*g is called a *hermitian square* and the set of all sums of hermitian squares will be denoted by Σ^2. Clearly, $\Sigma^2 \subsetneq \operatorname{Sym}\mathbb{R}\langle\underline{X}\rangle$.

Example 2.1. The nc polynomial $f = X^2 - X^2Y - YX^2 + YX^2Y + XY^2X \in \operatorname{Sym}\mathbb{R}\langle\underline{X}\rangle$, is a sum of hermitian squares, in fact, $f = (X - XY)^*(X - XY) + (YX)^*(YX)$. In particular, $f(A, B)$ is positive semidefinite for all symmetric matrices A, B.

The next notation we need is cyclic equivalence [KS08a] whose definition is motivated by the fact that we are interested in the *trace* of a given nc polynomial under matrix evaluations.

Definition 2.2. An element of the form $[p, q] := pq - qp$, where p, q are polynomials from $\mathbb{R}\langle\underline{X}\rangle$, is a *commutator*. Polynomials $f, g \in \mathbb{R}\langle\underline{X}\rangle$ are called *cyclically equivalent* ($f \overset{\text{cyc}}{\sim} g$) if $f - g$ is a sum of commutators:

$$f - g = \sum_{i=1}^{k}[p_i, q_i] = \sum_{i=1}^{k}(p_iq_i - q_ip_i) \text{ for some } k \in \mathbb{N} \text{ and } p_i, q_i \in \mathbb{R}\langle\underline{X}\rangle.$$

It is clear that $\overset{\text{cyc}}{\sim}$ is an equivalence relation. The following remark motivates its name and shows how to test if given nc polynomials are cyclically equivalent.

Remark 2.3.

(a) For $v, w \in \langle\underline{X}\rangle$, we have $v \overset{\text{cyc}}{\sim} w$ if and only if there are $v_1, v_2 \in \langle\underline{X}\rangle$ such that $v = v_1v_2$ and $w = v_2v_1$. That is, $v \overset{\text{cyc}}{\sim} w$ if and only if w is a cyclic permutation of v.

(b) Polynomials $f = \sum_{w\in\langle\underline{X}\rangle} a_w w$ and $g = \sum_{w\in\langle\underline{X}\rangle} b_w w$ ($a_w, b_w \in \mathbb{R}$) are cyclically equivalent if and only if for each $v \in \langle\underline{X}\rangle$,

$$\sum_{\substack{w\in\langle\underline{X}\rangle \\ w \overset{\text{cyc}}{\sim} v}} a_w = \sum_{\substack{w\in\langle\underline{X}\rangle \\ w \overset{\text{cyc}}{\sim} v}} b_w. \tag{1}$$

Example 2.4. We have $2X^2Y^2X^3 + XY^2X^2 + XY^2X^4 \overset{\text{cyc}}{\sim} 3YX^5Y + YX^3Y$ as

$$\begin{aligned}2X^2Y^2X^3 + XY^2X^2 + XY^2X^4 - (3YX^5Y + YX^3Y)& \\ = [2X^2Y, YX^3] + [XY, YX^4] + [XY, YX^2].&\end{aligned}$$

Definition 2.5. Let

$$\Theta^2 := \{f \in \mathbb{R}\langle\underline{X}\rangle \mid \exists g \in \Sigma^2 : \ f \overset{\text{cyc}}{\sim} g\}$$

denote the convex cone of all nc polynomials cyclically equivalent to a sum of hermitian squares. By definition, the elements in Θ^2 are exactly the nc polynomials which can be written as sums of hermitian squares and commutators.

Example 2.6. Consider $f = X^2Y^2 + XY^2X + XYXY + YX^2Y + YXYX + Y^2X^2 \in \mathbb{R}\langle X, Y\rangle$. This nc polynomial is of the form

$$\begin{aligned} f &= (XYXY + YXYX + XY^2X + YX^2Y) + 2XY^2X + [Y^2X, X] + [X, XY^2] \\ &= (XY + YX)^*(XY + YX) + 2(YX)^*(YX) + [Y^2X, X] + [X, XY^2], \end{aligned}$$

hence $f \in \Theta^2$. In particular, $\operatorname{tr}(f(A, B)) \geq 0$ for all symmetric matrices A, B but in general $f(A, B)$ is not positive semidefinite.

3. The improved tracial Gram matrix method

Testing whether a given $f \in \mathbb{R}\langle \underline{X} \rangle$ is an element of Σ^2 or Θ^2 can be done efficiently by using semidefinite programming as first observed in [KS08b, Section 3], see also [KP10, BCKP]. The method behind it is a variant of the Gram matrix method and arises as a natural extension of the results for sums of hermitian squares (cf. [Hel02, Section 2.2] or [KP10, Theorem 3.1 and Algorithm 1]) or for polynomials in commuting variables [CLR95, Section 2]; see also [Par03]. In this section we present the improved tracial Gram matrix method which is based on a tracial version of the classical Newton polytope used to reduce the size of the underlying semidefinite programming problem. The concrete formulation is a bit technical but the core idea is straightforward and goes as follows. Define the Newton polytope of an nc polynomial f as the Newton polytope of an appropriate interpretation of f as a polynomial in commuting variables. Now apply the Newton polytope method and then lift the obtained set of monomials in commuting variables to a set of monomials in noncommuting variables.

3.1. The cyclic degree.

Our viewpoint focuses on the dual description of the tracial version of the Newton polytope, described by the so-called cyclic-α-degree. This viewpoint clarifies the chosen interpretation of an nc polynomial as a polynomial in commuting variables which is used in the algorithm.

We will need to consider the free monoid $[\underline{x}]$ in *commuting* variables $\underline{x} := (x_1, \dots, x_n)$ and its semigroup algebra $\mathbb{R}[\underline{x}]$ of polynomials. Its monomials are of the form $\underline{x}^{\underline{d}} = x_1^{d_1} \cdots x_n^{d_n}$ for $\underline{d} = (d_1, \dots, d_n) \in \mathbb{N}^n$. There is a natural mapping $\langle \underline{X} \rangle \to [\underline{x}]$. For a given word $w \in \langle \underline{X} \rangle$ its image under this mapping is of the form $\underline{x}^{\underline{d_w}}$, where $d_{w,i}$ denotes how many times X_i appears in w. It is called the *commutative collapse* of w. Similarly, we introduce the commutative collapse of a set of words $V \subseteq \mathbb{R}\langle \underline{X} \rangle$. For $f = \sum_w a_w w \in \mathbb{R}\langle \underline{X} \rangle$ we define the set

$$\operatorname{cc}(f) := \{\underline{x}^{\underline{d_w}} \in [\underline{x}] \mid a_w \neq 0\}.$$

We generalize the degree of an nc polynomial as follows: given $\underline{\alpha} = (\alpha_1, \dots, \alpha_n) \in \mathbb{R}^n$ we define the $\underline{\alpha}$*-degree* $\deg_{\underline{\alpha}}$ of a word $w \in \langle \underline{X} \rangle$ as the standard scalar product between $\underline{\alpha}$ and the exponent of the commutative collapse $\underline{x}^{\underline{d_w}}$ of w, i.e.,

$$\deg_{\underline{\alpha}} w := \sum_{i=1}^{n} \alpha_i d_{w,i} = \langle \underline{\alpha}, \underline{d_w} \rangle. \tag{2}$$

We also set $\deg_{\underline{\alpha}} 0 := -\infty$. Note that for all $\underline{\alpha} \in \mathbb{R}^n$, we have $u \overset{\text{cyc}}{\sim} v \Rightarrow \deg_{\underline{\alpha}} u = \deg_{\underline{\alpha}} v$ and $\deg_{\underline{\alpha}}(uv) = \deg_{\underline{\alpha}} u + \deg_{\underline{\alpha}} v$.

This notion extends naturally to the $\underline{\alpha}$-degree of an nc polynomial $f = \sum_w a_w w \in \mathbb{R}\langle \underline{X} \rangle$:

$$\deg_{\underline{\alpha}} f := \max_{a_w \neq 0} \deg_{\underline{\alpha}} w. \tag{3}$$

As special cases, note that the (total) degree corresponds to the $\underline{\alpha}$ with all ones and the degree in variable X_i corresponds to the standard unit vectors e_i.

Two cyclically equivalent nc polynomials in general do not have the same $\underline{\alpha}$-degree. We therefore modify the definition to obtain the more robust *cyclic-*$\underline{\alpha}$*-degree* $\operatorname{cdeg}_{\underline{\alpha}}$:

$$\operatorname{cdeg}_{\underline{\alpha}} f := \min_{g \overset{\text{cyc}}{\sim} f} \deg_{\underline{\alpha}} g. \tag{4}$$

For instance, for $f = X_1^2 X_2^2 X_1^2 + X_2^4 X_3^4 - X_3^4 X_2^4 + X_1 X_2 - X_2 X_1 \overset{\text{cyc}}{\sim} X_1^4 X_2^2$ we have

$$\deg_{(1,1,3)} f = 16\ ,\ \operatorname{cdeg}_{(1,1,3)} f = 6.$$

Definition 3.1. Let $w \in \mathbb{R}\langle \underline{X} \rangle$. The canonical representative $[w]$ of w is the first with respect to the lexicographic order among words cyclically equivalent to w. For $f = \sum_w a_w w \in \operatorname{Sym} \mathbb{R}\langle \underline{X} \rangle$ we define the *canonical representative* $[f]$ of f as follows:

$$[f] := \sum_{[w]} a_{[w]} [w] \in \mathbb{R}\langle \underline{X} \rangle.$$

That is, $[f]$ contains only canonical representatives of words from f with coefficients

$$a_{[w]} := \sum_{u \overset{\text{cyc}}{\sim} w} a_u.$$

For example, if $f = 2Y^2X^2 - XY^2X + XY - YX$, then $[f] = X^2Y^2$.

The next proposition shows that the cyclic-α-degree is compatible with the equivalence relation $\overset{\text{cyc}}{\sim}$ and equals the degree of the canonical representative.

Proposition 3.2.

(1) *If* $f = \sum_w a_w w \overset{\text{cyc}}{\sim} g = \sum_w b_w w$, *then* $a_{[w]} = b_{[w]}$ *for all* $w \in \langle \underline{X} \rangle$.
(2) *For all* $\alpha \in \mathbb{R}^n$ *and* $f \in \mathbb{R}\langle \underline{X} \rangle$ *we have* $\operatorname{cdeg}_{\underline{\alpha}} f = \deg_{\underline{\alpha}} [f]$.

Proof. Property (1) is obvious. Let us consider (2). Since $f \overset{\text{cyc}}{\sim} [f]$, $\operatorname{cdeg}_{\underline{\alpha}} f \leq \deg_{\underline{\alpha}} [f]$. Suppose there exists $g \overset{\text{cyc}}{\sim} f$ with $\deg_{\underline{\alpha}_0} g < \deg_{\underline{\alpha}_0} [f]$ for some $\underline{\alpha}_0 \in \mathbb{R}^n$. There is a word $[w]$ with $\deg_{\underline{\alpha}_0} [w] = \deg_{\underline{\alpha}_0} [f]$, and the coefficient of $[w]$ in $[f]$ is non-zero. But by the first part of the proposition the same is true for g, hence $\deg_{\underline{\alpha}_0} g \geq \deg_{\underline{\alpha}_0} [f]$, which is a contradiction. ■

3.2. **The tracial Newton polyotope.** Given a polynomial $f \in \mathbb{R}[\underline{x}]$ (in commuting variables) the *Newton polytope* $N(f)$ consists of all integer lattice points in the convex hull of the degrees $\underline{d} = (d_1, \ldots, d_n)$ of words appearing in f, considered as vectors in $\mathbb{R}^n$ (see e.g. [Rez78] for details). That is, for $f = \sum_{\underline{d}} a_{\underline{d}} \underline{x}^{\underline{d}} \in \mathbb{R}[\underline{x}]$,

$$N(f) := \mathbb{Z}^n \cap \operatorname{conv}\big(\{\underline{d} \in \mathbb{Z}^n \mid a_{\underline{d}} \neq 0\}\big).$$

We will also refer to the set $\frac{1}{2}N(f) := \{\underline{d} \in \mathbb{Z}^n \mid 2\underline{d} \in N(f)\}$. Alternatively, by dualization, one can describe the Newton polytope via the α-degree, namely

$$N(f) = \mathbb{Z}^n \cap \operatorname{conv}\big(\{\underline{d} \in \mathbb{Z}^n \mid \deg_{\underline{\alpha}}(\underline{x}^{\underline{d}}) \leq \deg_{\underline{\alpha}}(f) \quad \text{for all } \underline{\alpha} \in \mathbb{R}^n\}\big).$$

Similarly, $N(S)$ and $\frac{1}{2}N(S)$ are defined, where S is a set of words in commuting variables.

By dualization one immediately derives the following lemma.

Lemma 3.3. *A word* $w \in \langle \underline{X} \rangle$ *with comutative collapse* $\underline{x}^{\underline{d}_w}$ *satisfies* $\deg_{\underline{\alpha}}(w) \leq \operatorname{cdeg}_{\underline{\alpha}}(f)$ *for all* $\underline{\alpha} \in \mathbb{R}^n$ *if and only if* $\underline{d}_w$ *is contained in the convex hull of the vectors* $\{\underline{d}_v \mid v \in \operatorname{cc}([f])\}$.

In other words, the *tracial Newton polytope* of an nc polynomial $f \in \mathbb{R}\langle \underline{X} \rangle$ is given by the classical Newton polytope for the commutative collapse of the canonical representative $[f]$ of f. Hence a word $w \in \langle \underline{X} \rangle$ should be included in the sum of hermitian squares and commutators factorization for a given noncommutative polynomial f if and only if the exponent $\underline{d}_w$ of its commutative collapse is contained in one half times the Newton polytope of the commutative collapse of $[f]$. In fact, this will be shown in Theorem 3.7, where we present the augmented tracial Gram matrix method.

Example 3.4. Let $f = 1 - XY^3 + Y^3X + 2Y^2 - 4X^5 \in \mathbb{R}\langle X, Y \rangle$. Then $[f] = 1 + 2Y^2 - 4X^5$,

$$\operatorname{cc}(f) = \{1, xy^3, y^2, x^5\} \subseteq [x, y], \quad \operatorname{cc}([f]) = \{1, y^2, x^5\} \subseteq [x, y],$$

$$\begin{aligned} N(\mathrm{cc}([f])) &= \mathbb{Z}^2 \cap \mathrm{conv}\big(\{(0,0),\,(0,2),\,(5,0)\}\big) \\ &= \big\{(0,0),\,(1,0),\,(2,0),\,(3,0),\,(4,0),\,(5,0),\,(0,1),\,(1,1),\,(2,1),\,(0,2)\big\}. \end{aligned}$$

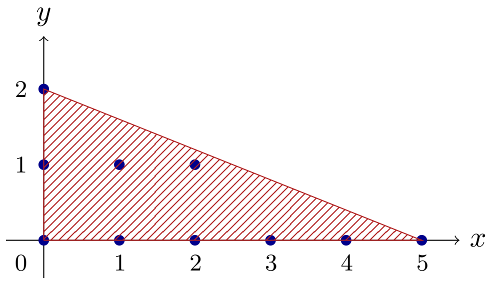

FIGURE 1. The Newton polytope of $f = 1 - XY^3 + Y^3X + 2Y^2 - 4X^5$

We note that by taking the canonical representative $[f]$ instead of f itself we get a unique Newton polytope for f which is also the smallest Newton polytope among all Newton polyotpes of possible interpretations of f in $\mathbb{R}[\underline{x}]$.

3.3. **The tracial Gram matrix method.** In this section we present the improved tracial Gram matrix method based on the tracial Newton polytope. That is, to construct a tracial Gram matrix for an nc polynomial $f \in \mathbb{R}\langle \underline{X}\rangle$ we will only consider words $w \in \langle \underline{X}\rangle$ whose exponent $\underline{d}_w$ of its commutative collapse is contained in one half times the tracial Newton polytope of f. This will be expressed by the cyclic-α-degree using the following corollary, which is an immediate consequence of Proposition 3.2(2) and Lemma 3.3.

Corollary 3.5. *Let $f \in \mathbb{R}\langle \underline{X}\rangle$ be an nc polynomial. Then*

$$\mathrm{cc}(W) = \big\{\underline{x}^{\underline{d}} \mid \underline{d} \in \frac{1}{2}N(\mathrm{cc}([f]))\big\} \tag{5}$$

for the vector W consisting of all words $w \in \langle \underline{X}\rangle$ satisfying

$$2\deg_{\underline{\alpha}}(w) \le \mathrm{cdeg}_{\underline{\alpha}}(f) \quad \text{for all } \underline{\alpha} \in \mathbb{R}^n.$$

Example 3.6. For $f = 1 - XY^3 + Y^3X + 2Y^2 - 4X^5 \in \mathbb{R}\langle X, Y\rangle$ from Example 3.4 we have $\frac{1}{2}N(\mathrm{cc}([f])) = \big\{(0,0),\,(0,1),\,(1,0),\,(2,0)\big\}$. One can easily verify that $W = \begin{bmatrix}1 & Y & X & X^2\end{bmatrix}^t$ and hence (5) holds.

The next theorem is the theoretical underpinning of our improved tracial Gram matrix method.

Theorem 3.7. *Suppose $f \in \mathbb{R}\langle \underline{X}\rangle$. Then $f \in \Theta^2$ if and only if there exists a positive semidefinite matrix G such that*

$$f \overset{\mathrm{cyc}}{\sim} W^*GW, \tag{6}$$

where W is a vector consisting of all words $w \in \langle \underline{X}\rangle$ satisfying

$$2\deg_{\underline{\alpha}}(w) \le \mathrm{cdeg}_{\underline{\alpha}}(f) \quad \text{for all } \underline{\alpha} \in \mathbb{R}^n. \tag{7}$$

Furthermore, given such a positive semidefinite matrix G of rank r, one can construct nc polynomials $g_1, \ldots, g_r \in \mathbb{R}\langle \underline{X}\rangle$ with $f \overset{\mathrm{cyc}}{\sim} \sum_{i=1}^r g_i^* g_i$.

For the proof we need one last ingredient, namely that the cyclic-α-degree of a sum of hermitian squares is equal to its α-degree.

Lemma 3.8. *If $f \overset{\mathrm{cyc}}{\sim} g = \sum_i g_i^* g_i$, then $\mathrm{cdeg}_{\underline{\alpha}} f = \deg_{\underline{\alpha}} g$.*

Proof. If $g = 0$ then lemma is true for trivial reasons. Otherwise, by definition, $\operatorname{cdeg}_{\underline{\alpha}} f \leq \deg_{\underline{\alpha}} g$ for all $\underline{\alpha} \in \mathbb{R}^n$. Suppose there exists $\underline{\alpha}_0 \in \mathbb{R}^n$ with $\operatorname{cdeg}_{\underline{\alpha}_0} f < \deg_{\underline{\alpha}_0} g$. For $[f] \overset{\text{cyc}}{\sim} f$ we have $\operatorname{cdeg}_{\underline{\alpha}_0} f = \deg_{\underline{\alpha}_0}[f] < \deg_{\underline{\alpha}_0} g =: 2\Delta \neq 0$. Let p_i be the homogeneous part of g_i with $\underline{\alpha}_0$-degree equal to Δ and $r_i = g_i - p_i$. Then $\deg_{\underline{\alpha}_0}(r_i) < \Delta$ and

$$[f] \overset{\text{cyc}}{\sim} \sum g_i^* g_i = \sum (p_i + r_i)^*(p_i + r_i) = \sum p_i^* p_i + \sum p_i^* r_i + \sum r_i^* p_i + \sum r_i^* r_i. \tag{8}$$

Since each word w in $p_i^* r_i$, $r_i^* p_i$ and $r_i^* r_i$ has $\deg_{\underline{\alpha}_0} w < 2\Delta$, none of these can be cyclically equivalent to a nontrivial word in $p_i^* p_i$, because each nontrivial word in $p_i^* p_i$ has $\underline{\alpha}_0$-degree equal to $2\Delta \neq 0$ (note that for each i, $p_i^* p_i \overset{\text{cyc}}{\not\sim} 0$ or $p_i = 0$ due to [KS08b, Lemma 3.2]). Similarly, by assumption there is no word in $[f]$ with $\underline{\alpha}_0$-degree equal to 2Δ. Thus

$$0 \overset{\text{cyc}}{\sim} \sum p_i^* p_i, \quad [f] \overset{\text{cyc}}{\sim} \sum p_i^* r_i + \sum r_i^* p_i + \sum r_i^* r_i.$$

However, [KS08b, Lemma 3.2] implies $p_i = 0$ for all i contradicting $\deg_{\underline{\alpha}_0} g = 2\Delta$. ∎

Proof of Theorem 3.7. If $f \overset{\text{cyc}}{\sim} g = \sum_i g_i^* g_i \in \Sigma^2$, then $\deg_{\underline{\alpha}} g = \operatorname{cdeg}_{\underline{\alpha}} f$ for all $\underline{\alpha} \in \mathbb{R}^n$, as follows from Lemma 3.8. Therefore, $2 \deg_{\underline{\alpha}} g_i \leq \deg_{\underline{\alpha}} g = \operatorname{cdeg}_{\underline{\alpha}} f$ for all i and for all $\underline{\alpha} \in \mathbb{R}^n$, hence g_i contains only words satisfying (7). Write $g_i = G_i^t W$, where G_i^t is the (row) vector consisting of the coefficients of g_i. Then $g_i^* g_i = W^t G_i G_i^t W$ and, by setting $G := \sum_i G_i G_i^t$, property (6) clearly holds. The inverse of this claim is obvious.

Given a positive semidefinite $G \in \mathbb{R}^{N \times N}$ of rank r satisfying (6), write $G = \sum_{i=1}^r G_i G_i^t$ for $G_i \in \mathbb{R}^{N \times 1}$. Defining $g_i := G_i^t W$ yields $f \overset{\text{cyc}}{\sim} \sum_{i=1}^r g_i^* g_i$. ∎

A matrix G satisfying (6) is called a *tracial Gram matrix* for f, which motivates the name of the method. For an nc polynomial $f \in \mathbb{R}\langle \underline{X} \rangle$ the tracial Gram matrix is in general *not* unique, hence determining whether $f \in \Theta^2$ amounts to finding *a* positive semidefinite tracial Gram matrix from the affine set of all tracial Gram matrices for f. Problems like this can in theory be solved *exactly* using quantifier elimination. However, this only works for problems of small size, so a numerical approach is needed in practice. Thus we turn to semidefinite programming, which has become a standard tool in mathematical optimization in the last two decades. The readers not familiar with this topic are referred to [WSV00, Tod01, VB96].

4. Implementation and computational algorithms

In this section we discuss an algorithm based on the tracial Gram matrix method for testing the membership in Θ^2 and its improvement using the tracial version of the Newton polytope which we call the Newton cyclic chip method.

4.1. Sums of hermitian squares and commutators and semidefinite programming.

In this subsection we present a *conceptual algorithm* based on semidefinite programming for checking whether an nc polynomial of degree $\leq 2d$ is cyclically equivalent to a sum of hermitian squares. Following Theorem 3.7 we must determine whether there exists a positive semidefinite matrix G such that $f \overset{\text{cyc}}{\sim} W^* G W$. This is a semidefinite feasibility problem in the matrix variable G, where the constraints $\langle A_i, G \rangle = b_i$ are essentially equations (1).

Example 4.1. Let

$$\begin{aligned} f &= 2XY^2XYX + 4XYX^2YX + XY^4X + 2YXY^2X^2 \\ &= (Y^2X + 2XYX)^*(Y^2X + 2XYX) - 2XYXY^2X + 2YXY^2X^2 \\ &\overset{\text{cyc}}{\sim} (Y^2X + 2XYX)^*(Y^2X + 2XYX). \end{aligned}$$

If we take $W = \begin{bmatrix} XYX & Y^2X \end{bmatrix}^t$, then a tracial Gram matrix G for f is obtained as a solution to the following semidefinite program (SDP):

$$\begin{array}{rrcl} \inf & \langle C, G\rangle & & \\ \text{s. t.} & & & \\ XYX^2YX: & G_{1,1} & = & 4 \\ XYXY^2X: & G_{1,2} & = & 2 \\ XY^2XYX: & G_{2,1} & = & 2 \\ XY^4X: & G_{2,2} & = & 1 \\ & G & \succeq & 0. \end{array}$$

Remark 4.2. The matrix C in Example 4.1 is arbitrary. One can use $C = I$, a commonly used heuristic for matrix rank minimization [RFP10]. Often, however, a solution of *high-rank* is desired (this is the case when we want to extract *rational* certificates, a topic we shall discuss elsewhere). Then $C = 0$ is used, since under a strict feasibility assumption the interior point methods yield solutions in the relative interior of the optimal face, which is in our case the whole feasibility set. If strict complementarity is additionally provided, the interior point methods lead to the analytic center of the feasibility set [HdKR02]. Even though these assumptions do not always hold for the instances of SDPs we construct, in our experiments the choice $C = 0$ in the objective function almost always gave a solution of higher rank than the choice $C = I$.

Remark 4.3. As we restrict our attention to nc polynomials which are cyclically equivalent to symmetric nc polynomials (the others are clearly not in Θ^2), we may always merge the equations corresponding to a particular word and its involution, e.g. in Example 4.1 we can replace the second and the third equation with a single constraint $G_{1,2} + G_{2,1} = 4$.

We formalize the lesson from Remark 4.3 as follows:

Lemma 4.4. *If $f = \sum_w a_w w \in \Theta^2$, then for every $v \in \langle \underline{X}\rangle$*

$$\sum_{w \overset{\text{cyc}}{\sim} v} a_w = \sum_{w \overset{\text{cyc}}{\sim} v^*} a_w. \tag{9}$$

Corollary 4.5. *Given $f \in \mathbb{R}\langle \underline{X}\rangle$ we have:*

(1) *If f does not satisfy (9), then $f \notin \Theta^2$.*
(2) *If f satisfies (9), then we can determine whether $f \in \Theta^2$ by solving the following SDP with only symmetric constraints:*

$$\begin{array}{lrcl} \inf & \langle C, G\rangle & & \\ \text{s. t.} & \displaystyle\sum_{\substack{p,q,\ p^*q \overset{\text{cyc}}{\sim} v \\ \vee\ p^*q \overset{\text{cyc}}{\sim} v^*}} G_{p,q} & = & \displaystyle\sum_{w \overset{\text{cyc}}{\sim} v} (a_w + a_{w^*}),\ \forall v \in W \\ & G & \succeq & 0. \end{array} \tag{CSOHS$_{\text{SDP}}$}$$

Thus we are left with the construction of W, which is a linear porgramming problem by the following lemma.

Lemma 4.6. *Verifying whether $w \in \langle \underline{X}\rangle$ satisfies (7) is a linear programming problem.*

Proof. Indeed, let $f = \sum a_v v \in \mathbb{R}\langle \underline{X}\rangle$ of degree $\leq 2d$ be given and let $w \in \langle \underline{X}\rangle$ be a word for which we want to verify (7). Then the following is true:

$$\begin{array}{rrcl} & 2\deg_{\underline{\alpha}} w & \leq & \operatorname{cdeg}_{\underline{\alpha}} f \quad \text{for all } \underline{\alpha} \in \mathbb{R}^n \\ \Leftrightarrow & 2\deg_{\underline{\alpha}} w & \leq & \deg_{\underline{\alpha}}[f] \quad \text{for all } \underline{\alpha} \in \mathbb{R}^n \\ \Leftrightarrow & 2\langle \underline{\alpha}, \underline{d}_w\rangle & \leq & \max_{v \in \operatorname{cc}([f])}\{\langle \underline{\alpha}, \underline{d}_v\rangle\} \quad \text{for all } \underline{\alpha} \in \mathbb{R}^n \\ \Leftrightarrow & 0 & \leq & \inf_{\underline{\alpha} \in \mathbb{R}^n} \max_{v \in \operatorname{cc}([f])}\{\langle \underline{\alpha}, \underline{d}_v - 2\underline{d}_w\rangle\} \\ \Leftrightarrow & 0 & \leq & \inf\{t \mid \langle \underline{\alpha}, \underline{d}_v - 2\underline{d}_w\rangle \ \leq\ t,\ \forall v \in \operatorname{cc}([f]),\ \underline{\alpha} \in \mathbb{R}^n\}. \end{array}$$

Verifying the last inequality can be done in two steps: (i) solve the linear programming problem

$$\begin{array}{rrcl} & t_{\mathrm{opt}} & = & \inf\ t \\ \text{s.t.} & \langle \underline{\alpha}, \underline{d}_v - 2\underline{d}_w \rangle & \leq & t,\ \forall v \in \mathrm{cc}([f]) \\ & \underline{\alpha} & \in & \mathbb{R}^n, \end{array} \tag{LP}$$

and (ii) check if $t_{\mathrm{opt}} \geq 0$. ■

The conceptual algorithm to determine whether a given nc polynomial is cyclically equivalent to a sum of hermitian squares (the *tracial Gram matrix method*) is described in Algorithm 1:

```
INPUT: $f \in \mathbb{R}\langle \underline{X} \rangle$ with $f = \sum_{w \in \langle \underline{X} \rangle} a_w w$, where $a_w \in \mathbb{R}$.
STEP 1: If $f$ does not satisfy (9), then $f \notin \Theta^2$. Stop.
STEP 2: Construct $W$.
STEP 3: Construct data $A_v, b_v, C$ corresponding to (CSOHS$_{\mathrm{SDP}}$).
STEP 4: Solve (CSOHS$_{\mathrm{SDP}}$) to obtain $G$. If it is not feasible, then $f \notin \Theta^2$.
        Stop.
STEP 5: Compute a decomposition $G = R^t R$.
OUTPUT: Sum of hermitian squares cyclically equivalent to $f$: $f \overset{\mathrm{cyc}}{\sim} \sum_i g_i^* g_i$,
where $g_i$ denotes the $i$-th component of $RW$.
```

Algorithm 1: The tracial Gram matrix method for finding Θ^2-certificates.

In Step 5 we can take different decompositions, e.g. a Cholesky decomposition (which is not unique if G is not positive definite), the eigenvalue decomposition, etc.

The implementation of Step 2 of the tracial Gram matrix method requires according to Lemma 4.6 solving a small linear programming problem (LP) for each candidate w for the set W. Each linear program has $n+1$ variables with card$\,(\mathrm{cc}([f]))$ linear inequalities. Solving such linear programs can be done easily for the problems we are interested in (note that due to other limitations we are considering only nc polynomials f with $n+d \leq 20$). If f is an nc polynomial in 2 variables and has 10000 monomials, then we obtain a linear program (LP) in 3 variables with at most 10000 constraints. Nowadays LP solvers solve such problems easily (within a second); see [Mit03] for a comparison of the state-of-the-art LP solvers and [MPRW09] for a list of efficient alternative methods to solve semidefinite programs. If $f \in \mathbb{R}\langle \underline{X} \rangle$ is polynomial in n variables with $\deg f = 2d$ then it is enough to consider at Step 2 only words $w \in \langle \underline{X} \rangle$ such that $[w]$ has degree at most d. Since there are $\binom{n+d}{d}$ different $[w]$ of this type Step 2 might be still time consuming.

We present the details about the implementation of Step 2 of Algorithm 1 in Algorithm 2 below (the *Newton cyclic chip method*).

Remark 4.7. As mentioned above we need to run Step 3.1 $\binom{n+d}{d}$-times. For each word w which satisfies the condition in Step 3.2 we add at most $d!$ words to W. Nevertheless, the length of the constructed W is usually much smaller than the number of all words $w \in \langle \underline{X} \rangle$ of degree $\leq d$. On the other hand, it is often much larger than the number of words obtained by the Newton chip method [KP10] developed for the sum of hermitian squares decomposition.

4.2. **Software implementation.**

Coding the tracial Gram matrix method together with the Newton cyclic chip method needs to be done carefully due to several potential bottlenecks. Obviously the most expensive part of the Gram matrix method is Step 4 (solving CSOHS$_{\mathrm{SDP}}$). Its complexity is determined by the order of the matrix variable G and the number of linear

```
INPUT: $f \in \mathbb{R}\langle \underline{X} \rangle$ with $\deg f \leq 2d$, $f = \sum_{w \in \langle \underline{X} \rangle} a_w w$, where $a_w \in \mathbb{R}$.
STEP 1: Let $V_d$ be the vector of all words in $[\underline{x}]$ with degree $\leq d$.
STEP 2: $W := \varnothing$.
STEP 3: For every $w \in V_d$:
        STEP 3.1: Solve (LP) related to $w$ to obtain $t_{\text{opt}}$.
        STEP 3.2: If $t_{\text{opt}} \geq 0$ then
                  $W = W \cup \{$all (noncommutative) permutations of $w\}$.
OUTPUT: $W$.
```

Algorithm 2: The Newton cyclic chip method

equations. Both parameters are strongly related to the vector W from Step 2. Indeed, the order of G is exactly the length $|W|$ and the number of linear equations is at least $\frac{|W|^2}{(d+1)(2d-1)!}$. This follows from the fact that for each product $u^*v,\ u,v \in W$ there are at most $d+1$ pairs u_i, v_i such that $u_i^* v_i = u^*v$ and at most $(2d-1)!$ cyclically equivalent products.

The vector W constructed by the Newton cyclic chip method is in general the best possible and is the default procedure used by `NCcycSos` in our package `NCSOStools` [CKP11]. `NCcycSos` takes an nc polynomial as input and returns the answer if it is a member of Θ^2. It might be time consuming, as we have already pointed out in Remark 4.7. However, if we know in advance that it is enough to consider products u^*v for some V and $u, v \in V (\subseteq W)$, then we can add this V as an input to `NCcycSos` and skip Step 2 in the tracial Gram matrix method.

Remark 4.8. In a special case we can construct a further reduced vector W. Namely, if we know that for a representation $f \overset{\text{cyc}}{\sim} g \in \Sigma^2$ we have that $\sum_{w \overset{\text{cyc}}{\sim} v^*v} g_w \neq 0$ for all hermitian squares v^*v appearing in g, then we can construct W by a slight generalization of the Newton chip method from [KP10]. In this case we take the right chips satisfying (7) of all hermitian squares which are cyclically equivalent to words from f instead of all words $w \in \langle \underline{X} \rangle$ satisfying (7). This works e.g. for the BMV polynomials[1] but does not work for e.g.

$$f = 1 - 4XYX + 2X^2 + X^2Y^4X^2 \overset{\text{cyc}}{\sim} 2(XY - X)(YX - X) + (X^2Y^2 - 1)(Y^2X^2 - 1).$$

In fact, the hermitian square $2XY^2X$ cancels with $-X^2Y^2$ and $-Y^2X^2$ and we don't get the necessary words XY and YX in W by applying the Newton chip method.

We point out that in general the semidefinite program (CSOHS$_{\text{SDP}}$) might have no strictly feasible points. Absence of (primal) strictly feasible points might cause numerical difficulties while solving (CSOHS$_{\text{SDP}}$). However, as in [KP10], we can enforce strong duality which is crucial for all SDP solvers by setting the matrix C in (CSOHS$_{\text{SDP}}$) equal to I (actually any full rank matrix will do); see [KP10, Section 4.1] for details. Another source of numerical problems is the infeasibility of (CSOHS$_{\text{SDP}}$), which is the case when $f \notin \Theta^2$. We point out that SDP solvers which are supported by `NCSOStools` have easily overcome these difficulties on all tested instances.

Our implementation of the Newton cyclic chip method is augmented by an additional test used to further reduce the length of W. Indeed, if $w \in W$ satisfies the following properties:

(a) if $u^*v \overset{\text{cyc}}{\sim} w^*w$ for some $u, v \in W$, then $u = v$ (i.e., any product cyclically equivalent to w^*w is a hermitian square);
(b) neither w^*w nor any other product cyclically equivalent to w^*w appears in f,

[1]A BMV polynomial $S_{m,k}(X, Y)$ is the sum of all words in X, Y of total degree m and degree k in Y.

then we can delete w from W, and also all u with $u^*u \overset{\text{cyc}}{\sim} w^*w$. This test is implemented in `NCcycSos` and is run before solving (CSOHS$_{\text{SDP}}$). It amounts to finding (iteratively) all equations of the type $\langle A_w, G\rangle = 0$ with A_w diagonal.

Example 4.9. Consider the nc polynomial $f = 4X^2Y^{10} + 2XY^2XY^4 + 4XY^6 + Y^2$ of degree 12. There are 127 words in 2 variables of degree ≤ 6. Using the Newton cyclic chip method (Algorithm 2) we get only 16 monomials and after the additional test mentioned in the previous paragraph, we are reduced to only 12 words in W as we can see with the aid of `NCSOStools`:

```
>> NCvars x y
>> f = 4*x^2*y^10 + 2*x*y^2*x*y^4 + 4*x*y^6 + y^2;
>> par.precision = 1e-4;
>> [IsCycEq,G,W,sohs,g] = NCcycSos(f, par)
```

This yields a vector `sohs` $= [XY^5 + Y + Y^5X \quad YXY^2 \quad Y^2XY \quad XY^5 - Y^5X]^t$ of nc polynomials g_i with $f \overset{\text{cyc}}{\sim} \sum_i g_i^* g_i =$ `g`; a Gram matrix G for the monomial vector W and `IsCycEq` $= 1$ since f is an element of Θ^2.

Example 4.10. Consider the BMV polynomial $f = S_{8,2}(X, Y)$. To prove that $f \in \Theta^2$ with the aid of `NCSOStools`, proceed as follows:

(1) Define two noncommuting variables:

```
>> NCvars x y
```

(2) Our nc polynomial f is constructed using `BMV(8,2)`. For a numerical test whether $f \in \Theta^2$, run

```
>> params.obj = 0;
>> [IsCycEq,G0,W,sohs,g,SDP_data] = NCcycSos(BMV(8,2), params);
```

This yields a *floating point* Gram matrix G_0

$$G_0 = \begin{bmatrix} 3.9135 & 2.0912 & -0.1590 & 0.9430 \\ 2.0912 & 4.4341 & 1.0570 & -0.1298 \\ -0.1590 & 1.0570 & 4.1435 & 1.9088 \\ 0.9430 & -0.1298 & 1.9088 & 4.0865 \end{bmatrix}$$

for the word vector

$$W = \begin{bmatrix} X^3Y & X^2YX & XYX^2 & YX^3 \end{bmatrix}^t.$$

The rest of the output: `IsCycEq` $= 1$ since f is (numerically) an element of Θ^2; `sohs` is a vector of nc polynomials g_i with $f \overset{\text{cyc}}{\sim} \sum_i g_i^* g_i =$ `g`; `SDP_data` is the SDP data for (CSOHS$_{\text{SDP}}$) constructed from f.

(3) To obtain rational decomposition for f we need to round and project the obtained floating point solution `G0`. This can be done by feeding `G0` and `SDP_data` into `RprojRldlt`. This rationalization scheme is based on the Peyrl-Parrilo technique [PP08] and will be described elsewhere; see also [KLYZ12]. Note that their results are stated for SDPs arising from sum of squares problems, but they carry over verbatim to the setting of (the seemingly more) general SDPs.

```
>> [G,L,D,P,err]=RprojRldlt(G0,SDP_data,true)
```

This produces a rational Gram matrix `G` for f with respect to W and its LDU decomposition $PLDL^tP^t$, where P is a permutation matrix, L lower unitriangular, and D a diagonal matrix with positive entries. We caution the reader that `L,D`, and `G` are cells, each containing numerators and denominators separately as a matrix. Finally, the obtained rational sum of hermitian squares certificate for $f = S_{8,2}(X,Y)$ is

$$f \overset{\text{cyc}}{\sim} \sum_{i=1}^{4} \lambda_i g_i^* g_i,$$

where

$$\begin{aligned} g_1 &= X^3Y + \tfrac{1}{2}X^2YX + \tfrac{1}{4}YX^3 & g_3 &= XYX^2 + \tfrac{13}{22}YX^3 \\ g_2 &= X^2YX + \tfrac{1}{3}XYX^2 - \tfrac{1}{6}YX^3 & g_4 &= YX^3 \end{aligned}$$

and

$$\lambda_1 = 4, \qquad \lambda_2 = 3, \qquad \lambda_3 = \frac{11}{3}, \qquad \lambda_4 = \frac{105}{44}.$$

5. Conclusions

In this paper we considered polynomials in free noncommuting variables which can be decomposed as a sum of hermitian squares and commutators. We presented a systematic way of finding such a decomposition using our open source computer algebra system NCSOStools, freely available at http://ncsostools.fis.unm.si/. The main part of the method – a variant of the classical Gram matrix method – is given by the construction of a semidefinite program. Its solution (if it exists) yields a numerical certificate for the decomposition. The presented Newton cyclic chip method is used to reduce the size of the underlying semidefinite program using Newton polytopes and linear programming. The results are illustrated by numerous examples which also provide demonstrations how to use the proposed algorithm with our computer algebra system NCSOStools.

Acknowledgments

The authors thank the three anonymous referees whose attentive comments helped improve the paper.

References

[BCKP] S. Burgdorf, K. Cafuta, I. Klep, and J. Povh. The tracial moment problem and trace-optimization of polynomials. To appear in *Math. Program.*, http://www.optimization-online.org/DB_HTML/2010/04/2595.html. 2, 4

[BMV75] D. Bessis, P. Moussa, and M. Villani. Monotonic converging variational approximations to the functional integrals in quantum statistical mechanics. *J. Math. Phys.*, 16(11):2318–2325, 1975. 2

[CKP11] 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):363–380, 2011. http://ncsostools.fis.unm.si/ 2, 10

[CLR95] M.D. Choi, T.Y. Lam, and B. Reznick. Sums of squares of real polynomials. In B. Jacob and A. Rosenberg, editors, *K-theory and algebraic geometry: connections with quadratic forms and division algebras (Santa Barbara, CA, 1992)*, volume 58 of *Proc. Sympos. Pure Math.*, pages 103–126. Amer. Math. Soc., Providence, RI, 1995. 4

[Con76] A. Connes. Classification of injective factors. Cases II_1, II_∞, III_λ, $\lambda \neq 1$. *Ann. of Math. (2)*, 104:73–115, 1976. 2

[dOHMP08] M.C. de Oliveira, J.W. Helton, S. McCullough, and M. Putinar. Engineering systems and free semi-algebraic geometry. In M. Putinar and S. Sullivant, editors, *Emerging applications of algebraic geometry*, volume 149 of *IMA Vol. Math. Appl.*, pages 17–61. Springer, New York, 2008. 1

[HdKR02] M. Halická, E. de Klerk, and C. Roos. On the convergence of the central path in semidefinite optimization. *SIAM J. Optim.*, 12(4):1090–1099, 2002. 8

[Hel02] J.W. Helton. "Positive" noncommutative polynomials are sums of squares. *Ann. of Math. (2)*, 156(2):675–694, 2002. 1, 2, 4

[KLYZ12] E. Kaltofen, B. Li, Z. Yang, and L. Zhi. Exact certification in global polynomial optimization via sums-of-squares of rational functions with rational coefficients. *J. Symbolic Comput.*, 47(1):1–15, 2012. 11

[KP10] I. Klep and J. Povh. Semidefinite programming and sums of hermitian squares of noncommutative polynomials. *J. Pure Appl. Algebra*, 214:740–749, 2010. 2, 4, 9, 10

[KS08a] I. Klep and M. Schweighofer. Connes' embedding conjecture and sums of Hermitian squares. *Adv. Math.*, 217(4):1816–1837, 2008. 2, 3

[KS08b] I. Klep and M. Schweighofer. Sums of Hermitian squares and the BMV conjecture. *J. Stat. Phys.*, 133(4):739–760, 2008. 4, 7

[Las01] J. B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM J. Optim.*, 11(3):796–817, 2000/01. 1

[Las09] J. B. Lassere. *Moments, positive polynomials and their applications*, volume 1 of *Imperial College Press Optimization Series*. Imperial College Press, London, 2009. 1

[McC01] S. McCullough. Factorization of operator-valued polynomials in several non-commuting variables. *Linear Algebra Appl.*, 326(1-3):193–203, 2001. 2

[Mit03] D. Mittelmann. An independent benchmarking of SDP and SOCP solvers. *Math. Program. B*, 95:407–430, 2003. http://plato.asu.edu/bench.html 9

[MP05] S. McCullough and M. Putinar. Noncommutative sums of squares. *Pacific J. Math.*, 218(1):167–171, 2005. 2

[MPRW09] J. Malick, J. Povh, F. Rendl, and A. Wiegele. Regularization methods for semidefinite programming. *SIAM J. Optim.*, 20(1):336–356, 2009. 9

[Par03] P.A. Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Math. Program.*, 96(2, Ser. B):293–320, 2003. 1, 4

[PNA10] S. Pironio, M. Navascués, and A. Acín. Convergent relaxations of polynomial optimization problems with noncommuting variables. *SIAM J. Optim.*, 20(5):2157–2180, 2010. 1

[PP08] H. Peyrl and P.A. Parrilo. Computing sum of squares decompositions with rational coefficients. *Theoret. Comput. Sci.*, 409(2):269–281, 2008. 11

[PS03] P.A. Parrilo and B. Sturmfels. Minimizing polynomial functions. In *Algorithmic and quantitative real algebraic geometry (Piscataway, NJ, 2001)*, volume 60 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pages 83–99. Amer. Math. Soc., Providence, RI, 2003. 1

[Rez78] B. Reznick. Extremal PSD forms with few terms. *Duke Math. J.*, 45(2):363–374, 1978. 2, 5

[RFP10] B. Recht, M. Fazel, and P. A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Rev.*, 52(3):471–501, 2010. 8

[Sch05] M. Schweighofer. Optimization of polynomials on compact semialgebraic sets. *SIAM J. Optim.*, 15(3):805–825, 2005. 1

[Sta] H.R. Stahl. Proof of the BMV conjecture. http://arxiv.org/abs/1107.4875 2

[Tod01] M. J. Todd. Semidefinite optimization. *Acta Numer.*, 10:515–560, 2001. 7

[VB96] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Rev.*, 38(1):49–95, 1996. 7

[WSV00] H. Wolkowicz, R. Saigal, and L. Vandenberghe. *Handbook of semidefinite programming*. International Series in Operations Research & Management Science, 27. Kluwer Academic Publishers, Boston, MA, 2000. Theory, algorithms, and applications. 7

Sabine Burgdorf, EPFL, SB – MATHGEOM – EGG, 1015 Lausanne, Switzerland

E-mail address: sabine.burgdorf@epfl.ch

Kristijan Cafuta, Univerza v Ljubljani, Fakulteta za elektrotehniko, Laboratorij za uporabno matematiko, Tržaška 25, 1000 Ljubljana, Slovenia

E-mail address: kristijan.cafuta@fe.uni-lj.si

Igor Klep, The University of Auckland, Department of Mathematics, Private Bag 92019, Auckland 1142, New Zealand

E-mail address: igor.klep@auckland.ac.nz

Janez Povh, Fakulteta za informacijske študije v Novem mestu, Novi trg 5, 8000 Novo mesto, Slovenia

E-mail address: janez.povh@fis.unm.si

NOT FOR PUBLICATION

Contents