Theoretical Computer Science 409 (2008) 269–281

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Computing sum of squares decompositions with rational coefficients

Helfried Peyrl [a,*], Pablo A. Parrilo [b]

[a] Automatic Control Laboratory, ETH Zürich, Physikstrasse 3, 8092 Zürich, Switzerland

[b] Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, 77 Massachusetts Ave., Cambridge, MA 02139-4307, USA

ARTICLE INFO

Keywords:
Symbolic–numerical methods
Algebraic geometry
Sum of squares
Semidefinite programming

ABSTRACT

Sum of squares (SOS) decompositions for nonnegative polynomials are usually computed numerically, using convex optimization solvers. Although the underlying floating point methods in principle allow for numerical approximations of arbitrary precision, the computed solutions will never be exact. In many applications such as geometric theorem proving, it is of interest to obtain solutions that can be exactly verified. In this paper, we present a numeric–symbolic method that exploits the efficiency of numerical techniques to obtain an approximate solution, which is then used as a starting point for the computation of an exact rational result. We show that under a strict feasibility assumption, an approximate solution of the semidefinite program is sufficient to obtain a rational decomposition, and quantify the relation between the numerical error versus the rounding tolerance needed. Furthermore, we present an implementation of our method for the computer algebra system Macaulay 2.

1. Introduction

An important question in computational mathematics is to decide whether a multivariate polynomial $p(x) \in \mathbb{R}[x]$ only takes nonnegative values for all $x \in \mathbb{R}^n$. Clearly, a sufficient condition for nonnegativity is that $p(x)$ can be written as a sum of squared polynomials, i.e., $p(x) = \sum_i p_i(x)^2$. The question of whether every nonnegative polynomial can be written as a sum of squares (SOS) dates back to Hilbert. The negative answer was given by Hilbert himself, and famous counterexamples were found later by Motzkin, Robinson, Choi, and Lam. We refer the reader to [37] for a survey on nonnegative polynomials, SOS, and their relations to Hilbert's 17th problem.

The algorithmic questions of deciding whether a sum of squares decomposition exists, and effectively finding one when it does, have been studied only relatively recently. A key structural element to these issues is provided by the "Gram matrix" method discussed in Section 2.2. This was presented in full form by Choi, Lam, and Reznick in [6], but there are clear traces of it in those authors' earlier works. Based on this characterization, perhaps the first work in the algebraic literature presenting an effective algorithm is Powers and Wörmann [30]. Their method is based on the Gram matrix technique, and relied on general decision theory algorithms such as quantifier elimination. For this reason, despite its high conceptual value, the methodology was not too applicable, except for very small problems.

A key development in this direction was the recognition that this problem has some very attractive properties from the geometric viewpoint, namely the convexity of the underlying feasible sets. These ideas were presented in [21,25], and in fact go back to Shor's pioneering work on global lower bounds on polynomials [39]. In fact, as we review in Section 2.2, these problems can be posed in a quite natural way in terms of the class of convex optimization problems known as

* Corresponding author.
E-mail addresses: peyrl@control.ee.ethz.ch (H. Peyrl), parrilo@mit.edu (P.A. Parrilo).
URLs: http://www.control.ee.ethz.ch/~hpeyrl (H. Peyrl), http://www.mit.edu/~parrilo (P.A. Parrilo).

0304-3975/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2008.09.025

semidefinite programs (SDPs). Since SDPs can be efficiently solved by interior point methods (cf. e.g., [42]), SOS problems have now become computationally tractable. After establishing the links between SDP, SOS and the Positivstellensatz, SOS techniques based on semidefinite programming gained widespread use in various fields, such as continuous and combinatorial optimization [25,18,22], control and dynamical systems [33,31] and quantum information theory [7] to cite a few.

In many applications, particularly those arising from problems in pure mathematics, it is often desirable to obtain exact algebraic solutions. Examples of this are the use of SOS methods for geometric theorem proving as in [27], or for establishing the validity of certain algebraic inequalities as in [16]. An interesting recent application is the work in [2], where SOS methods were used to prove new upper bounds on kissing numbers, a well-known problem in sphere packings. A common element in all these papers is the use of exact algebraic identities obtained from inspection of a numerically computed solution, as the basic ingredients in a rigorous proof.

In principle, semidefinite programming problems can be defined and solved purely algebraically. This can be done through real algebraic techniques such as the general decision methods as in the already mentioned [30], or slightly more efficient versions that partially exploit the convexity of the underlying sets (e.g., [1]). A possible alternative approach, relying on the solution of zero dimensional systems, is to focus on a specific element of the feasible set such as its analytic center (Section 2.2), and provide algebraic equations that uniquely define it. The considerable price to pay here is the algebraic degree of the corresponding solution. As has been recently shown by Nie et al. in [23], optimal solutions of relatively small semidefinite programs generically have minimum defining polynomials of astronomically high degree (an example of von Bothmer and Ranestad in [43] shows that for a generic semidefinite program with a matrix constraint with $n = 20, m = 105$, the degree of the optimal solution is $\approx 1.67 \times 10^{41}$). Despite the fact that an explicit algebraic representation of this solution is absolutely impossible to compute, it is a simple task using interior point methods to produce arbitrary precision numerical approximations to its solution.

While this and other dramatic examples suggest the superiority of numerical methods for these tasks, approximate numerical solutions computed via floating point (even with arbitrary precision) are often useless for certain applications such as the already mentioned ones. The reason is that they will never exactly satisfy the constraints, and thus do not serve as true certificates of the SOS property of the given polynomial, but only of nearby approximations.

There are solid theoretic reasons to justify the use of a mixed symbolic-numerical approach to the SOS problem. The aforementioned facts point to the necessity of an approach where the advantages of numerical computation are exploited for numerical efficiency, but at the same time the obtained solutions yield exact, unconditionally valid certificates of the existence of a SOS representation. This is exactly the objective of this paper, where we present a technique to use a numerical solution obtained from computationally efficient interior-point solvers as a starting point for the computation of an exact one. In this paper we develop a simple method based on this idea.

Our main contributions in this paper are the following:

- We show that under a strict feasibility assumption, it is sufficient to compute an *approximate* solution to the semidefinite program in order to obtain a rational sum of squares representation. In particular, we quantify the relation between the numerical error in the subspace and semidefinite constraints, versus the rounding tolerance, that guarantee that the rounded and projected solution will remain feasible. See Proposition 8 for the exact statement.
- We discuss several rounding procedures to convert the computed floating point solutions into rational numbers, and compare their relative advantages.
- We describe our implementation of these techniques through a Macaulay 2 package. This software formulates and numerically solves the required optimization problems, and uses this to produce certified rational solutions, guaranteed by construction to be correct.

The reminder of the paper is organized as follows: in Section 2 we will give a brief introduction to semidefinite programming and show the connection to the SOS problem. The basic ideas of our method are presented in Section 3. We conclude the paper with Section 4 in which we show how to use our Macaulay 2 software package.

1.1. Notation

1.1.1. Matrices

Let $\mathcal{S}^n \subset \mathbb{R}^{n\times n}$ denote the space of symmetric matrices with inner product between two elements $A, B \in \mathcal{S}^n$ denoted by $\langle A, B\rangle := \operatorname{trace} AB$. A matrix $A \in \mathcal{S}^n$ is called *positive semidefinite (PSD)* if $x^{\mathrm{T}}Ax \geq 0,\ \forall x \in \mathbb{R}^n$, and A is called *positive definite* if $x^{\mathrm{T}}Ax > 0,\ \forall x \in \mathbb{R}^n\backslash\{0\}$. Equivalently, A is positive semidefinite if and only if all eigenvalues are nonnegative, and A is positive definite if and only if all eigenvalues are strictly positive. Denoting the cone of PSD matrices by $\mathcal{S}^n_+$, we write $A \succeq B$ if $A - B \in \mathcal{S}^n_+$. Similarly, $A \succ B$ if $A - B \in \operatorname{int} \mathcal{S}^n_+$. Furthermore, we write $A \preceq B$ for $B \succeq A$ and $A \prec B$ for $B \succ A$ respectively.

1.1.2. Polynomials

Let $\mathbb{K}[x] := \mathbb{K}[x_1, \ldots, x_n]$ denote the ring of polynomials in n variables with coefficients in the field $\mathbb{K}$. Throughout this paper $\mathbb{K} = \mathbb{R}$ or $\mathbb{Q}$. We will use the multi-index $\alpha \in \mathbb{Z}^n_+$ to denote the monomial $x_1^{\alpha_1} \cdots x_n^{\alpha_n}$ by x^α. The degree of x^α is $|\alpha| = \sum_{i=1}^n \alpha_i$. A polynomial $f(x) \in \mathbb{K}[x]$ can be written as $\sum_{\alpha\in\mathbb{Z}^N_+} f_\alpha x^\alpha$, where f_α is the coefficient of monomial x^α.

The degree of $f(x)$ is determined by the largest $|\alpha|$ with $f_\alpha \neq 0$. Let S_k denote the set of exponents with degree at most k: $S_k = \{\alpha \in \mathbb{Z}_+^n \,|\, |\alpha| \leq k\}$. A polynomial $f(x) = \sum_{\alpha \in S_k} f_\alpha x^\alpha$ can be represented by its coefficient vector $f \in \mathbb{K}^{S_k}$ if $\deg f(x) \leq k$.

2. Sum of squares and semidefinite programming

In the following we will give a concise introduction to semidefinite programming and show its connection to the SOS problem.

2.1. Semidefinite programming background

In this section we will give a brief introduction to semidefinite programming and its basic underlying ideas. We refer the reader to [42,5] for a comprehensive treatment of the topic.

A semidefinite program is defined as the following *convex* optimization problem:

$$\begin{aligned} &\text{minimize} && \langle C, X\rangle \\ &\text{subject to} && \langle A_i, X\rangle = b_i, \quad i = 1, \ldots, m, \\ & && X \succeq 0, \end{aligned} \tag{P}$$

where $X \in \mathcal{S}^n$ is the decision variable and the matrices $C, A_i \in \mathcal{S}^n$, and $b \in \mathbb{R}^m$ are the problem data. The problem is convex since its objective function and the feasible region defined by the constraints are convex. A geometric interpretation is the minimization of a linear function over the intersection of the set of positive semidefinite matrices with an affine subspace. Problem (P) is called *strictly feasible* if there exists some $X \succ 0$ which satisfies the equality constraints in (P). The problem above has an associated *dual problem* being

$$\begin{aligned} &\text{maximize} && \langle b, y\rangle \\ &\text{subject to} && A(y) := C - \sum_{i=1}^m y_i A_i \succeq 0, \end{aligned} \tag{D}$$

with decision variable $y \in \mathbb{R}^m$. Problem (D) is called strictly feasible if there exists a y such that $A(y) \succ 0$. The value of any feasible solution of the dual problem provides a lower bound on any achievable value of the primal. This crucial property is referred to as *weak duality* and follows since for every feasible pair X and y

$$\langle C, X\rangle - \langle y, b\rangle = \langle C, X\rangle - \sum_{i=1}^m y_i \langle A_i, X\rangle = \left\langle C - \sum_{i=1}^m y_i A_i, X \right\rangle \geq 0,$$

where the last inequality follows from the fact that the inner product of two positive semidefinite matrices is nonnegative. The difference between the value of a primal feasible and a dual feasible solution is called the *duality gap*. Under certain constraint qualifications, e.g., the existence of a strictly feasible solution (Slater's condition), strong duality will hold and the optimal values of the primal and the dual problem will be equal, i.e., there is no duality gap. Furthermore, if both problems have nonempty interior, the optima will be attained by some X^* and y^* which satisfy the Karush–Kuhn–Tucker (KKT) optimality conditions stated in the following theorem:

Theorem 1 (*Cf. e.g., [42]*). *Assume that both the primal SDP (P) and the dual SDP (D) are strictly feasible. Then there exist optimal solutions X^* and y^* that achieve a zero duality gap, i.e., $\langle C, X^*\rangle = \langle b, y^*\rangle$. Furthermore, X^* and y^* are optimal if and only if they satisfy the optimality conditions*

$$\langle A_i, X\rangle = b_i, \quad i = 1, \ldots, m, \tag{1a}$$

$$A(y)X = 0, \tag{1b}$$

$$A(y) \succeq 0 \text{ and } X \succeq 0.$$

Eq. (1b) is called *complementary slackness* condition and is a direct consequence of strong duality and the existence of optimal solutions. Note that Eqs. (1a) and (1b) form a system of polynomial equations and at least in principle, one could solve them symbolically, for example, using Gröbner bases. However, as shown in [23], the degrees of the polynomials arising in the solution process when eliminating variables is usually enormous. On the other hand, numerical algorithms based on *interior point methods* can solve SDPs efficiently with *polynomial* worst-case complexity (cf. e.g., [42]). These methods generally use a *barrier function* to encode the feasible set in the objective function. For example, to represent the constraint $X \succeq 0$, a typical approach is to augment the objective function with the logarithm of the determinant of X and solve instances of the problem

$$\begin{aligned} &\text{minimize} && t\,\langle C, X\rangle - \log\det X \\ &\text{subject to} && \langle A_i, X\rangle = b_i, \quad i = 1, \ldots, m, \end{aligned} \tag{2}$$

where $t \geq 0$ is a real parameter. For $t = 0$ (a pure feasibility problem) the solution minimizes the barrier function and is called the *analytic center* of the feasible region. Since the barrier function tends to infinity along the boundary of the feasible set (i.e., when any of the eigenvalues of X gets close to zero), the returned solution will be well-centered in the interior of the feasible set. In contrast to simplex-like algorithms, the optimal solution X^* of (P) is approached iteratively along the so-called *central path* in the interior of the feasible set as t increases in each iteration step. For large values for t, the optimal value of (2) will get close to the optimum of (P).

Most SDP solvers are primal–dual methods which create sequences of primal feasible points $\{X^k\}$ and dual feasible points $\{y^k\}$, and use the duality gap as a stopping criteria. They can be interpreted as solving a relaxed system of KKT conditions of problem (2):

$$\begin{aligned} \langle A_i, X\rangle &= b_i, \quad i = 1, \ldots, m, \\ A(y)X &= (1/t)\, I, \\ A(y) \succeq 0 \text{ and } X &\succeq 0. \end{aligned}$$

For large values for t, the above system almost satisfies the optimality conditions (1). Hence the central path can be regarded as a continuous deformation of the KKT conditions.

Nowadays, there exist several efficient open source SDP solvers, e.g., SeDuMi [40], SDPA [8], CSDP [3], SDPT3 [41], just to mention a few of them.

2.2. SDP formulation of SOS problems

Although verifying nonnegativity of a polynomial $p(x)$ is in general a difficult problem, there exists a sufficient condition which is easier to solve: $p(x)$ is nonnegative if it can be decomposed into a sum of squared polynomials, i.e., $p(x) = \sum_i p_i(x)^2$. As already mentioned in the introduction, computing a sum of squares decomposition is equivalent to solving a semidefinite program. To pose the SOS problem in a semidefinite programming formulation, we express the given polynomial of degree $2d$ as a quadratic form

$$p(x) = z(x)^{\mathrm{T}} Q z(x), \tag{3}$$

where $z(x)$ is the vector of all monomials of degree less than or equal to d, i.e., $z_\alpha = x^\alpha,\ \alpha \in S_d$ and $Q \in \mathcal{S}^{S_d}$ is a symmetric matrix indexed by the exponent tuples in S_d. Since the components of $z(x)$ are *not algebraically independent*, Q is in general not unique. Expansion of the right-hand side of (3) and matching coefficients of the monomials yields a set of linear equations for the entries of Q. Hence the set of all matrices Q for which (3) holds is an *affine subspace* of the set of symmetric matrices. Let this affine subspace be denoted by $\mathcal{L}$:

$$\mathcal{L} := \{Q \in \mathcal{S}^{S_d} \mid p(x) = z(x)^{\mathrm{T}} Q z(x)\}. \tag{4}$$

If the intersection of $\mathcal{L}$ with the cone of PSD matrices is nonempty, $p(x)$ can be written as a sum of squares:

Theorem 2 (*[6, p. 106]*). *Let $p(x) \in \mathbb{R}[x]$ be a polynomial of degree less than or equal to $2d$. The following assertions are equivalent:*

(i) *$p(x)$ is a sum of squares.*
(ii) *There exists a positive semidefinite $Q \in \mathcal{L}$.*

Proof. Assume that (ii) is true. Then (i) follows from a factorization of Q: $p(x) = z^{\mathrm{T}} Q z = z^{\mathrm{T}} L^{\mathrm{T}} L z = \sum_i (L_i z)^2 = \sum_i p_i(x)^2$. Conversely, if $p(x)$ is a SOS, there exists a positive semidefinite matrix Q such that (3) holds. □

A positive semidefinite matrix $Q \in \mathcal{L}$ is called a *Gram matrix* of $p(x)$ (with respect to some expression $p(x) = \sum_i p_i(x)^2$). Since finding a positive semidefinite matrix in an affine space is a semidefinite program, computing an SOS decomposition is equivalent to solving a *feasibility SDP*. In the following we will discuss several issues arising in SOS problems such as exploitation of sparsity, different descriptions of $\mathcal{L}$, and parametrized sum of squares.

2.2.1. Sparseness

If $p(x)$ is a sparse polynomial, i.e., only a few coefficients p_α are different from zero, not all monomials in S_d might be needed in the monomial vector $z(x)$. Techniques exploiting sparseness can dramatically reduce the size of the underlying SDP. Sparseness can be exploited using the *Newton polytope* associated to the polynomial $p(x)$. This polytope is defined as the convex hull of the polynomial's exponent set: $C(p) := \mathrm{conv}(\{\alpha \mid p_\alpha \neq 0\})$. Reznick proved in [36] that only monomials with exponents contained in $\frac{1}{2}C(p)$ can appear in an SOS decomposition:

Theorem 3 (*[36]*). *If $p(x) = \sum_i p_i(x)^2$, then $C(p_i) \subseteq \frac{1}{2}C(p)$.*

2.2.2. Description of $\mathcal{L}$

The affine space $\mathcal{L}$ can be presented either through a set of basis matrices (*image* or *explicit* representation)

$$Q = G_0 + \sum_i y_i G_i, \tag{5}$$

where $G_0 \in \mathcal{L}$ and the G_i are a basis of the subspace $\mathcal{L} - G_0$, or by a system of defining equations (*kernel* or *implicit* representation):

$$\sum_{\substack{\beta,\gamma \in \frac{1}{2}C(p) \\ \beta+\gamma=\alpha}} Q_{\beta,\gamma} = p_\alpha, \quad \alpha \in C(p). \tag{6}$$

Depending on the dimension of $\mathcal{L}$, it is computationally advantageous to use either the kernel or the image representation. For polynomials of large degree d, the implicit form turns out to be more efficient. We refer the reader to [26] for a comprehensive complexity analysis of either representation. In any case, an SOS problem will be cast into either an SDP in primal form (P) or an SDP in dual form (D). In the following example we will derive the kernel and image representation for a simple polynomial.

Example 1. We consider the quartic form $p(x_1, x_2) = 2x_1^4 + 2x_1^3x_2 - x_1^2x_2^2 + 5x_2^4$. Note that since $p(x_1, x_2)$ is homogeneous of degree 4, it suffices to restrict the components of $z(x)$ to monomials of degree 2:

$$\begin{aligned} 2x_1^4 + 2x_1^3x_2 - x_1^2x_2^2 + 5x_2^4 &= \begin{bmatrix} x_1^2 \\ x_1x_2 \\ x_2^2 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{12} & q_{22} & q_{23} \\ q_{13} & q_{23} & q_{33} \end{bmatrix} \begin{bmatrix} x_1^2 \\ x_1x_2 \\ x_2^2 \end{bmatrix} \\ &= q_{11}x_1^4 + 2q_{12}x_1^3x_2 + (2q_{13} + q_{22})x_1^2x_2^2 + 2q_{23}x_1x_2^3 + q_{33}x_2^4. \end{aligned}$$

Matching coefficients yields the following linear defining equations for $\mathcal{L}$ (kernel representation):

$$q_{11} = 2, \quad 2q_{12} = 2, \quad (2q_{13} + q_{22}) = -1, \quad 2q_{23} = 0, \quad q_{33} = 5.$$

An image representation of the same affine subspace is given by the parametrization

$$Q(y) = \begin{bmatrix} 2 & 1 & -y \\ 1 & -1 + 2y & 0 \\ -y & 0 & 5 \end{bmatrix}.$$

A positive semidefinite matrix $Q \in \mathcal{L}$ can be obtained using semidefinite programming. A particular solution is

$$Q = \begin{bmatrix} 2 & 1 & -2 \\ 1 & 3 & 0 \\ -2 & 0 & 5 \end{bmatrix} = L^{\mathrm{T}}DL, \quad \text{where } L = \begin{bmatrix} -2/5 & 0 & 1 \\ 1/3 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \text{ and } D = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 13/15 \end{bmatrix}.$$

Hence $p(x_1, x_2)$ can be written as a sum of 3 squares:

$$p(x_1, x_2) = 5\left(-\frac{2}{5}x_1^2 + x_2^2\right)^2 + 3\left(\frac{1}{3}x_1^2 + x_1x_2\right)^2 + \frac{13}{15}(x_1^2)^2.$$

When the kernel representation is used for an SOS problem, the equality constraints (6) can be easily written in standard form (we use again exponent tuples for indexing the matrices):

$$\langle A^\alpha, Q \rangle = p_\alpha, \quad \text{where} \quad A^\alpha_{\beta,\gamma} = \begin{cases} 1 & \text{if } \beta + \gamma = \alpha, \\ 0 & \text{otherwise.} \end{cases}$$

As described, an SOS problem corresponds to an SDP *feasibility* problem, and any feasible matrix Q will yield a valid Gram matrix. To convert this into a problem with a unique solution, we can compute for instance the analytic center of the feasible set. Under the assumption of strict feasibility, it is easy to verify that the analytic center Q_0, i.e., the solution minimizing the barrier $-\log\det Q$, has to satisfy the following optimality conditions:

$$\begin{aligned} \langle A^\alpha, Q_0 \rangle &= p_\alpha, \qquad \alpha \in C(p), \\ \nabla(\log\det Q_0) = Q_0^{-1} &= \sum_\alpha A^\alpha y_\alpha. \end{aligned}$$

Note the remarkable multivariate Hankel structure of Q_0^{-1} which follows from the definition of the matrices A^α. Again, one could try to solve the optimality conditions symbolically, but the algebraic degree of the solution will in general be prohibitive already for problems of moderate size.

2.2.3. Parametrized SOS problems

A tremendous advantage of the SOS approach to polynomial nonnegativity is that the method can be easily extended to the problem of *finding* a sum of squares in a convex set of polynomials. To see this, consider the polynomial family $p(x, \lambda)$, where $p(x, \lambda)$ is affinely parametrized in λ, and λ is either free or belongs to a convex set described by semidefinite constraints. We can use semidefinite programming to efficiently search for parameters λ which render $p(x, \lambda)$ to be a sum of squares. The procedure is exactly as before: matching coefficients of the identity $p(x, \lambda) = z(x)^{\mathrm{T}} Q z(x)$ yields linear equations for Q and λ. Since both Q and λ are defined by semidefinite constraints, the problem is again an SDP. This fact is exploited in many applications, e.g., computing a lower bound on $p(x)$ or searching for a polynomial Lyapunov function for a system with a polynomial vector field.

Example 2. Let us revisit the polynomial from Example 1, but now assume that the coefficient of the monomial $x_1^3 x_2$ is a free parameter. An image representation of the corresponding affine subspace $\mathcal{L}$ is then given by

$$Q(\lambda, y) = \begin{bmatrix} 2 & \lambda & -y \\ \lambda & -1+2y & 0 \\ -y & 0 & 5 \end{bmatrix}.$$

The kernel representation of the same space is obtained by dropping the constraint $2q_{12} = 2$ in Example 1.

With software tools like SOSTOOLS [32] and YALMIP [20] there are two MATLAB packages available relieving the user from the task of casting a SOS problem into the corresponding SDP. However, since these are pure numerical methods, their answers will never yield exact results. Additionally, we would like to mention GloptiPoly 3 [12] and SparsePOP [44], two related MATLAB packages specialized on solving generalized problems of moments which are dual to SOS problems.

3. Computing rational SOS decompositions

We are interested in solving the following problem: given a polynomial with rational coefficients, i.e., $p(x) \in \mathbb{Q}[x]$, compute an exact SOS decomposition consisting only of squares of polynomials in $\mathbb{Q}[x]$. If such a decomposition is possible, we call $p(x)$ a *rational sum of squares*. To our knowledge, it is still an open question whether there always exists such a decomposition for every SOS polynomial. Landau showed in [17] that this is indeed possible for univariate polynomials that can be written as a sum of 8 squares in $\mathbb{Q}[x]$. Pourchet was able to improve Landau's estimate and proved in [29] that already 5 squares are sufficient (we refer the reader interested in the proof to Chapter 17 in [35]). Recently, it was shown by Hillar in [13] that sums of polynomial squares over totally real number fields are sum of squares in $\mathbb{Q}[x]$. Schweighofer presented an algorithmic proof in [38] showing that every univariate polynomial with coefficients in a subfield of $\mathbb{R}$ is a sum of squares of polynomials with coefficients in the same subfield. Unfortunately, the algorithm is restricted to univariate polynomials. The following proposition links rational SOS with the Gram matrix method:

Proposition 4. *The existence of a rational SOS decomposition, i.e., $p(x) = \sum_i p_i(x)^2$ where $p_i(x) \in \mathbb{Q}[x]$, is equivalent to the existence of a Gram matrix with rational entries.*

Proof. Assume that there exists a rational positive semidefinite Gram matrix Q for $p(x)$. Using diagonalization of quadratic forms over a field (cf. e.g., [34, Theorem 3.1.5]), the quadratic form $z^{\mathrm{T}} Q z$ can be written as a weighted sum of squares in $\mathbb{Q}[x]$:

$$p(x) = z^{\mathrm{T}} Q z = z^{\mathrm{T}} P^{\mathrm{T}} D P z = \sum_{i=1}^{k} d_i (Pz)_i^2 = \sum_{i=1}^{k} d_i p_i(z(x))^2,$$

where P and D are rational matrices, $D := \operatorname{diag}(d_1, \ldots, d_k)$ being diagonal. The rational weights d_i are nonnegative because Q was assumed to be positive semidefinite. Observe that $d_i = a_i/b_i = a_i b_i / b_i^2$. Hence $p(x)$ can be written as a sum of at most $a_1 b_1 + \cdots + a_k b_k$ squares in $\mathbb{Q}[x]$. A slightly different argument, using Lagrange's four-square theorem, yields an upper bound of $4k$ squares.

Conversely, if $p(x)$ can be written as a sum of squared polynomials in $\mathbb{Q}[x]$, there exists a positive semidefinite Gram matrix Q with rational entries. □

The basic idea of our approach for computing rational sums of squares is to take advantage of interior point solvers' computational efficiency: we compute an approximate numerical solution and in a second step we round the numerical solution to an exact rational one. We have the following standing assumption:

Assumption 1. There exists a strictly feasible Gram matrix for $p(x)$.

A crucial factor in obtaining an exact SOS decomposition with our method is the strict feasibility of the underlying SDP, i.e., the existence of a Gram matrix Q with full rank. Consequently, the method could fail in general for sum of squares that are not strictly positive: if there is an x^* such that $p(x^*) = 0$, it follows from the identity $p(x^*) = z(x^*)^{\mathrm{T}} Q z(x^*)$ that the monomial vector $z(x^*)$ is in the kernel of Q. Hence Q cannot be positive definite.

If the real zeros of $p(x)$ are known, then it may be possible to remove them using the linear constraint $Qz(x^*) = 0$, to obtain a smaller semidefinite program that will likely be strictly feasible. Both this procedure, and the already mentioned

Theorem 3, can be understood in terms of a *facial reduction* procedure [4], where the full-dimensional SOS cone is replaced by a smaller (but not necessarily minimal) face containing the given polynomial. It would be of interest to extend the Newton polytope theory to a fully general facial reduction scheme.

As already mentioned, a plain SOS problem is just a feasibility SDP without any objective function. Hence an interior point solver that minimizes the log-barrier function will return a solution which is "well-centered" in the cone of PSD matrices. Under the strict feasibility assumption, this analytic center will be a positive definite matrix, since the optimization is maximizing the determinant, and there exists at least one solution with strictly positive determinant. Thus chances are good that a rational approximation of the numeric solution is positive semidefinite as well. Consequently, we have to verify in a last step that the Gram matrix corresponding to the rational solution is indeed positive semidefinite.

In the following we will briefly discuss different methods to symbolically verify positive semidefiniteness of a rational matrix:

Characteristic polynomial. The following theorem links positive semidefiniteness of a matrix with the signs of the coefficients of its characteristic polynomial.

Theorem 5 (*Cf. e.g., [14]*). *An $n \times n$ symmetric matrix Q is positive semidefinite if and only if all the coefficients of its characteristic polynomial $p(\lambda) = \det(\lambda I - Q) = \lambda^n + p_{n-1}\lambda^{n-1} + \cdots + p_0$ alternate in sign, i.e., they satisfy $p_i(-1)^{n-i} \geq 0$.*

If Q has only rational entries, the coefficients of the characteristic polynomial will be rational numbers that can be computed exactly. Checking their signs according to the theorem yields an unconditionally valid test for positive semidefiniteness.

Matrix diagonalization. Another way to verify positive semidefiniteness of a matrix is to diagonalize it as in the proof of Proposition 4. A particular diagonalization is obtained via the LDL^{T} decomposition, a variant of the LU decomposition appropriate for symmetric matrices:

Theorem 6 (*Cf. e.g., [9, pp. 134ff.]*). *Let Q be a symmetric positive semidefinite $n \times n$ matrix. Then there exist a diagonal matrix $D = \operatorname{diag}(d_1, \ldots, d_n)$, a lower triangular matrix L with unit diagonal, and a nonsingular permutation matrix P such that $P^{\mathrm{T}}QP = LDL^{\mathrm{T}}$.*

Since the LDL^{T} factorization of a matrix only involves basic arithmetic computations, the decomposition can be computed exactly in the field of rational numbers. The matrix Q is positive semidefinite if and only if all the diagonal elements d_i are nonnegative.

Alternatively, an essentially similar matrix diagonalization may also be obtained from a Gram–Schmidt orthogonalization process (cf. e.g., [34, Theorem 3.1.5]).

For our Macaulay 2 SOS package we decided to use the LDL^{T} decomposition, since it turned out to be significantly faster than the computation of the characteristic polynomial. As is to be expected, solving the SDP is evidently the bottleneck in our algorithm.

In the approximation step we have to distinguish two cases depending on whether the SOS problem is posed as an SDP in primal form (P) or dual form (D):

3.1. Kernel representation

If the SOS problem is posed as an SDP in primal form (P), the numerical solution Q will not exactly fulfill identity (3). For an exact representation of the original polynomial $p(x)$, we have to find a rational approximation of Q which satisfies the equality constraints (6). The simplest procedure is to compute a rational approximation $\tilde{Q}$, for example by using continued fractions (cf. e.g., [15]) which represent a real number r with an expression as follows

$$r = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ddots}}},$$

where the a_i are integer numbers. Continued fractions are a sensible choice as truncating the series yields *best rational approximations*: a best rational approximation of a real number r is a rational number $\frac{a}{b}$, $b > 0$ such that there is no rational number with smaller denominator which is closer to r.

To satisfy (6), the rational approximation $\tilde{Q}$ is projected onto the subspace $\mathcal{L}$. Since the affine space is defined by rational data, i.e., the coefficients of $p(x)$, an orthogonal projection Π onto $\mathcal{L}$ will yield a rational matrix $\Pi(\tilde{Q})$ satisfying (3). The special structure of $\mathcal{L}$ results in a very simple projection formula:

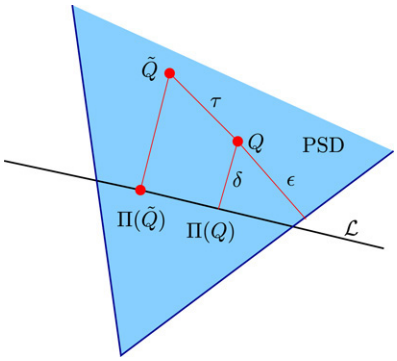

Fig. 1. Projection of a rounded solution. The orthogonal projections of the matrices Q and $\tilde{Q}$ are denoted by $\Pi(Q)$ and $\Pi(\tilde{Q})$ respectively. The shaded cone PSD represents the cone of positive semidefinite matrices.

Proposition 7. *The orthogonal projection Π of a symmetric matrix Q onto the space $\mathcal{L}$ defined in (4) is given by*

$$\Pi(Q)_{\alpha,\beta} = Q_{\alpha,\beta} - \frac{1}{n(\alpha+\beta)} \underbrace{\left[\sum_{\alpha'+\beta'=\alpha+\beta} Q_{\alpha',\beta'} - p_{\alpha+\beta} \right]}_{e_{\alpha+\beta}} \quad \text{for all } \alpha, \beta \in \frac{1}{2}C(p),$$

where $n(\alpha+\beta)$ denotes the number of pairs (α', β') such that $\alpha'+\beta' = \alpha+\beta$, i.e.,

$$n(\alpha+\beta) := |\{(\alpha', \beta') \mid \alpha'+\beta' = \alpha+\beta,\ \alpha', \beta' \in \frac{1}{2}C(p)\}|.$$

The proof for this proposition can be found in Appendix A of this paper. Note that the expression in the square brackets, $e_{\alpha+\beta}$, is the error in the coefficient of monomial $x^{\alpha+\beta}$ of the polynomial $z(x)^{\mathrm{T}}Qz(x)$. Thus the orthogonal projection is obtained by just subtracting the weighted errors of the coefficients from the Gram matrix. Furthermore, note that since only basic arithmetic operations are used, $\Pi(Q)$ will be a rational matrix if Q and p are rational.

We conclude with an estimate of the rounding tolerance needed. Assuming strict feasibility of the numerical solution, we quantify how "far away" it is from the boundary of the PSD cone and the affine subspace. In other words, there are an $\epsilon > 0$ and a $\delta \geq 0$ such that $Q \succeq \epsilon I$ and $d(Q, \Pi(Q)) \leq \delta$, where $d(\cdot,\cdot)$ denotes the Euclidean distance between two matrices. Note that the condition $Q \succeq \epsilon I$ is equivalent to the minimum eigenvalue of Q being greater than or equal to ϵ. The matrix Q is approximated by a rational matrix $\tilde{Q}$ such that $d(Q, \tilde{Q}) \leq \tau$. Fig. 1 depicts the whole situation.

Proposition 8. *Let ϵ, δ, and τ be defined as above. Assume $\tau^2 + \delta^2 \leq \epsilon^2$. Then, the orthogonal projection of the rounded matrix $\tilde{Q}$ on the affine subspace $\mathcal{L}$ is positive semidefinite, and thus it is a valid SOS decomposition.*

Proof. Since the projection of $\tilde{Q}$ onto $\mathcal{L}$ is orthogonal, $Q - \Pi(Q)$ and $\Pi(\tilde{Q}) - \Pi(Q)$ are orthogonal. Therefore, by Pythagoras' theorem:

$$d(\Pi(\tilde{Q}), Q)^2 = d(Q, \Pi(Q))^2 + d(\Pi(\tilde{Q}), \Pi(Q))^2.$$

Clearly, $d(\Pi(\tilde{Q}), \Pi(Q)) \leq d(\tilde{Q}, Q) \leq \tau$, so

$$d(\Pi(\tilde{Q}), Q) \leq \sqrt{\delta^2 + \tau^2}.$$

Let $\lambda_i(\cdot)$ denote the i-th largest eigenvalue of a symmetric matrix, and let $\bar{\sigma}(\cdot) = \max_i |\lambda_i(\cdot)|$. Note that because $|\lambda_i(\cdot)| \leq \sqrt{\sum_i \lambda_i^2(\cdot)} = d(\cdot, 0)$, it holds that $|\lambda_i(\Pi(\tilde{Q}) - Q)| \leq d(\Pi(\tilde{Q}) - Q, 0) = d(\Pi(\tilde{Q}), Q) \leq \sqrt{\delta^2 + \tau^2}$. To show that $\Pi(\tilde{Q})$ is positive semidefinite, we rewrite $\Pi(\tilde{Q})$ as follows:

$$\Pi(\tilde{Q}) = Q + \Big(\Pi(\tilde{Q}) - Q\Big) \succeq \Big(\epsilon - \bar{\sigma}(\Pi(\tilde{Q}) - Q)\Big) I \succeq (\epsilon - \sqrt{\delta^2 + \tau^2}) I \succeq 0,$$

where the last inequality follows from the assumption that $\tau^2 + \delta^2 \leq \epsilon^2$. □

Hence if the SDP is strictly feasible, and $\delta^2 < \epsilon^2$, it is in principle always possible to compute a valid rational solution by using sufficiently many digits for the approximated solution. The allowed rounding tolerance τ depends on the minimum eigenvalue of the positive definite matrix Q and its distance from the affine space $\mathcal{L}$. Under the strict feasibility assumption (Assumption 1), there always exists a solution with δ sufficiently small such that the inequality above can be fulfilled (in particular, we can just take $\delta = 0$). From a practical point of view, however, it could conceivably happen that a fixed-precision floating-point solver returns a solution where $\delta \geq \epsilon$. This is not too serious an issue, for two reasons. A simple reformulation described in the next section, using the image representation and the dual form of SDP, will guarantee

Table 1
Functions provided by the SOS package

Function	Description
`(g,d) = getSOS(f)`	Computes a rational SOS decomposition of the input polynomial `f` and returns a list of rational weights `d` and a list of polynomials `g` in $\mathbb{Q}[x]$ such that $f(x) = \sum_i d_i g_i(x)^2$. An error message is displayed if no valid SOS decomposition is found.
`(ok,Q,z) = findSOS(f)`	Has the same functionality as `getSOS` but returns the corresponding Gram matrix `Q` and a list of monomials `z` such that $f(x) = z(x)^{\mathrm{T}} Q z(x)$. `ok` is a Boolean variable which is true when the decomposition algorithm was successful and false otherwise.
`f = sumSOS(g,d)`	For checks: given a list of polynomials `g` and a list of weights `d`, `sumSOS` computes the expression $\sum_i d_i g_i(x)^2$.

that the computed solution exactly satisfies the constraints (i.e., $\delta = 0$). Alternatively, many available SDP solvers such as SDPT3 or SDPA allow the user to specify the accepted error in the equality constraints for an solution. Nevertheless, we think that arbitrary precision floating point SDP solvers could be an important step to overcome this potential difficulty for more complicated problems. Furthermore, please note that Proposition 8 only provides a sufficient condition for the rounding tolerance; in many examples, even not strictly feasible ones, we were able to obtain valid rational solutions using much coarser roundings.

Alternatively to the approximation with continued fractions and its subsequent projection, a rational solution may be obtained using the LLL lattice basis reduction developed by Lenstra, Lenstra, and Lovász. Already in their seminal paper [19] Lenstra et al. presented the simultaneous approximation of a vector of rational numbers as a possible application. It is not difficult to extend their algorithm for the simultaneous approximation in a subspace. We refer the reader to Appendix B for the details.

3.2. Image representation

If the SOS problem is formulated as an SDP in dual form, the polynomial identity (3) holds for any value of the decision variables y since the base matrices G_i are exact. Thus it suffices to approximate the numerical solution y by a vector of rational numbers $\tilde{y}$. Again, a sensible choice for the rounding procedure are truncated continued fractions. While for a given precision they will yield the best rational approximation of y, the denominators of the $\tilde{y}_i$ will in general be different. Similar as in the case of the kernel representation, the LLL algorithm can be used to obtain a rational approximation with common denominator. We refer the reader to [19] and Appendix B for the details.

We have the following estimate for the rounding tolerance that guarantees a valid solution:

Proposition 9. *Let the subspace $\mathcal{L}$ be described by a set of basis matrices as in (5). Assume that $Q \succeq \epsilon I \succ 0$, and let the rational approximation $\tilde{y}$ be such that $|y_i - \tilde{y}_i| \leq \tau$, where $\tau \leq \frac{\epsilon}{\sum_i \bar{\sigma}(G_i)}$ and $\bar{\sigma}(\cdot) = \max_i |\lambda_i(\cdot)|$. Then the rational approximation $\tilde{y}$ will yield an exact SOS decomposition.*

Proof. Since identity (3) is fulfilled for any $\tilde{y}$, we only have to verify that $Q(\tilde{y})$ is positive semidefinite. Under the assumptions of the proposition it is easy to see that this is indeed the case:

$$Q(\tilde{y}) = G_0 + \sum_i \tilde{y}_i G_i \succeq G_0 + \sum_i (y_i G_i - \tau \bar{\sigma}(G_i) I) \succeq \left(\epsilon - \tau \sum_i \bar{\sigma}(G_i)\right) I \succeq 0. \quad \square$$

4. Macaulay 2 SOS package

We used the computer algebra system Macaulay 2 [10] to implement a SOS package based on the ideas presented in this paper. The package together with packages for solving SDPs and computing LDL^{T} decompositions are available for download at [28]. To solve the SDP we use a simple, pure dual interior point method based on damped Newton steps as described in [5]. The algorithm for the LDL^{T} decomposition is taken from [9]. Similar ideas to the ones presented in this paper have recently been implemented by Harrison in the open source theorem prover HOL Light [11].

Table 1 summarizes the functions provided by the `SOS` package. The main function is `getSOS` which tries to compute a rational SOS decomposition for a given polynomial. In the following example we demonstrate how to use the `getSOS` for computing an SOS decomposition of a polynomial of degree 4 with 4 variables.

Example 3. Consider the polynomial

$$p(x, y, z, w) = 2x^4 + x^2y^2 + y^4 - 4x^2z - 4xyz - 2y^2w + y^2 - 2yz + 8z^2 - 2zw + 2w^2.$$

To begin with, we have to load the `SOS` package and define $p(x, y, z, w)$:

```
i1 : loadPackage "SOS";
i2 : P = QQ[x,y,z,w];
i3 : p = 2*x^4 + x^2*y^2 + y^4 - 4*x^2*z - 4*x*y*z - 2*y^2*w + y^2 - 2*y*z + 8*z^2 - 2*z*w + 2*w^2;
```

Table 2
Parameter optimization with `getSOS`

`(g,t [,pval]) = getSOS (f [,p [,ofun [,pmin, pmax]]] [,rndTol=>n])`

Argument:	Description:
`f`	Input polynomial
`p`	List of affine parameters
`pmin/pmax`	List of lower/upper bounds for the parameters
`ofun`	Linear objective function of the parameters, `ofun` is minimized
`rndTol=>n`	Set required precision to `n` binary digits ($0 \leq n \leq 52$)
`g`	List of polynomials
`d`	List of rational weights
`pval`	List of rational parameters

Optional arguments are given in square brackets.

If successful, the function `getSOS` returns a weighted SOS representation such that $p(x, y, z, w) = \sum_i d_i g_i(x, y, z, w)^2$. Otherwise an error message is displayed.

```
i4 : (g,d) = getSOS p

... omitted output ...
```

```
          1  2   1       1         1       2  2    2        8  2    1
o8 = ({- -*x  - -*x*y - -*y + z - -*w, - --*x  - --*x*y - --*y  - --*y + w,
          4      4       8         8      15       15       15      15

     ------------------------------------------------------------------------
      2    4        4  2    2          18  2   20       81  2       2
     x  - --*x*y - --*y  - --*y, x*y - --*y  - --*y, - ---*y  + y, y },
          11       11      11         59      59       205
     ------------------------------------------------------------------------
         15  22  59  41   66
     {8, --, --, --, --, ----})
          8  15  55  59  1025
```

Hence $p(x, y, z, w)$ may be written as

$$p(x,y,z,w) = 8\left(-\frac{1}{4}x^2 - \frac{1}{4}xy - \frac{1}{8}y + z - \frac{1}{8}w\right)^2 + \frac{15}{8}\left(-\frac{2}{15}x^2 - \frac{2}{15}xy - \frac{8}{15}y^2 - \frac{1}{15}y + w\right)^2$$
$$+ \frac{22}{15}\left(x^2 - \frac{4}{11}xy - \frac{4}{11}y^2 - \frac{2}{11}y\right)^2 + \frac{59}{55}\left(xy - \frac{18}{59}y^2 - \frac{20}{59}y\right)^2 + \frac{41}{59}\left(-\frac{81}{205}y^2 + y\right)^2 + \frac{66}{1025}y^4.$$

Correctness of the obtained decomposition may be verified with the function `sumSOS` which expands a weighted sum of squares decomposition:

```
i5 : sumSOS (g,d) - p

o5 = 0

o5 : P
```

As discussed in Section 2.2, one of the strengths of the SDP approach towards polynomial nonnegativity is that one can search for (and optimize over) coefficients which render a polynomial to be a SOS. The `SOS` package contains rudimentary support for handling linearly parametrized polynomials. Please note that this functionality is still at an early stage. Table 2 shows the syntax of the command `getSOS` when used with parametrized polynomials. In the subsequent example we will show how to compute a verified lower bound for a given polynomial.

Example 4. In [24] the value -2.11291382 was obtained as a numerical lower bound for the polynomial

$$f(x, y, z) = x^4 + y^4 + z^4 - 4xyz + x + y + z.$$

We can compute an rational approximation of this lower bound with the function `getSOS`. We start by defining the polynomial with an additional variable t for the lower bound:

```
i7 : P = QQ[x,y,z,t];
i8 : p = x^4 + y^4 + z^4 - 4*x*y*z + x+y+z - t;
```

To compute a rational lower bound, we want to find the biggest t such that $p(x, y, z, t)$ can still be written as a sum of squares (we restrict t to be in the interval $[-10, 0]$):

```
i9 : (g,d,v) = getSOS (f,{t},-t,{-10},{0});

... omitted output ...

i10 : v

        35448817
o10 = {- --------}
        16777216
```

Hence a certified lower bound for $f(x, y, z)$ is $-\frac{35448817}{16777216}$ which is roughly -2.112914145.

5. Conclusion and outlook

In this paper we presented a method for computing rational SOS decompositions which serve as exact certificates of the SOS property. The proposed method is a symbolic–numeric approach that uses efficient interior point solvers to obtain a numerical approximate solution which is then rounded to an exact rational solution. We showed that under a strict feasibility assumption, an approximate solution of the underlying semidefinite program is sufficient to obtain an exact SOS representation. We discussed several rounding procedures to convert the floating point solutions into rational ones. Furthermore, we described an implementation of the proposed method through a Macaulay 2 package. An extended version of this package which is able to handle several SOS constraints at once and has interfaces to external SDP solvers is currently under development. Future research could address the case of non strictly feasible SOS problems.

Acknowledgements

The authors would like to thank the referees for their valuable feedback and useful comments.

Appendix A. Proof of Proposition 7

Proof. (i) The matrix $\Pi(Q)$ is an element of $\mathcal{L}$ because for all $\gamma \in C(p)$

$$\sum_{\alpha+\beta=\gamma} \Pi(Q)_{\alpha,\beta} = \sum_{\alpha+\beta=\gamma} Q_{\alpha,\beta} - \underbrace{\sum_{\alpha+\beta=\gamma} \frac{1}{n(\gamma)}}_{=1} \left[\sum_{\alpha'+\beta'=\gamma} Q_{\alpha',\beta'} - p_\gamma \right] = p_\gamma.$$

(ii) $Q - \Pi(Q)$ is orthogonal to $\mathcal{L}$, i.e., the inner product between $Q - \Pi(Q)$ and the kernel of the linear map defining $\mathcal{L}$ is zero. Let Δ be an element of the kernel of this linear map, i.e., $\sum_{\alpha+\beta=\gamma} \Delta_{\alpha,\beta} = 0$ for all $\gamma \in C(p)$. Then

$$\begin{aligned} \langle Q - \Pi(Q),\ \Delta \rangle &= \sum_{\alpha,\beta \in \frac{1}{2}C(p)} \left(Q_{\alpha,\beta} - \Pi(Q)_{\alpha,\beta} \right) \Delta_{\alpha,\beta} = \sum_{\alpha,\beta \in \frac{1}{2}C(p)} \frac{e_{\alpha+\beta}}{n(\alpha+\beta)} \Delta_{\alpha,\beta} \\ &= \sum_{\gamma \in C(p)} \sum_{\alpha+\beta=\gamma} \frac{e_\gamma}{n(\gamma)} \Delta_{\alpha,\beta} = \sum_{\gamma \in C(p)} \frac{e_\gamma}{n(\gamma)} \sum_{\alpha+\beta=\gamma} \Delta_{\alpha,\beta} = 0, \end{aligned}$$

where the last equation follows from the assumption that Δ is contained in the kernel of the map $Q \mapsto p$. □

Appendix B. Simultaneous approximation using the LLL algorithm

The LLL algorithm by Lenstra et al. computes a set of *short*, *nearly orthogonal* basis vectors for a given lattice. In [19] the authors showed how the basis reduction algorithm can be used for the simultaneous approximation of a vector. To compute a rational approximation $\tilde{y}$ with common denominator of a vector $y \in \mathbb{R}^m$, consider the lattice spanned by the column vectors of the matrix

$$L = \begin{bmatrix} N & 0 & \cdots & 0 & -\lceil Ny_1 \rceil \\ 0 & N & \dots & 0 & -\lceil Ny_2 \rceil \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & N & -\lceil Ny_m \rceil \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix},$$

where N is a large integer number. The LLL algorithm will compute a reduced basis spanning the same lattice as the columns of L. Hence for every reduced basis vector $\hat{g}$, there exist integers $a \in \mathbb{Z}^m$ and $b \in \mathbb{Z}$ such that

$$\hat{g} = N \begin{bmatrix} a_1 \\ \vdots \\ a_m \\ 0 \end{bmatrix} - b \begin{bmatrix} \lceil Ny_1 \rceil \\ \vdots \\ \lceil Ny_m \rceil \\ -1 \end{bmatrix}.$$

Since the reduced basis is short, we expect $\hat{g}_i/N = a_i - b\frac{\lceil Ny_i \rceil}{N}$, $i = 1, \ldots, m$, to be a small. In other words, $\frac{a_i}{b}$ is a good approximation of $\frac{\lceil Ny_i \rceil}{N}$. For bounds on the quality of the approximation we refer the reader to [19].

When the SOS problem is formulated in the kernel representation, the LLL algorithm can also be used to find a rational approximate solution directly in the affine space $\mathcal{L}$. For the approximation we assume that a basis of $\mathcal{L}$ is available and consider $\mathcal{L}$ in the vectorized form

$$q = g_0 + \sum_{i=1}^{m} y_i g_i, \quad y_i \in \mathbb{R}^m,$$

where the $g_i := \mathrm{vec}(G_i)$ denote the vectors containing the columns of the basis matrices G_i stacked below each other. Without loss of generality we will assume that the polynomial $p(x)$ has only integer coefficients (otherwise scale $p(x)$ appropriately) and hence the g_i are integer vectors. Let q denote the vectorized floating point solution Q and consider the lattice spanned by the column vectors of

$$L = \begin{bmatrix} Ng_1 & Ng_2 & \cdots & Ng_m & Ng_0 - \lceil Nq \rceil \\ 0 & 0 & \ldots & 0 & 1 \end{bmatrix},$$

where N is again a large integer. For a reduced basis vector $\hat{g}$, there exist integers $a \in \mathbb{Z}^m$ and $b \in \mathbb{Z}$ such that

$$\hat{g} = \begin{bmatrix} \sum_{i=1}^{m} a_i N g_i + b(Ng_0 - \lceil Nq \rceil) \\ b \end{bmatrix}.$$

Since the reduced basis is short, we expect $\sum_{i=1}^{m} a_i g_i + b g_0 - b\frac{\lceil Nq \rceil}{N}$ to be small. In other words, $\sum_{i=1}^{m} \frac{a_i}{b} g_i + g_0$ is a good approximation of $\frac{\lceil Nq \rceil}{N}$.

References

[1] H. Anai, P.A. Parrilo, Convex quantifier elimination for semidefinite programming, in: Proceedings of the International Workshop on Computer Algebra in Scientific Computing, CASC 2003, 2003.
[2] C. Bachoc, F. Vallentin, New upper bounds for kissing numbers from semidefinite programming. arXiv:math/0608426v4[math.MG], 2006.
[3] B. Borchers, A C library for semidefinite programming, Optim. Meth. Software 11 (1) (1999) 613–623.
[4] J.M. Borwein, H. Wolkowicz, Facial reduction for a cone-convex programming problem, J. Aust. Math. Soc. Ser. A 30 (3) (1980–81) 369–380.
[5] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.
[6] M.D. Choi, T.Y. Lam, B. Reznick, Sums of squares of real polynomials, in: K-Theory and Algebraic Geometry: Connections with Quadratic Forms and Division Algebras, in: Proc. Sympos. Pure Math., vol. 58, Amer. Math. Soc., Providence, RI, 1995, pp. 103–126.
[7] A.C. Doherty, P.A. Parrilo, F.M. Spedalieri, Distinguishing separable and entangled states, Phys. Rev. Lett. 88 (18) (2002) 187904.
[8] K. Fujisawa, M. Kojima, K. Nakata, M. Yamashita, SDPA (SemiDefinite Programming Algorithm). Available from http://grid.r.dendai.ac.jp/sdpa.
[9] G. Golub, C. van Loan, Matrix Computations, 2nd ed., in: Johns Hopkins Series in the Mathematical Sciences, The Johns Hopkins University Press, Baltimore, Maryland, 1989.
[10] D.R. Grayson, M.E. Stillman, Macaulay 2, a software system for research in algebraic geometry. Available from http://www.math.uiuc.edu/Macaulay2.
[11] J. Harrison, Verifying nonlinear real formulas via sums of squares, in: K. Schneider, J. Brandt (Eds.), Proceedings of the 20th International Conference on Theorem Proving in Higher Order Logics, TPHOLs 2007, in: Lect. Notes Comput. Sci., vol. 4732, Springer-Verlag, Kaiserslautern, Germany, 2007.
[12] D. Henrion, J.B. Lasserre, J. Löfberg, Gloptipoly 3. Available from http://www.laas.fr/~henrion/software/gloptipoly3, 2007.
[13] C.J. Hillar, Sums of squares over totally real fields are rational sums of squares. arXiv:0704.2824v2[math.AC], 2007.
[14] R.A. Horn, C.R. Johnsob, Matrix Analysis, Cambridge University Press, Cambridge, 1995.
[15] W. Jones, W. Thron, Continued Fractions: Analytic Theory and Applications, in: Encyclopedia of Mathematics and its Applications, vol. 11, Addison-Wesley Publishing Company, 1980.
[16] I. Klep, M. Schweighofer, Sums of hermitian squares and the BMV conjecture. Available from http://perso.univ-rennes1.fr/markus.schweighofer/publications/bmv.pdf, 2007.
[17] E. Landau, Über die Darstellung definiter Funktionen als Summe von Quadraten, Math. Ann. 62 (1906) 290–329.
[18] J.B. Lasserre, Global optimization with polynomials and the problem of moments, SIAM J. Optim. 11 (2001) 796–817.
[19] A.K. Lenstra, H.W. Lenstra, L. Lovász, Factoring polynomials with rational coefficients, Math. Ann. 261 (4) (1982) 515–534.
[20] J. Löfberg, YALMIP: A toolbox for modeling and optimization in MATLAB, in: Proceedings of the CACSD Conference, Taipei, Taiwan, 2004. Available from http://control.ee.ethz.ch/~joloef/yalmip.php.
[21] Y. Nesterov, Squared functional systems and optimization problems, in: J. Frenk, C. Roos, T. Terlaky, S. Zhang (Eds.), High Performance Optimization, Kluwer Academic Publishers, 2000, pp. 405–440.
[22] J. Nie, Sum of squares method for sensor network localization, Computational Optimization and Applications (in press) published online at http://www.springerlink.com/content/xj58880l17615817/, 2007.
[23] J. Nie, K. Ranestad, B. Sturmfels, The algebraic degree of semidefinite programming. arXiv:math/0611562v2[math.OC], 2006.
[24] P. Parrilo, B. Sturmfels, Minimizing polynomial functions, in: Algorithmic and Quantitative Real Algebraic Geometry, in: DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 60, Amer. Math. Soc, 2003, pp. 83–99.
[25] P.A. Parrilo, Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization, Ph.D. Thesis, California Institute of Technology, 2000.

[26] P.A. Parrilo, Semidefinite programming relaxations for semialgebraic problems, Math. Program. 96 (2) (2003) 293–320.
[27] P.A. Parrilo, R. Peretz, A geometric inequality for circle packings, Discrete Comput. Geom. 31 (3) (2004) 357–367.
[28] H. Peyrl, P.A. Parrilo, SOS.m2, a sum of squares package for Macaulay 2. Available from http://www.control.ee.ethz.ch/~hpeyrl/index.php, 2007.
[29] Y. Pourchet, Sur la représentation en somme de carrés des polynômes à une indéterminée sur un corps de nombres algébriques, Acta Arithm. 19 (1971) 89–104.
[30] V. Powers, T. Wörmann, An algorithm for sums of squares of real polynomials, J. Pure Appl. Algebra 127 (1998) 99–104.
[31] S. Prajna, A. Jadbabaie, G.J. Pappas, A framework for worst-case and stochastic safety verification using barrier certificates, IEEE Trans. Automat. Control 52 (8) (2007) 1415–1428.
[32] S. Prajna, A. Papachristodoulou, P. Seiler, P.A. Parrilo, SOSTOOLS: Sum of squares optimization toolbox for MATLAB. Available from http://www.cds.caltech.edu/sostools and http://www.mit.edu/~parrilo/sostools, 2004.
[33] S. Prajna, P.A. Parrilo, A. Rantzer, Nonlinear control synthesis by convex optimization, IEEE Trans. Automat. Contr. 49 (2) (2004) 310–314.
[34] A. Prestel, C.N. Delzell, Positive Polynomials, Springer, 2001.
[35] A.R. Rajwade, Squares, in: London Mathematical Society Lecture Notes, vol. 171, 1993.
[36] B. Reznick, Extremal psd forms with few terms, Duke Math. J. 45 (1978) 363–374.
[37] B. Reznick, Some concrete aspects of Hilbert's 17th problem, in: Real Algebraic Geometry and Ordered Structures, in: Contemporary Mathematics, vol. 253, Amer. Math. Soc, Providence, RI, 2000, pp. 251–272.
[38] M. Schweighofer, Algorithmische Beweise für Nichtnegativ- und Positivstellensätze, Master's Thesis, Universität Passau, 1999.
[39] N.Z. Shor, Class of global minimum bounds of polynomial functions, Cybernetics 23 (6) (1987) 731–734.
[40] J.F. Sturm, Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. Interior point methods, Optim. Meth. Software 11–12 (1–4) (1999) 625–653.
[41] R.H. Tütüncü, K.C. Toh, M.J. Todd, Solving semidefinite-quadratic-linear programs using SDPT3, Math. Program. Ser. B 95 (2003) 189–217.
[42] L. Vandenberghe, S. Boyd, Semidefinite programming, SIAM Rev. 38 (1) (1996) 49–95.
[43] H.-C. von Bothmer, K. Ranestad, A general formula for the algebraic degree in semidefinite programming. math.AG/0701877, 2007.
[44] H. Waki, S. Kim, M. Kojima, M. Muramatsu, Sums of squares and semidefinite programming relaxation for polynomial optimization problems with structured sparsity, SIAM J. Optim. 17 (1) (2006) 218–242. Available from http://www.is.titech.ac.jp/~kojima/SparsePOP.