Exact certification in global polynomial optimization via sums-of-squares of rational functions with rational coefficients*

Erich L. Kaltofen[1], Bin Li[2], Zhengfeng Yang[3] and Lihong Zhi[2]

[1]Dept. of Mathematics, North Carolina State University,
Raleigh, North Carolina 27695-8205, USA
kaltofen@math.ncsu.edu; http://www.kaltofen.us

[2]Key Laboratory of Mathematics Mechanization, AMSS
Beijing 100190, China
{bli,lzhi}@mmrc.iss.ac.cn; http://www.mmrc.iss.ac.cn/~lzhi/

[3]Shanghai Key Laboratory of Trustworthy Computing, SEI
East China Normal University, Shanghai 200062, China
zfyang@sei.ecnu.edu.cn

In memory of Wenda Wu (1929–2009)

Abstract

We present a hybrid symbolic-numeric algorithm for certifying a polynomial or rational function with rational coefficients to be non-negative for all real values of the variables by computing a representation for it as a fraction of two polynomial sum-of-squares (SOS) with rational coefficients. Our new approach turns the earlier methods by Peyrl and Parrilo at SNC'07 and ours at ISSAC'08 both based on polynomial SOS, which do not always exist, into a universal algorithm for all inputs via Artin's theorem.

Furthermore, we scrutinize the all-important process of converting the numerical SOS numerators and denominators produced by block semidefinite programming into an exact rational identity. We improve on our own Newton iteration-based high precision refinement algorithm by compressing the initial Gram matrices and by deploying rational vector recovery aside from orthogonal projection. We successfully demonstrate our algorithm on 1. various exceptional SOS problems with necessary polynomial denominators from the literature and on 2. very large (thousands of variables) SOS lower bound certificates for Rump's model problem (up to $n = 18$, factor degree $= 17$).

*This material is based on work supported in part by the National Science Foundation under Grants CCF-0830347 (Kaltofen) and CCF-0514585 and DMS-0532140 (Kaltofen and Yang).

This research was partially supported by NKBRPC (2004CB318000) and the Chinese National Natural Science Foundation under Grant 10401035 (Li and Zhi).

1. Introduction

1.1. Overview of Results

Semidefinite programming (SDP) and the Gram matrix representation allow the computation of a polynomial sum-of-squares (SOS) of a positive semidefinite real polynomial. SDP has been deployed successfully to numerically compute a global minimum or infimum in many a polynomial or rational function optimization problem. Recently [Peyrl and Parrilo 2007; Kaltofen et al. 2008; Peyrl and Parrilo 2008], the numerical polynomial SOSes of some have been converted to exact ("symbolic") polynomial identities over the rational numbers, thus certifying rational lower bounds that are near the exact algebraic minima or infima.

This hybrid symbolic-numeric certification approach is complicated by several obstacles. For one, neither a polynomial SOS nor one with rational coefficients may exist for the optimization problem at hand. For the former, counterexamples like the Motzkin polynomial have been constructed, the latter is subject to conjecture (cf. [Hillar 2009]). However, Emil Artin's original solution to Hilbert's 17th problem shows that a rational sum-of-squares of rational functions always exists for any rational positive semidefinite rational polynomial (cf. (1) on page 3 below). Therefore we base our new certificates on expressing any arising positive semidefinite rational polynomial (floating point numbers are rational numbers) as a fraction of two polynomial SOSes with rational coefficients. By Artin's theorem such fractions always exist, and they can be computed via a primal SDP with 2 semidefinite block matrices provided the optimal value is known or computed by local methods. Since the degree of the denominator SOS controls the problem size, we use variable SOS denominators rather than Artin's original polynomial squares or Reznick's uniform denominators [Reznick 1995], and we present a case that has provably fewer control variables (see Example 2 on page 7 below).

Finally, we formulate our all-important process of conversion to an exact rational identity. Already in [Kaltofen et al. 2008] the problems with the necessary precision of the approach of [Peyrl and Parrilo 2007, 2008] were addressed, by performing "after-the-fact" high precision Newton iteration on the numerical SOSes from SDP. Since our rational lower bounds can be chosen (or have to be, if the optimum is irrational, as is the case in Rump's model problem of Section 3) so that the SDP becomes strictly feasible, singularities in the Gram matrix introduced by the actual real optimizers are avoided. With modification to our ISSAC'08 code, we now can certify better lower bounds for Rump's model problem, and can go further ($n = 18$, factor degree $= 17$). We conjecture that the arising polynomials are polynomial SOSes for all n, hence no SOS denominators need to be computed. Those are our largest SOS certificates with thousands of variables and thousands of decimal digits in the numerators and denominators of the rational scalars (see Table 2 on page 15 below). It is not known if nearness of the polynomial to a real root increases the degree of the necessary SOS denominators.

However, our conversion algorithm also yields SOS fractions if the polynomial input to our SOS construction has a real root. For that it turns out to be crucial that the numerically computed Gram matrices are analyzed and properly truncated before Newton refinement. If numerator SOS's Gram matrix is on the boundary of the cone of feasible solutions, rational coefficient vector recovery can yield an exact SOS when orthogonal projection always fails. We demonstrate that our method works on various exceptional SOS problems in the literature by Motzkin, Delzell, Reznick, Leep and Starr, the IMO'71 problem by A. Lax and P. Lax, and the polynomial Vor2 in [Everett et al. 2007]. In several cases, we have discovered entirely new SOS solutions. As stated in [Kaltofen et al. 2008], even though our approach uses numerical SDP and Newton iteration, all our certificates are exact without numerical error.

1.2. Rational Function Sum-of-squares and Semidefinite Programming

We shall now give a brief introduction to sum-of-squares optimization. Let $K = \mathbb{R}$ or $K = \mathbb{Q}$ and let $f \in K[X_1, \ldots, X_n]$ or $f \in K(X_1, \ldots, X_n)$. Emil Artin's [1927] solution of Hilbert's 17th Problem states that

$$\begin{gathered}\forall \xi_1, \ldots, \xi_n \in \mathbb{R}\colon f(\xi_1, \ldots, \xi_n) \not< 0 \text{ (f is [\textit{positive semi-}] \textit{definite})} \\ \Updownarrow \\ \exists u_0, \ldots, u_m \in K[X_1, \ldots, X_n]\colon f(X_1, \ldots, X_n) = \sum_{i=1}^{l} \left(\frac{u_i}{u_0}\right)^2 .\end{gathered} \tag{1}$$

Note that if f is a rational function, in the definition (1) of definiteness all real roots of the denominator of f, i.e., where f is undefined, are excluded by $\not<$ rather than $\geq$ ("für kein reelles Wertsystem der x_i negativ" [for no real value system of the x_i negative]). If f is a polynomial, a non-constant denominator u_0 may be necessary, the first explicit such example being given by Theodore Motzkin in 1967

$$\begin{aligned} \mathit{motzkin}(X, Y, Z) &= (3 \text{ arithm. mean} - 3 \text{ geom. mean})(X^4Y^2, X^2Y^4, Z^6) \\ &= X^4Y^2 + X^2Y^4 + Z^6 - 3X^2Y^2Z^2. \end{aligned}$$

Note that $(X^2+Z^2) \cdot \mathit{motzkin}(X, Y, Z) = (Z^4 - X^2Y^2)^2 + (XYZ^2 - X^3Y)^2 + (XZ^3 - XY^2Z)^2$ and that for general positive semidefinite $f \in K[X_1, \ldots, X_n]$ a denominator u_0 in any $n-1$ variables suffices, that without extending the coefficient field [Artin 1927, Satz 9]. In addition, $(X^2 + Y^2 + Z^2) \cdot \mathit{motzkin}(X, \lambda Y, \lambda Z)$ is a [*polynomial*] *sum-of-squares* (*SOS*) if and only if $0 \leq \lambda \leq 2$ [Reznick 2005]. For f a positive [definite] form (= homogeneous polynomial with no non-trivial real zero, the polynomial *motzkin* is not) with coefficients in $K = \mathbb{R}$ one can take the denominator as $u_0 = (X_1^2 + \cdots + X_n^2)^s$ for some integer exponent s (or, obviously, $2s'$ which solves Artin's equation) [Reznick 1995]. In Example 2 we show that $(X^2 + Y^2 + Z^2)^2 \cdot \mathit{motzkin}(X, 3Y, 3Z)$ is a polynomial SOS. However, there are positive semidefinite forms which admit no uniform denominator (see Example 1 below). We remark that

$$\mathit{motzkin}(X, Y, Z) + \epsilon \neq \frac{1}{(aX + bY + cZ + d)^2} \sum_j v_j(X, Y, Z)^2 \tag{2}$$

for any real $\epsilon \geq 0$, any non-zero $aX + bY + cZ + d \in \mathbb{R}[X, Y, Z]$ and any finite set of $v_j \in \mathbb{R}[X, Y, Z]$. The impossibility (2) exhibits that our polynomial SOS denominators may yield provably lower degrees than pure polynomial squares and that relaxing the lower bound of a polynomial may never yield a polynomial SOS ($a = b = c = 0$).

Even if $u_0 = 1$, there are polynomials for which the representation by squares of rational functions has provably fewer squares [Leep and Starr 2001]. Mihai Putinar's 1993 "Positivstellensatz" produces polynomial sums-of-squares under additional polynomial constraints that satisfy certain conditions [Nie and Schweighofer 2007]. Here we focus on unconstrained real polynomial optimization problems.

Polynomial sums-of-squares are related to positive semidefinite matrices in the following way. Let W be a real *symmetric* matrix. We define $W \succeq 0$ (positive semidefinite) if all its eigenvalues are non-negative. The $\mathrm{PLDL^TP^T}$-decomposition [Golub and Van Loan 1996, Section 4.2.9] gives the equivalent characterization

$$W \succeq 0 \Longleftrightarrow \exists L, D, P\colon P^T W P = L\ D\ L^T,\ P \text{ a permut. matrix, } D \text{ diagonal with } D_{i,i} \geq 0.$$

Therefore,

$$\exists u_i \in \mathbb{R}[X_1, \ldots, X_n]\colon f(X_1, \ldots, X_n) = \sum_{i=1}^{k} u_i(X_1, \ldots, X_n)^2$$

is equivalent to

$$\exists A\colon f = m_d(X_1, \ldots, X_n)^T\ A^T A\ m_d(X_1, \ldots, X_n) = \sum_{i=1}^{k} (A_i\ m_d(X_1, \ldots, X_n))^2$$

is equivalent to

$$\exists W \succeq 0\colon f = m_d(X_1, \ldots, X_n)^T\ W\ m_d(X_1, \ldots, X_n) = \sum_{i=1}^{\operatorname{rank} W} (\sqrt{D_{i,i}}\ L_i\ m_d(X_1, \ldots, X_n))^2,$$

with A_i (L_i) the i-th row of A ($L^T P^T$) and $m_d(X_1, \ldots, X_n)$ the vector of terms of degree $\leq d$ in the polynomials u_i.

Semidefinite Programming (SDP) [Vandenberghe and Boyd 1996; Wolkowicz et al. 2000] generalizes linear programming by restricting the decision variables to form positive semidefinite matrices. Let $A^{[i]}, C, W$ be real **symmetric** matrices. We define the scalar product on $\mathbb{R}^{n\times n}$ space as

$$C \bullet W = \sum_i \sum_j c_{i,j} w_{i,j} = \operatorname{Trace}(CW)$$

Letting $A^{[i,j]}, C^{[j]}, W^{[j]}$ be real symmetric matrices and letting $W =$ block diagonal($W^{[1]}$, ..., $W^{[k]}$), the *blocked* primal semidefinite program is

$$\begin{aligned} \min_{W^{[1]},\ldots,W^{[k]}} \quad & C^{[1]} \bullet W^{[1]} + \cdots + C^{[k]} \bullet W^{[k]} \\ \text{s. t.} \quad & \begin{bmatrix} A^{[1,1]} \bullet W^{[1]} + \cdots + A^{[1,k]} \bullet W^{[k]} \\ \vdots \\ A^{[m,1]} \bullet W^{[1]} + \cdots + A^{[m,k]} \bullet W^{[k]} \end{bmatrix} = b \in \mathbb{R}^m, \end{aligned}$$

$$\boxed{W^{[j]} \succeq 0, W^{[j]} = (W^{[j]})^T, j = 1, \ldots, k.}$$

We can now apply SDP to proving a polynomial $f \in \mathbb{Q}[X_1, \ldots, X_n]$ positive semidefinite by computing for a chosen denominator degree e Gram matrices $W^{[1]}$, $W^{[2]}$ such that

$$f(\bar{X}) = \frac{u_1(\bar{X})^2 + \cdots + u_l(\bar{X})^2}{v_1(\bar{X})^2 + \cdots + v_{l'}(\bar{X})^2} = \frac{m_d(\bar{X})^T\ W^{[1]}\ m_d(\bar{X})}{m_e(\bar{X})^T\ W^{[2]}\ m_e(\bar{X})}.$$

We show that $W^{[1]}$ and $W^{[2]}$ are solutions to a block SDP without an objective function.

First, let the term vector $m_d^T = [\tau_1, \tau_2, \ldots]$. Then

$$m_d^T\, W^{[1]}\, m_d = \begin{bmatrix} & \vdots & \\ \ldots & \tau_i \tau_j & \ldots \\ & \vdots & \end{bmatrix} \bullet W^{[1]} = \sum_i (G^{[i]} \bullet W^{[1]})\, t_i$$

where $G^{[i]}$ are scalar symmetric matrices and t_i are terms in $\bar{X}$. Similarly,

$$f(\bar{X})\, (m_e(\bar{X})^T\, W^{[2]}\, m_e(\bar{X})) = \sum_j (H^{[j]} \bullet W^{[2]})\, s_j$$

where $H^{[j]}$ are scalar symmetric matrices and s_j are terms in $\bar{X}$. Matching like terms we get block constraints of the form $G^{[i]} \bullet W^{[1]} - H^{[j]} \bullet W^{[2]} = 0$.

The block SDP was already used for rational function optimization. Suppose g is a positive real multivariate polynomial, and that the lower bound of $\mu_n = \min f/g$ is non-negative. In [Kaltofen et al. 2008] we have solved the sparse block SDP program

$$\left.\begin{array}{l} \mu_n^* := \sup\limits_{r \in \mathbb{R}, W}\ r \\ \text{s. t.}\quad f(\bar{X}) = m_{\mathcal{G}}(\bar{X})^T \cdot W \cdot m_{\mathcal{G}}(\bar{X}) + r g(\bar{X}) \\ \qquad\quad (\text{i.e., } f(\xi_1, \ldots, \xi_n) = \text{SOS} + r g(\xi_1, \ldots, \xi_n) \geq r g(\xi_1, \ldots, \xi_n)) \\ \qquad\quad W \succeq 0,\ W^T = W,\ r \geq 0 \end{array}\right\} \tag{3}$$

where $m_{\mathcal{G}}(\bar{X})$ is a term vector whose sparsity arises from the nature of f and g. Note that if $\mu_n < 0$ one can solve a second SDP in the decision variable $r' = -r \geq 0$ and the objective function $\sup -r'$, but in our cases that is not necessary.

2. Exact Rational Function Sum-Of-Squares Certificates

In the following, we focus on how to certify a rational lower bound $\tilde{r}$ of a polynomial $f \in \mathbb{Q}[X_1, \ldots, X_n]$. An initial floating point guess r^* for the lower bound can be obtained by computing local minima of f. Suppose we guess the degree of the denominator of the polynomials v_i, then the sizes of the $W^{[1]}$ and $W^{[2]}$ matrices are fixed and we solve the following SOS program:

$$\left.\begin{array}{l} \inf\limits_{W}\ \ \text{Trace}(W) \\ \text{s. t. } f(\bar{X}) - r^* = \dfrac{m_d(\bar{X})^T\, W^{[1]}\, m_d(\bar{X})}{m_e(\bar{X})^T\, W^{[2]}\, m_e(\bar{X})} \\ \qquad W = \begin{bmatrix} W^{[1]} & 0 \\ 0 & W^{[2]} \end{bmatrix}, W \succeq 0, W^T = W. \end{array}\right\} \tag{4}$$

Here $\text{Trace}(W)$ acts as a dummy objective function that is commonly used in SDP for optimization problem without an objective function. Unlike in (3) above, the optimum r^* in (4) cannot be found by SDP. However in (3) the assumption is made that $f - rg$ is actually a polynomial SOS.

The SOS program (4) can be solved efficiently by algorithms in GloptiPoly [Henrion and Lasserre 2005], SOSTOOLS [Prajna et al. 2004], YALMIP [Löfberg 2004] and SeDuMi [Sturm 1999]. However, since we are running fixed precision SDP solvers in Matlab, we can only obtain numerical positive semidefinite matrices $W^{[1]}, W^{[2]}$ which satisfy approximately

$$f(\bar{X}) - r^* \approx \frac{m_d(\bar{X})^T\, W^{[1]}\, m_d(\bar{X})}{m_e(\bar{X})^T\, W^{[2]}\, m_e(\bar{X})}, W^{[1]} \succeq 0\ , W^{[2]} \succeq 0. \tag{5}$$

So r^* is a lower bound of $\inf_{\xi\in\mathbb{R}^n} f(\xi)$, approximately! For some applications, such as Rump's model problem [Rump 2006; Kaltofen et al. 2008], due to the numerical error, the computed lower bounds can even be significantly larger than upper bounds, see, e.g., Table 1 in [Kaltofen et al. 2008]. These numerical problems motivate us to consider how to use symbolic computation tools to certify the lower bounds computed by SDP.

The lower bound $\tilde{r}$ is certified if $\tilde{r}$ and $\widetilde{W}^{[1]}$, $\widetilde{W}^{[2]}$ satisfy the following conditions exactly:

$$f(\bar{X}) - \tilde{r} = \frac{m_d(\bar{X})^T\, \widetilde{W}^{[1]}\, m_d(\bar{X})}{m_e(\bar{X})^T\, \widetilde{W}^{[2]}\, m_e(\bar{X})}, \quad \widetilde{W}^{[1]} \succeq 0\ , \widetilde{W}^{[2]} \succeq 0. \tag{6}$$

In the following subsections, we start with finding a rational positive semidefinite matrix $\widetilde{W}^{[2]}$ near to $W^{[2]}$ by solving the SOS program (4), then for the fixed denominator and lower bound, we use Gauss-Newton iterations to refine the matrix $W^{[1]}$. The rational number $\tilde{r}$ and rational positive semidefinite symmetric matrix $\widetilde{W}^{[1]}$ which satisfy (6) exactly can be computed by orthogonal projection ($\widetilde{W}^{[1]}$ is of full rank) or rational vector recovery ($\widetilde{W}^{[1]}$ is singular).

Without SDP, one may check that $(f - \tilde{r})y^2 + 1$ has no real point for a fresh variable y [Becker et al. 2000] by any of the generalizations of Seidenberg's algorithm [Safey El Din 2001; Aubry et al. 2002].

2.1. Newton Iteration

Our first two benchmark examples deal with Reznick's uniform denominators discussed in Section 1.2.

Example 1. The polynomial *delzell* [Delzell 1980] is a positive semidefinite polynomial which cannot be written as a polynomial SOS:

$$\mathit{delzell}(X_1, X_2, X_3, X_4) = X_1^4X_2^2X_4^2 + X_2^4X_3^2X_4^2 + X_1^2X_3^4X_4^2 - 3\,X_1^2X_2^2X_3^2X_4^2 + X_3^8.$$

It has been shown in [Delzell 1980; Reznick 2000] that this polynomial has a "bad point" at $(X_1, X_2, X_3, X_4) = (0, 0, 0, 1)$, i.e., for any polynomial q which is nonzero at $(0, 0, 0, 1)$, $q^2 \cdot$ *delzell* will never be a polynomial SOS. Reznick's uniform denominators do not vanish at this point, hence do not work for this example. Letting $r^* = 0$ and $m_e(\bar{X}) = [1, X_4, X_3, X_2, X_1]^T$

and solving the SOS program (4), we obtain the matrix $W^{[2]}$:

$$\begin{bmatrix} 0.00000000171 & 7.54\times10^{-14} & -1.88\times10^{-13} & -1.30\times10^{-17} & -1.58\times10^{-14} \\ 7.54\times10^{-14} & 0.00000000186 & 1.13\times10^{-12} & -5.91\times10^{-13} & 1.73\times10^{-12} \\ -1.88\times10^{-13} & 1.13\times10^{-12} & 2.08 & 0.00000000163 & -0.0000000144 \\ -1.30\times10^{-17} & -5.91\times10^{-13} & 0.00000000163 & 1.92 & -0.0000000322 \\ -1.58\times10^{-14} & 1.73\times10^{-12} & -0.0000000144 & -0.0000000322 & 1.95 \end{bmatrix}.$$

This shows clearly that the coefficient of X_4 in the denominator is near to zero. We choose the polynomial $2\,X_1^2+2\,X_2^2+2\,X_3^2$ as the denominator. The polynomial $\mathit{delzell}\cdot(2\,X_1^2+2\,X_2^2+2\,X_3^2)$ can be written as an SOS of 8 polynomials (14). □

Moreover, even when Reznick's uniform denominators work, a higher than necessary degree for that denominator may result.

Example 2. As stated in Section 1.2, the polynomial $\mathit{motzkin}(X_1, 3\,X_2, 3\,X_3)\cdot(X_1^2+X_2^2+X_3^2)$ is not a polynomial SOS.

We show in the Appendix that $s = 2$ is the least integer such that $\mathit{motzkin}(X_1, 3\,X_2, 3\,X_3)\cdot(X_1^2+X_2^2+X_3^2)^s$ can be written as an SOS of 5 polynomials (15). However, letting $r^* = 0$ and $m_e(\bar{X}) = [1, X_3, X_2, X_1]^T$ and solving the SOS program (4), we obtain the matrix $W^{[2]}$:

$$\begin{bmatrix} 0.000000151 & -4.02\times10^{-16} & -7.96\times10^{-18} & -1.84\times10^{-17} \\ -4.02\times10^{-16} & 0.237 & -1.45\times10^{-11} & 2.90\times10^{-12} \\ -7.96\times10^{-18} & -1.45\times10^{-11} & 0.134 & -1.38\times10^{-12} \\ -1.84\times10^{-17} & 2.90\times10^{-12} & -1.38\times10^{-12} & 0.0466 \end{bmatrix}.$$

A good candidate $\frac{1}{21}\,{X_1}^2 + \frac{1}{7}\,X_2^2 + \frac{1}{4}\,X_3^2$ for the denominator is discovered by converting the matrix $W^{[2]}$ to a nearby rational matrix. Actually, we show that the polynomial $\mathit{motzkin}(X_1, 3\,X_2, 3\,X_3)\cdot(4{X_1}^2+12X_2^2+21X_3^2)$ can be written as an SOS of 5 polynomials (16). □

As we have seen from the above two examples, it helps us to discover a proper denominator from the $W^{[2]}$ matrix computed by solving (4) for a chosen degree of denominator. Notice that we only convert the matrix $W^{[2]}$ to a nearby rational matrix which usually gives us a good candidate for the denominator.

Let us denote the computed denominator by $g(\bar{X})$. Then we are going to certify that $(f-r^*)\,g = f\cdot g - r^*g$ is nonnegative, where g is a polynomial SOS. By denoting $f\cdot g$ as f, and $W^{[1]}$ by W, we are facing the certification problem (3) which has already been addressed in [Kaltofen et al. 2008].

Suppose we have

$$f(\bar{X}) - r^*g(\bar{X}) \approx m_d(\bar{X})^T\cdot W\cdot m_d(\bar{X}),\quad W\succeq 0,\quad W^T = W.$$

In order to use structure-preserving Gauss-Newton iteration to refine W, we compute the $\mathrm{PLDL^TP^T}$-factorization of W and obtain the quadratic form:

$$f(\bar{X}) - r^*g(\bar{X}) \approx \sum_{i=1}^{k}(\sum_{\alpha} c_{i,\alpha}\bar{X}^{\alpha})^2 \in \mathbb{R}[\bar{X}]. \tag{7}$$

Here k is the rank of the matrix W.

We apply Gauss-Newton iteration to compute $\Delta c_{i,\alpha}\bar{X}^{\alpha}$ such that

$$f(\bar{X}) - r^* g(\bar{X}) = \sum_{i=1}^{k}(\sum_{\alpha} c_{i,\alpha}\bar{X}^{\alpha} + \Delta c_{i,\alpha}\bar{X}^{\alpha})^2 + O(\sum_{i=1}^{k}(\sum_{\alpha}\Delta c_{i,\alpha}\bar{X}^{\alpha})^2). \tag{8}$$

The matrix W is updated accordingly to $W + \Delta W$ and the iteration is stopped when θ is less than the given tolerance τ, θ denotes the backward error:

$$\theta = \|f(\bar{X}) - r^* g(\bar{X}) - m_d(\bar{X})^T \cdot W \cdot m_d(\bar{X})\|. \tag{9}$$

If θ remains greater than the given tolerance τ after several Gauss-Newton iterations, we may increase the precision of the SDP and Gauss-Newton iteration computations or use a smaller r^* and try the computations again.

The total number of $\bar{X}^{\alpha}$ in $m_d(\bar{X})$ is $\binom{n+d}{d}$. So the computation of Gauss-Newton iteration is very heavy. It is necessary to exploit the sparsity of the polynomials appearing on the right side of the equation (7) and the SOS program (4). Fortunately, for many optimization problems arising from approximate polynomial computation, the sparsity can be discovered by analyzing the Newton polytope. For example, we show in [Kaltofen et al. 2008] how to explore the sparsity for the Rump's model problem. Furthermore, the appearance of small entries $c_{i,\alpha}$, $1 \le i \le k$ also illustrates the sparsity of the SDP, see Example 3.

2.2. Rationalizing an SOS

2.2.1. Case 1: W is a full rank matrix

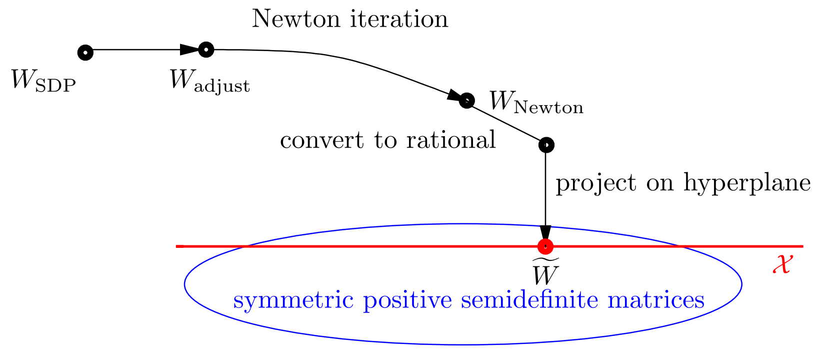

Figure 1: Rationalization of SOS

In [Peyrl and Parrilo 2007], a Macaulay 2 package is presented to compute an exact SOS decomposition from a numerical solution for nonnegative polynomials with rational coefficients. We extend their technique to construct an exact rational SOS decomposition for the polynomial $f(\bar{X}) - \tilde{r}g(\bar{X})$ in [Kaltofen et al. 2008].

Suppose W has been refined by Gauss-Newton iterations such that the error defined in (9) is less than the given tolerance, i.e., $\theta < \tau$. We approximate r^* by a nearby rational number

$\tilde{r} \lessapprox r^*$ and convert W to a rational matrix. The refined matrix W is projected to the rational matrix $\widetilde{W}$ on the hyperplane $\mathcal{X}$ defined by

$$\mathcal{X} = \{A \mid A^T = A, f(\bar{X}) - \tilde{r}g(\bar{X}) = m_d(\bar{X})^T \cdot A \cdot m_d(\bar{X}).\} \tag{10}$$

The orthogonal projection is achieved by solving exactly the following least squares problems:

$$\left.\begin{array}{l} \min\limits_{\widetilde{W}} \|W - \widetilde{W}\|_F^2 \\ \text{s. t. } f(\bar{X}) - \tilde{r}g(\bar{X}) = m_d(\bar{X})^T \cdot \widetilde{W} \cdot m_d(\bar{X}) \end{array}\right\} \tag{11}$$

which is equivalent to solve a set of smaller least squares problems:

$$\left.\begin{array}{l} \min\limits_{\widetilde{W}} \sum\limits_{\alpha} \sum\limits_{\beta+\gamma=\alpha} (W_{\beta,\gamma} - \widetilde{W}_{\beta,\gamma})^2 \\ \text{s. t. } f_\alpha - \tilde{r}g_\alpha = \sum\limits_{\beta+\gamma=\alpha} \widetilde{W}_{\beta,\gamma} \end{array}\right\} \tag{12}$$

By solving the least squares problem (12) for each α, we get the minimal rational solution, denoted by $\widetilde{W}$. Then we compute the exact PLDL$^{\mathrm{T}}$P$^{\mathrm{T}}$-decomposition [Golub and Van Loan 1996, see Exercise P5.2-3(c)] to check whether $\widetilde{W}$ is a symmetric positive semidefinite matrix. We also could use a validated numeric method, as Siegfried Rump has suggested to us. Even with exact PLDL$^{\mathrm{T}}$P$^{\mathrm{T}}$-decomposition, the step constitutes a small fraction of time of our procedure. The rational number $\tilde{r}$ is verified as the lower bound if

$$\left.\begin{array}{rl} f(\bar{X}) - \tilde{r}g(\bar{X}) = & m_d(\bar{X})^T \cdot \widetilde{W} \cdot m_d(\bar{X}) \\ = & m_d(\bar{X})^T \cdot PL \cdot D \cdot L^T P^T \cdot m_d(\bar{X}), \text{ such that } \forall i\colon D_{i,i} \geq 0, \end{array}\right\} \tag{13}$$

where $D_{i,i}$ is the i-th diagonal entry of the diagonal matrix D.

Suppose the minimal rational solution $\widetilde{W}$ is in the interior of the positive semidefinite cone, i.e., $\widetilde{W}$ is of full rank, then the orthogonal projection will always project the refined W matrix to $\widetilde{W}$ if θ is small enough [Peyrl and Parrilo 2007]. The Figure 1 is similar to that in [Kaltofen et al. 2008].

2.2.2. Case 2: W is a rank deficient matrix

In Figure 2 we show the situation when the matrix W obtained after applying Newton iteration is not of full rank or is near to a singular matrix. The hyperplane $\mathcal{X}$ defined by the linear constraints is tangent to the cone of symmetric positive semidefinite matrices, i.e., the $\widetilde{W}$ is not in the interior of the cone. The orthogonal projection introduced in the above section cannot project W on to the cone.

The rank deficiency of the matrix W can be caused by several reasons. Here we only list some typical cases:

1. There are extra monomials used in the SOS decomposition, see Leep and Starr's second example [Leep and Starr 2001].

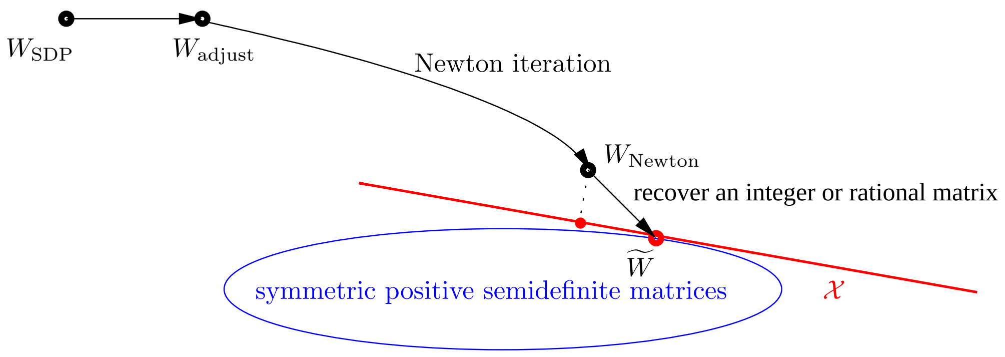

Figure 2: The Singular W Case

2. Suppose the lower bound $\tilde{r}$ is the global minimum of f/g and it is reached at nonzero real point $(\xi_1, \ldots, \xi_n) \in \mathbb{R}^n$, then the monomial vector $m_d(\bar{X})$ evaluated at $(\xi_1, \ldots, \xi_n)$ is a null vector of the matrix $\widetilde{W}$, i.e., $\widetilde{W}$ is singular.
   (a) The global minimum is achieved at few nonzero points on $\mathbb{R}^n$, such as Siegfried Rump's model problem [Rump 2006].
   (b) The global minimum is achieved at a manifold, for example Lax and Lax problem [Lax and Lax 1978] and Vor2 [Everett et al. 2007].

The first case can be avoided by exploring the sparsity structure of the polynomials or deleting the entire rows and columns of the W matrix with small elements, i.e., deleting the monomials which should not appear in the SOS of polynomials.

Example 3. [Example 2 in [Leep and Starr 2001]] Leep and Starr showed that the polynomial

$$\begin{aligned} \textit{leepstarr2}(X_1, X_2) \;&=\; 8 + \frac{1}{2}\, X_1^2 X_2^4 + X_1^2 X_2^3 - 2\, X_1^3 X_2^3 + 2\, X_1 X_2^2 + 10\, X_1^2 X_2^2 \\ &\quad + 4\, X_1^3 X_2^2 + 3\, X_1^4 X_2^2 + 4\, X_1 X_2 - 8\, X_1^2 X_2 \end{aligned}$$

is positive but cannot be written as a polynomial SOS. They showed that $\textit{leepstarr2}(X_1, X_2) \cdot (1 + X_1^2)^2$ can be written as an SOS of 3 polynomials.

For $r^* = 0$ and the fixed degree 2 for the denominator, we solve the SOS problem (4) and obtain the $W^{[2]}$ matrix:

$$\begin{bmatrix} 0.508 & 0.0155 & -0.0553 \\ 0.0155 & 0.598 & 0.0124 \\ -0.0553 & 0.0124 & 0.665 \end{bmatrix}$$

After converting it to a nearby rational matrix and multiplying by monomial vector $m_e(\bar{X}) = [1, X_2, X_1]^T$, we obtain the polynomial $\frac{1}{2} + \frac{2}{3}\, X_1^2 + \frac{3}{5}\, X_2^2$ as the denominator. We show that the polynomial $2 \cdot \textit{leepstarr2}(X_1, X_2) \cdot (15 + 20X_1^2 + 18\, X_2^2) \in \mathbb{Z}[X_1, X_1, X_3]$ can be written as an SOS of 9 polynomials (17).

The fifth and eighth rows of W corresponding to monomials X_1^2 and X_1^3 respectively can be deleted since there are of order 10^{-5}. After these two rows and two columns have been deleted, the new matrix W is of full rank. The Gauss-Newton iteration converges very quickly, and the backward error of the iteration θ can approach to zero as close as possible if the digits of computation is big enough.

The extra monomials can also be found by using Newton polytope in solving the SOS program (4). For Leep and Starr's example, if we run the Matlab's command solvesos and set sos.newton to 1, we obtain a full rank W matrix with dimension 9. The polynomial $2 \cdot \mathit{leepstarr2}(X_1, X_2) \cdot (15 + 20X_1^2 + 18\,X_2^2)$ can be written as an SOS of 9 polynomials (17) after we round the entries of W to integers and compute its exact $\mathrm{PLDL^TP^T}$-decomposition. $\square$

Suppose the global minimum $\tilde{r}$ of f/g is attained on finitely many real points. If the singularity is only introduced by the actual real optimizers, we can avoid the singularity by reducing the lower bound slightly and perform Gauss-Newton iteration to refine a full rank Gram matrix. Rump's model problem is this case. The global optima of Rump's model problem are achieved by one real optimizer if the degree n is even and two real optimizers if n is odd. We describe in [Kaltofen et al. 2008] that the numerical rank deficiency of the Gram matrix W computed by solving SOS program is 1 if n is even; and 2 if n is odd. Therefore, the singularity can be regarded as caused only by the global optimizers. In [Kaltofen et al. 2008], we compute the certified lower bound for Rump's model problem by forcing the rank deficient condition of W. Although we can compute the SOS decomposition to high precision, it is difficult to project the Gram matrix W on to the cone defined by the symmetric positive semidefinite matrices. From Table 2, we can see that if we ignore the rank deficient condition of W, and apply the Gauss-Newton iteration to refine the matrix W to a full rank matrix, then perform the orthogonal projection after the backward error θ being small enough, we can actually certify a much tighter lower bound.

If the global minimum is attained on some manifolds, adding small perturbation to the lower bound may not change the singularity of the matrix W. Hence we cannot avoid to work with the matrix W which is "intrinsically" singular. If the singular values of W have big jumps, we may estimate the rank deficiency of W and apply the Gauss-Newton iteration to the truncated triangular decomposition of W. When the residue θ is small, we can try to recover the rational entries of the matrix by a simultaneous diophantine approximation algorithm [Lagarias 1985].

Example 4. [Lax and Lax 1978] Consider the polynomial

$$\mathit{laxlax}(X_1, X_2, X_3, X_4) = X_1X_2X_3X_4 + \sum_{i=1}^{4}(-X_i)\prod_{j\neq i}(X_j - X_i)$$

which is positive semidefinite but cannot be written as a polynomial SOS. The minimum of *laxlax* is zero and is achieved at a manifold defined by

$$\{X_1 = 0, X_2 = X_3 = X_4\}.$$

We compute the polynomial SOS of *laxlax*$(X_1, X_2, X_3, X_4) \cdot (X_1^2 + X_2^2 + X_3^2 + X_4^2)$. The singular values of the 20×20 matrix W computed by the SDP solver are

$$12.999, 5.999, 5.999, 5.999, 2.999, 2.999, 8.19 \times 10^{-5}, 8.19 \times 10^{-5}, 8.19 \times 10^{-5}, 1.17 \times 10^{-9}, \ldots$$

The rank of W can be 7 or 10 for the given tolerance 10^{-4} or 10^{-8}, respectively. The norm of each row of W is at least 1, i.e., it is unlikely that we can eliminate the singularity by removing extra monomials. Let us truncate the matrix W to be of rank 7, i.e., the initial SOS consists of 7 polynomials computed from the truncated $\mathrm{LDL}^{\mathrm{T}}$-decomposition of W. By rational vector recovering, we obtain an exact polynomial SOS (18) which consists of 7 polynomials in all 20 monomials.

Since the last ten singular values are smaller than 10^{-8}, we can also guess that the rank of W could be 10. After performing Gauss-Newton iterations twice to the truncated W matrix, we can recover the same rational SOS (18)! □

There are also cases where the singularity caused by both extra monomials and the minimizers.

Example 5. [Everett et al. 2007; Safey El Din 2008] Let us look at the polynomial *voronoi2* which has 253 monomials and is of degree 18:

$$\textit{voronoi2}(a, \alpha, \beta, X, Y) = $$
$$a^{12}\alpha^6 + a^{12}\alpha^4 - 4\,a^{11}\alpha^3 Y - 4\,a^{11}\alpha^5 Y + 10\,a^{11}\alpha^4\beta\, X + \underbrace{\cdots}_{246 \text{ terms}} + 6\,a^{10}\alpha^2 Y^2 + 20\,a^{10}\alpha^2 X^2.$$

As claimed in [Everett et al. 2007], the polynomial *voronoi2* is nonnegative and the global minimum zero is reached on two manifolds defined by

$$\{Y + a\alpha, 2\,a\beta X + 4\,a^3\beta X + 4\,a^4\alpha^2 + 4\,a^4 + 4\,a^2\alpha^2 + 4\,a^2 - a^2X^2 - \beta^2\}$$

and

$$\{aX + \beta, -4\,\beta^2 - 4 - 2\,a^3\alpha Y - 4\,a\alpha Y + a^4\alpha^2 + a^2Y^2 - 4\,a^2\beta^2 - 4\,a^2\}.$$

The singular values of the computed W matrix have no big jumps:

$$196, 152.78, 152.29, 107.36, 68.64, 61.48, 43.05, 42.58, 25.06, 0.022, 0.02, \cdots$$

For different tolerances, we could get an exact polynomial SOS with the number of polynomials being 5, 7 or 8. Setting the tolerance being 43, we compute the truncated triangular decomposition of W with dimension 118×7. There are 42 rows with entries of order 10^{-5}. After eliminating 42 extra monomials, we round the product of the triangular matrix and it's transpose to an integer matrix. This gives us an exact polynomial SOS of the *voronoi2* (19). The number of polynomials in the SOS is 5. The actual number of monomials in the SOS is 76 instead of 118. For this example, if we choose different tolerances, for reasons unknown to us it is more difficult to obtain the exact SOS. □

The following table is presented to show the details of computation of all examples in this section. The exact sums of squares of these polynomials have been shown in the Appendix and http://www4.ncsu.edu/~kaltofen/software/certif and http://www.mmrc.iss.ac.cn/~lzhi/Research/hybrid/certif.

Table 1: SOSes for some well-known examples

Example	The Denominator	#iter	prec.	#sq	secs
delzell	$2X_1^2+2X_2^2+2X_3^2$	*Null*	2×15	8	0.02
$motzkin(X_1,3X_2,3X_3)$	$4X_1^2+12X_2^2+21X_3^2$	19	1×15	5	0.304
$motzkin(X_1,3X_2,3X_3)$	$(X_1^2+X_2^2+X_3^2)^2$	96	10×15	7	17.217
leepstarr2	$15+20X_1^2+18X_2^2$	*Null*	1×15	9	0.344
laxlax	$X_1^2+X_2^2+X_3^2+X_4^2$	*Null*	2×15	7	0.52
voronoi2	1	78	4×15	5	15.893

2.2.3. Algorithm

Algorithm *Lower Bound Verification*

Input: ▸ $f(\bar{X}_1,\dots,\bar{X}_n)\in\mathbb{Q}[\bar{X}_1,\dots,\bar{X}_n]$: a multivariate polynomial.
▸ r^*: the exact or approximate optimum of the minimization problem.
▸ $\tau\in\mathbb{R}_{>0}$: the given tolerance.

Output: ▸ $\tilde{r}$: the verified lower bound and its SOS certificate

1. Obtain the denominator
   (a) If $f-r^*$ can be written as an approximate SOS, then set the denominator be 1.
   (b) Otherwise, choose a degree for the denominator and solve the SDP system (4) and obtain $W^{[1]}$ and $W^{[2]}$ which satisfy (5).
2. Gauss-Newton refinement
   (a) Compute the numerical rank k of $W^{[1]}$ and exploit the sparsity structure of polynomials in the computed polynomial SOS.
   (b) Apply Gauss-Newton method to refine (7) and compute θ.
   (c) If $\theta<\tau$, then get the refined matrix W.
   Otherwise, decrease r^* and go back to step 2a.
3. Compute the exact SOS
   (a) Lower r^* to a rational number $\tilde{r}$ or let $\tilde{r}=r^*$ and convert W as a rational matrix. Check whether $\widetilde{W}$ satisfies (13). If so, return $\tilde{r}$. Otherwise, go to step 3b.
   (b) Compute the rational matrix $\widetilde{W}$ by solving (11) if W is of full rank or by rational vector recovering if W is singular.
   (c) Check whether $\widetilde{W}$ is positive semidefinite. If so, return $\tilde{r}$. Otherwise, decrease r^* and go back to step 2a.

Remark 1. In step 1b, the power can also be chosen as the denominator instead. For $s=1,2,\dots$, we try to find the least integer s such that $(f-r^*)\cdot(X_1^2+\cdots+X_n^2)^s$ can be written as an approximate SOS.

Remark 2. Our projection method tries to achieve positive semidefiniteness for a rational $\tilde{r}$ and $\widetilde{W}$ such that $\tilde{r}$ is as close as possible to r^*. We apply Gauss-Newton refinement to W for

r^* (or a lowered r^*) and project using the even smaller $\tilde{r}$. Refinement with the actual target $\tilde{r}$ seems to bring W too close to the boundary of the cone of positive semidefinite matrices, and orthogonal projection fails to preserve that property.

3. Siegfried Rump's Model Problem

Rump's [2006; 2009] model problem, related to structured condition numbers of Toeplitz matrices and polynomial factor coefficient bounds, asks for $n = 1, 2, 3, \ldots$ to compute the global minima

$$\mu_n = \min_{P,Q} \frac{\|PQ\|_2^2}{\|P\|_2^2 \|Q\|_2^2}$$
$$\text{s. t. } P(z) = \sum_{i=1}^{n} p_i z^{i-1}, Q(z) = \sum_{i=1}^{n} q_i z^{i-1} \in \mathbb{R}[z] \setminus \{0\}.$$

It has been shown in [Rump and Sekigawa 2006] that polynomials P, Q realizing the polynomials achieving μ_n must be symmetric (self-reciprocal) or skew-symmetric. Thus the problem can be rewritten into three optimization problems with three different constraints

$$\begin{array}{llll} k=1: & p_{n+1-i} = p_i, & q_{n+1-i} = q_i, & 1 \le i \le n, \\ k=2: & p_{n+1-i} = p_i, & q_{n+1-i} = -q_i, & 1 \le i \le n, \\ k=3: & p_{n+1-i} = -p_i, & q_{n+1-i} = -q_i, & 1 \le i \le n, \end{array}$$

and the smallest of three minima is equal to μ_n. For all three cases, we minimize the rational function $f(\bar{X})/g(\bar{X})$ with

$$f(\bar{X}) = \|PQ\|_2^2 = \sum_{k=2}^{2n} \Big(\sum_{i+j=k} p_i q_j\Big)^2, \quad g(\bar{X}) = \|P\|_2^2 \|Q\|_2^2 = \Big(\sum_{i=1}^{n} p_i^2\Big)\Big(\sum_{j=1}^{n} q_j^2\Big)$$

and the variables $\bar{X} = \{p_1, \ldots, p_{n(P)}\} \cup \{q_1, \ldots, q_{n(Q)}\}$, where $n(P) = n(Q) = \lceil n/2 \rceil$.

In [Kaltofen et al. 2008] we use Lagrangian multipliers with the polynomial constraints $\|P\|_2^2 = \|Q\|_2^2 = 1$ to compute local minima (upper bounds), which since then have been extended to $n = 95$: $\mu_{95} = 4.059969097152178\text{e–}93$ (Maple 12 with 12×15 decimal mantissa digits). They could be easily extended to even larger n. Furthermore, we certified in [Kaltofen et al. 2008] certain rational lower bounds via the sparse SOS-SDPs (3).

Having the upper bounds μ_n, the lower bounds r^* can be chosen by decreasing the upper bounds μ_n by a small amount. In [Kaltofen et al. 2008], after the Gauss-Newton refinement, the lower bound r_n was chosen by decreasing r^* by a tiny number, which is related to the backward error θ, and then we could certify that r_n is the lower bound. In this paper, we use higher precision and preserve the full rank of the matrix $W^{[1]}$ during Gauss-Newton iterations. As said before, the backward error of the iteration θ can be as arbitrarily small if the precision is big enough, that at least for the smaller n. Therefore, without decreasing r^*, we directly convert r^* to the exact rational number r_n and then certify successfully that r_n is the lower bound.

Table 2: The certified lower bounds

n	k	#iter	prec.	secs/iter	lower bound r_n	relative Δ_n	$\Delta_n^{[\text{ISSAC'08}]}$	#sq	logH
4	2	20	5×15	0.01	1.742917332e−02	5.738e−21	1.079e−16	4	219
5	1	30	7×15	0.04	2.339595548e−03	2.137e−20	5.309e−17	9	975
6	2	50	6×15	0.04	2.897318752e−04	4.934e−21	6.644e−15	9	881
7	1	60	10×15	0.27	3.418506980e−05	2.048e−14	2.018e−14	16	2485
8	2	80	6×15	0.24	3.905435600e−06	2.561e−15	7.681e−11	16	1563
9	1	280	10×15	1.75	4.360016539e−07	3.784e−14	6.881e−08	25	3919
10	2	280	12×15	1.89	4.783939568e−08	4.517e−13	8.361e−07	25	4660
11	1	510	13×15	9.62	5.178700000e−09	9.481e−06	1.931e−04	36	7201
12	2	210	5×15	8.79	5.545390000e−10	8.869e−05	5.439e−03	36	2881
13	1	270	5×15	41.93	5.881019273e−11	9.639e−04	1.728e−02	49	4271
14	2	440	25×15	33.68	6.100000000e−12	1.679e−02	9.368e−01	49	3121
15	1	1070	25×15	162.84	6.000000000e−13	8.239e−02	—	64	5751
16	2	640	25×15	153.94	6.000000000e−14	1.273e−01	—	64	5312
17	1	1650	10×15	504.10	1.000000000e−15	6.011e+00	—	81	12984
17	1	4200	10×15	380.75	6.000000000e−15	1.685e−01	—	81	13029
18	2	6440	10×15	344.75	1.000000000e−16	6.238e+00	—	81	12570
18	2	8800	10×15	352.62	3.000000000e−16	1.413e+00	—	81	12571
18	2	26800	10×15	330.36	7.000000000e−16	3.406e−02	—	81	12578

In Table 2 we report our currently achieved lower bounds for the Rump model problem. As stated above, for each n there are 3 cases to consider. In column 2 we give the case k which is the most costly to certify. This is the situation when the lower bound is near to the upper bound. In column 3, #iter denotes the number of Newton iterations performed in that case before the orthogonal projection yields a positive semidefinite Gram matrix. In our experiments, we try to project after each 10–200 iterations until we obtain the rational SOS identity. We perform our Newton iteration to the precision given in column 4, and report the timing as seconds per iteration in column 5. The total time is approximately the product of both, e.g., about 102.47 application CPU days for $n = 18$ and $k = 2$ and the sharpest bound 7.0e−16 (7.238394480e−16 being our upper bound) and about 18.51 application CPU days for $n = 17$ and $k = 1$ and the sharpest bound 6.0e−15 (7.011263198e−16 being our upper bound). Note that the smaller lower bounds for $n = 17$ and $n = 18$ required fewer iterations before rational positive semidefinite matrices were found. The same is true for the "easier cases:" for $n = 18$ and $k = 1$ we needed about 500 iterations (1.88 CPU days) and for $n = 18$ and $k = 3$ again about 500 iterations (1.87 CPU days) to prove the lower bound 7.3e−16 for both those cases, which is larger than the upper bound for $n = 18$, $k = 2$. The dimension of the matrix in our Newton iterations is $\binom{\lceil n/2 \rceil + 1}{2}^2 \times \lceil n/2 \rceil^4$, which is for $n = 18$ a 2025×6561 matrix. We ran our computation on several computers, including several MacPro's with 4, 8 and 16 cores (Intel Xeon 2.67GHz) with 4GB–32GB of real memory, respectively, running Linux version 2.6.22-16 (Ubuntu). We used Matlab 7.5.0.338, SeDuMi 1.2 and Maple 12. We noticed that running 2 processes with about 6GB memory allocation each on an older 4 core MacPro with 11GB real memory increased the process time by about 45% (11 CPU days), possibly due to memory bus contention (the Apple/Intel companies claim to have improved the memory bus on the new "Nehalem" quad-core processors). Table 2 exhibits the slowdown by the larger per iteration time for $n = 17$ for the smaller bound 1.0e$-$15.

The certified lower bound r_n is given in column 6, truncated as a floating point number to 11 mantissa digits. The actually computed lower bound is a rational number, and following a suggestion by Siegfried Rump, all digits are guaranteed in the stated floating point number, meaning that the rational lower bound may be slightly larger. Comparing to Mignotte's factor coefficient bound, we have

$$n = 17\colon\ 1/\mu_{17} \le 1.66 \cdot 10^{14} \text{ (ours above)} < \binom{32}{16}^2 \lessapprox 3.62 \cdot 10^{17} \text{ (Mignotte's)},$$
$$n = 18\colon\ 1/\mu_{18} \le 1.43 \cdot 10^{15} \text{ (ours above)} < \binom{34}{17}^2 \lessapprox 5.45 \cdot 10^{18} \text{ (Mignotte's)}.$$

In column 7 we give the relative distance to our computed upper bound. We believe that our upper bounds are the true (approximate) values, namely $\mu_n \approx r_n + r_n \times \Delta_n$. The given Δ_n use our more accurate rational lower bounds. In column 8 we compare to our earlier lower bounds in [Kaltofen et al. 2008], Table 2, again relative to the lower bounds given there: $\Delta_n^{\text{[ISSAC'08]}} \approx (\mu_n - r_n^{\text{[ISSAC'08]}})/r_n^{\text{[ISSAC'08]}}$. The number of squares in our certificate is given in column 9, and in column 10 we give the logarithmic height of the rational coefficients in the polynomials, i.e., the maximal number of decimal digits in any numerator or denominator.

In light of the immense size of our SOS certificates for the larger n in Table 2 we conclude with a brief discussion of our notion of what constitutes a certificate.

Definition 1. *A certificate for a problem that is given by input/output specifications is an input-dependent data structure and an algorithm that computes from that input and its certificate the specified output, and that has lower computational complexity than any known algorithm that does the same when only receiving the input. Correctness of the data structure is not assumed but valdidated by the algorithm (adversary-verifier model).*

We allow for Monte-Carlo randomization in our certification algorithm. Well-known examples are the (deterministic) Goldwasser-Kilian/Atkin certificates for the primality of an integer or Freivalds's randomized certification algorithm for a matrix product $A \times B = C$, where the output C also constitutes the data structure which the certification algorithm proves probabilistically in quadratic time by matrix-times-vector products $A(Bv) = Cv$ for vectors $v^T = [1\, y\, y^2 \ldots]$ with random y [Kimbrel and Sinha 1993].

A univariate integer polynomial can be certified positive definite by Sturm sequences. If one presents as the certificate the leading coefficients in the subresultant Sturm sequence, the certifying algorithm picks a small random prime and verifies those leading coefficients probabilistically in about quadratic bit complexity in the degree. If a polynomial sum-of-squares has small coefficients, better bit complexity is possible via that certificate.

Our certificates for the Rump model problem have large rational coefficients. Good certificates are the rational Gram matrix $\widetilde{W}$ and the diagonal matrix D in the LDL$^{\mathrm{T}}$-factorization of $\widetilde{W}$. Again the certifier picks a small random prime number and verifies both the identity $f(\bar{X}) \equiv m_{\mathcal{G}}(\bar{X})^T \cdot \widetilde{W} \cdot m_{\mathcal{G}}(\bar{X}) + \tilde{r} g(\bar{X}) \bmod p$ and $D \bmod p$ in the modular LDL$^{\mathrm{T}}$-factorization of $\widetilde{W} \bmod p$.

Acknowledgments: We thank Chris Hillar, Vicki Powers and Bruce Reznick for numerous comments on sums of squares, and Siegfried Rump for numerous comments on the model problem (Section 3).

References

Artin, Emil. Über die Zerlegung definiter Funktionen in Quadrate. *Abhandlungen Math. Seminar Univ. Hamburg*, 5(1):100–115, 1927.

Aubry, P., Rouillier, F., and Safey El Din, M. Real solving for positive dimensional systems. *J. Symbolic Comput.*, 34(6):543–560, December 2002. URL: http://www-spiral.lip6.fr/~safey/Articles/RR-3992.ps.gz.

Becker, E., Powers, V., and Wörmann, T. Deciding positivity of real polynomials. In [Delzell and Madden 2000], pages 251–272. URL: http://www.mathcs.emory.edu/~vicki/pub/psd.pdf.

Delzell, C. N. *A constructive, continuous solution to Hilbert's 17th problem, and other results in semi-algebraic geometry.* Ph.D. Thesis, Stanford University, 1980.

Delzell, Charles N. and Madden, James J., editors. *Real Algebraic Geometry and Ordered Structures*, volume 253 of *Contemporary Math.* AMS, 2000. ISBN 978-0-8218-0804-7.

Everett, Hazel, Lazard, Daniel, Lazard, Sylvain, and Safey El Din, Mohab. The Voronoi diagram of three lines in R^3. In *SoCG '07: Proceedings of the 23-rd Annual Symposium on Computational Geometry*, pages 255–264. ACM, New York, USA, 2007. ISBN 9781-595-9370-5-6.

Golub, G. H. and Van Loan, C. F. *Matrix Computations.* Johns Hopkins University Press, Baltimore, Maryland, third edition, 1996.

Henrion, Didier and Lasserre, Jean-Bernard. Detecting global optimality and extracting solutions in GloptiPoly. In Henrion, Didier and Garulli, Andrea, editors, *Positive polynomials in control*, volume 312 of *Lecture Notes on Control and Information Sciences*, pages 293–310. Springer Verlag, Heidelberg, Germany, 2005. URL: http://homepages.laas.fr/henrion/Papers/extract.pdf.

Hillar, Christopher. Sums of polynomial squares over totally real fields are rational sums of squares. *Proc. American Math. Society*, 137:921–930, 2009. URL: http://www.math.tamu.edu/~chillar/files/totallyrealsos.pdf.

Jeffrey, David, editor. *ISSAC 2008*, 2008. ACM Press. ISBN 978-1-59593-904-3.

Kaltofen, Erich, Li, Bin, Yang, Zhengfeng, and Zhi, Lihong. Exact certification of global optimality of approximate factorizations via rationalizing sums-of-squares with floating point scalars. In [Jeffrey 2008], pages 155–163, 2008. URL: EKbib/08/KLYZ08.pdf.

Kimbrel, Tracy and Sinha, Rakesh K. A probabilistic algorithm for verifying matrix products using $O(n^2)$ time and $\log_2 n + O(1)$ random bits. *Inf. Process. Lett.*, 45(2):107–110, 1993.

Lagarias, J. C. The computational complexity of simultaneous diophantine approximation problems. *SIAM J. Comp.*, 14:196–209, 1985.

Lax, A. and Lax, P. D. On sums of squares. *Linear Algebra and Applications*, 20:71–75, 1978.

Leep, David B. and Starr, Colin L. Polynomials in $\mathbb{R}[x, y]$ that are sums of squares in $\mathbb{R}(x, y)$. *Proc. AMS*, 129(11):3133–3141, 2001.

Löfberg, J. YALMIP : A toolbox for modeling and optimization in MATLAB. In *Proc. IEEE CCA/ISIC/CACSD Conf.*, Taipei, Taiwan, 2004. URL: http://control.ee.ethz.ch/~joloef/yalmip.php.

Nie, Jiawang and Schweighofer, Markus. On the complexity of Putinar's Positivstellensatz. *J. Complexity*, 23:135–70, 2007.

Peyrl, Helfried and Parrilo, Pablo A. A Macaulay 2 package for computing sum of squares decompositions of polynomials with rational coefficients. In Verschelde, Jan and Watt, Stephen M., editors, *SNC'07 Proc. 2007 Internat. Workshop on Symbolic-Numeric Comput.*, pages 207–208, New York, N. Y., 2007. ACM Press. ISBN 978-1-59593-744-5.

Peyrl, H. and Parrilo, P. A. Computing sum of squares decompositions with rational coefficients. *Theoretical Computer Science*, 409:269–281, 2008.

Prajna, S., Papachristodoulou, A., Seiler, P., and Parrilo, P. A. SOSTOOLS: Sum of squares optimization toolbox for MATLAB. Available from http://www.cds.caltech.edu/sostools and http://www.mit.edu/~parrilo/sostools, 2004.

Reznick, Bruce. Uniform denominators in Hilbert's seventeenth problem. *Math. Z.*, 220: 75–97, 1995.

Reznick, Bruce. Some concrete aspects of Hilbert's 17th problem. In [Delzell and Madden 2000], pages 251–272. Also in Seminaire de Structures Algébriques Ordonnées (F. Delon, M. A. Dickmann, D. Gondard eds.), Publ. Équipe de Logique, Univ. Paris VII, Jan. 1996. URL: http://www.math.uiuc.edu/~reznick/hil17.pdf.

Reznick, Bruce. On the absence of uniform denominators in Hilbert's 17th problem. *Proc. Amer. Math. Soc.*, 133:2829–2834, 2005.

Rump, Siegfried M. Global optimization: a model problem, 2006. URL: http://www.ti3.tu-harburg.de/rump/Research/ModelProblem.pdf.

Rump, Siegfried M. A model problem for global optimization, 2009. Manuscript, 6 pages.

Rump, Siegfried M. and Sekigawa, H. The ratio between the Toeplitz and the unstructured condition number, 2006. To appear. URL: http://www.ti3.tu-harburg.de/paper/rump/RuSe06.pdf.

Safey El Din, Mohab. *Résolution réelle des systèmes polynomiaux en dimension positive.* Thèse de doctorat, Univ. Paris VI (Univ. Pierre et Marie Curie), Paris, France, 2001. URL: http://www-spiral.lip6.fr/~safey/these_safey.ps.gz.

Safey El Din, Mohab. Computing the global optimum of a multivariate polynomial over the reals. In [Jeffrey 2008], 2008.

Sturm, Jos F. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11/12:625–653, 1999. ISSN 1055-6788.

Vandenberghe, L. and Boyd, S. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996.

Wolkowicz, Henry, Saigal, Romesh, and Vandenberghe, Lieven (Eds.). *Handbook of Semidefinite Programming.* Kluwer Academic, Boston, 2000. ISBN 0-7923-7771-0.

4. Appendix

The polynomials here can be copied and pasted into an ASCII file for computer processing.

Solution to Example 1:

$$\mathit{delzell}(X_1, X_2, X_3, X_4)*(2*X_1\hat{}2+2*X_2\hat{}2+2*X_3\hat{}2) = 1/2*f1\hat{}2+1/2*f2\hat{}2+1/2*f3\hat{}2 \\ +1/2*f4\hat{}2+1/2*f5\hat{}2+1/2*f6\hat{}2+1/2*f7\hat{}2+2/3*f8\hat{}2 \tag{14}$$

where

$f1 = 2 * X_3\hat{}5$

$f2 = 2 * X_2 * X_3\hat{}4$

$f3 = - X_1\hat{}2 * X_2\hat{}2 * X_4 - X_1\hat{}2 * X_3\hat{}2 * X_4 + 2 * X_2\hat{}2 * X_3\hat{}2 * X_4$

$f4 = - 2 * X_1\hat{}2 * X_2 * X_3 * X_4 + 2 * X_2\hat{}3 * X_3 * X_4$

$f5 = - 2 * X_1 * X_2\hat{}2 * X_3 * X_4 + 2 * X_1 * X_3\hat{}3 * X_4$

$f6 = 2 * X_1 * X_3\hat{}4$

$f7 = - 2 * X_1\hat{}3 * X_2 * X_4 + 2 * X_1 * X_2 * X_3\hat{}2 * X_4$

$f8 = - 3 / 2 * X_1\hat{}2 * X_2\hat{}2 * X_4 + 3 / 2 * X_1\hat{}2 * X_3\hat{}2 * X_4$

Solution to Example 2, Case 1:

$$\mathit{motzkin}(X_1, 3*X_2, 3*X_3)*(4*X_1\hat{}2+12*X_2\hat{}2+21*X_3\hat{}2) = \\ 1/15309*f1\hat{}2+1/8748*f2\hat{}2+1/2916*f3\hat{}2+1/6075*f4\hat{}2+100/26973*f5\hat{}2 \tag{15}$$

where

$f1 = - 1701 * X_1\hat{}2 * X_2\hat{}2 + 15309 * X_3\hat{}4$

$f2 = - 972 * X_1\hat{}2 * X_2 * X_3 + 8748 * X_2 * X_3\hat{}3$

$f3 = - 2916 * X_1 * X_2\hat{}2 * X_3 + 2916 * X_1 * X_3\hat{}3$

$f4 = - 891 / 2 * X_1\hat{}3 * X_2 - 4131 / 2 * X_1 * X_2\hat{}3 + 6075 * X_1 * X_2 * X_3\hat{}2$

$f5 = - 2997 / 100 * X_1\hat{}3 * X_2 + 26973 / 100 * X_1 * X_2\hat{}3$

Case 2:

$$\begin{aligned}&motzkin(X_1, 3{*}X_2, 3{*}X_3){*}(X_1\hat{}2{+}X_2\hat{}2{+}X_3\hat{}2)\hat{}2 = 1/324{*}f1\hat{}2{+}2182758400/284650799{*}f2\hat{}2\\ &{+}831180333080/3308566689{*}f3\hat{}2{+}1/324{*}f4\hat{}2{+}2182758400/388911398639{*}f5\hat{}2\\ &{+}1135621284025880/4397949999{*}f6\hat{}2{+}1/81{*}f7\hat{}2\end{aligned} \tag{16}$$

where

$f1 = - 687563 / 23360 * X_1\hat{}3 * X_2\hat{}2 + 256329 / 11680 * X_1 * X_2\hat{}4 - 1893231 / 23360 * X_1 * X_2\hat{}2 * X_3\hat{}2 + 324 * X_1 * X_3\hat{}4$

$f2 = - 2444707357 / 19644825600 * X_1\hat{}3 * X_2\hat{}2 + 1080028279 / 1091379200 * X_1 * X_2\hat{}4 + 284650799 / 2182758400 * X_1 * X_2\hat{}2 * X_3\hat{}2$

$f3 = - 367618521 / 831180333080 * X_1\hat{}3 * X_2\hat{}2 + 3308566689 / 831180333080 * X_1 * X_2\hat{}4$

$f4 = - 63459 / 11680 * X_1\hat{}4 * X_2 + 615591 / 23360 * X_1\hat{}2 * X_2\hat{}3 - 314289 / 23360 * X_1\hat{}2 * X_2 * X_3\hat{}2 + 324 * X_2 * X_3\hat{}4$

$f5 = - 13888467971 / 1091379200 * X_1\hat{}4 * X_2 - 138918975161 / 2182758400 * X_1\hat{}2 * X_2\hat{}3 + 388911398639 / 2182758400 * X_1\hat{}2 * X_2 * X_3\hat{}2$

$f6 = - 488661111 / 1135621284025880 * X_1\hat{}4 * X_2 + 4397949999 / 1135621284025880 * X_1\hat{}2 * X_2\hat{}3$

$f7 = 81 * X_1\hat{}2 * X_2\hat{}2 * X_3 - 81 * X_1\hat{}2 * X_3\hat{}3 - 81 * X_2\hat{}2 * X_3\hat{}3 + 81 * X_3\hat{}5$

Solution of Example 3:

$$\begin{aligned}&2{*}leepstarr2(X_1, X_2){*}(15{+}20{*}X_1\hat{}2{+}18{*}X_2\hat{}2) = 1/240{*}f1\hat{}2{+}1/320{*}f2\hat{}2{+}80/22991{*}f3\hat{}2\\ &{+}22531180/8104721423{*}f4\hat{}2{+}1191394049181/90484770084871{*}f5\hat{}2\\ &{+}5700540515346873/14099135780182345 72{*}f6\hat{}2\\ &{+}829029183874721928336/3090451998479592662 40713{*}f7\hat{}2\\ &{+}370854239817551119488 8556/1619663648500187724 4354939{*}f8\hat{}2\\ &{+}485899094550056317330 64817/2004304332536274655 6948178{*}f9\hat{}2\end{aligned} \tag{17}$$

where

$f1 = 31 * X_1\hat{}3 * X_2 - 114 / 7 * X_1\hat{}2 * X_2\hat{}2 + 2 * X_1 * X_2\hat{}3 - 24 * X_1\hat{}2 * X_2 + 26 * X_1 * X_2\hat{}2 + 46 * X_1 * X_2 + 240$

$f2 = - 160 * X_1\hat{}3 * X_2 + 76 * X_1\hat{}2 * X_2\hat{}2 + 5 * X_1 * X_2\hat{}3 + 49 * X_1\hat{}2 * X_2 + 68 / 7 * X_1 * X_2\hat{}2 - 96 * X_1 * X_2 + 320 * X_1 + 14 * X_2$

$f3 = 2 * X_1\hat{}3 * X_2 - 2573 / 40 * X_1\hat{}2 * X_2\hat{}2 + 1145 / 32 * X_1 * X_2\hat{}3 + 3999 / 1120 * X_1\hat{}2 * X_2 + 2783 / 40 * X_1 * X_2\hat{}2 + 41 / 5 * X_1 * X_2 + 22991 / 80 * X_2$

$f4 = 174094309 / 19312440 * X_1\hat{}3 * X_2 + 3219899 / 38780 * X_1\hat{}2 * X_2\hat{}2 + 86607559 / 9656220 * X_1 * X_2\hat{}3 + 13879281 / 804685 * X_1\hat{}2 * X_2 + 8104721423 / 22531180 * X_1 * X_2\hat{}2 - 744052543 / 9656220 * X_1 * X_2$

$f5 = - 520660500937 / 340398299766 * X_1\hat{}3 * X_2 + 90484770084871 / 1191394049181 * X_1\hat{}2 * X_2\hat{}2 - 10590902628287 / 340398299766 * X_1 * X_2\hat{}3 - 504715094021 / 113466099922 * X_1\hat{}2 * X_2 - 248023668923 / 170199149883 * X_1 * X_2$

$f6 = - 7215876065759977 / 1628725861527678 * X_1\hat{}3 * X_2 - 1487532334547597 / 1628725861527678 * X_1 * X_2\hat{}3 + 65629067980123895 / 1266786781188194 * X_1\hat{}2 * X_2 + 1409913578018234572 / 5700540515346873 * X_1 * X_2$

$f7 = 4262841971110918127267 / 39477580184510568016 * X_1\hat{}3 * X_2 + 36667815464017918127 / 39477580184510568016 * X_1 * X_2\hat{}3 + 309045199847959266240713 / 829029183874721928336 * X_1\hat{}2 * X_2$

$f8 = 16196636485001877244354939 / 3708542398175511194888556 * X_1\hat{}3 * X_2 + 484464299588342701863274 / 927135599543877798722139 * X_1 * X_2\hat{}3$

$f9 = 20043043325362746556948178 / 48589909455005631733064817 * X_1 * X_2\hat{}3$

Solution to Example 4:

$$laxlax(X_1, X_2, X_3, X_4)*(X_1\hat{}2+X_2\hat{}2+X_3\hat{}2+X_4\hat{}2) = f1\hat{}2+4/7*f2\hat{}2 +7/6*f3\hat{}2+8/11*f4\hat{}2+11/9*f5\hat{}2+f6\hat{}2+4*f7\hat{}2 \tag{18}$$

where

$f1 = - 1 / 2 * X_1\hat{}2 * X_4 + X_1 * X_2 * X_4 + X_1 * X_3 * X_4 - 1 / 2 * X_1 * X_4\hat{}2 - 1 / 2 * X_2\hat{}2 * X_4 + X_2 * X_3 * X_4 - 1 / 2 * X_2 * X_4\hat{}2 - 1 / 2 * X_3\hat{}2 * X_4 - 1 / 2 * X_3 * X_4\hat{}2 + X_4\hat{}3$

$f2 = - 1 / 2 * X_1\hat{}2 * X_3 + 3 / 4 * X_1\hat{}2 * X_4 - 1 / 2 * X_1 * X_2 * X_3 + X_1 * X_3\hat{}2 - 1 / 2 * X_1 * X_3 * X_4 - 3 / 4 * X_1 * X_4\hat{}2 - 1 / 2 * X_2\hat{}2 * X_3 + 3 / 4 * X_2\hat{}2 * X_4 + X_2 * X_3\hat{}2 - 1 / 2 * X_2 * X_3 * X_4 - 3 / 4 * X_2 * X_4\hat{}2 - 1 / 2 * X_3\hat{}3 - 5 / 4 * X_3\hat{}2 * X_4 + 7 / 4 * X_3 * X_4\hat{}2$

$f3 = 9 / 14 * X_1\hat{}2 * X_3 - 3 / 14 * X_1\hat{}2 * X_4 - 6 / 7 * X_1 * X_2 * X_3 + 3 / 14 * X_1 * X_3\hat{}2 - 6 / 7 * X_1 * X_3 * X_4 + 3 / 14 * X_1 * X_4\hat{}2 + 9 / 14 * X_2\hat{}2 * X_3 - 3 / 14 * X_2\hat{}2 * X_4 + 3 / 14 * X_2 * X_3\hat{}2 - 6 / 7 * X_2 * X_3 * X_4 + 3 / 14 * X_2 * X_4\hat{}2 - 6 / 7 * X_3\hat{}3 + 6 / 7 * X_3\hat{}2 * X_4$

$f4 = - 1 / 2 * X_1\hat{}2 * X_2 - 3 / 8 * X_1\hat{}2 * X_3 + 9 / 8 * X_1\hat{}2 * X_4 + X_1 * X_2\hat{}2 - 1 / 2 * X_1 * X_2 * X_3 - 1 / 2 * X_1 * X_2 * X_4 + 3 / 8 * X_1 * X_3\hat{}2 - 9 / 8 * X_1 * X_4\hat{}2 - 1 / 2 * X_2\hat{}3 + 5 / 8 * X_2\hat{}2 * X_3 - 7 / 8 * X_2\hat{}2 * X_4 - 1 / 8 * X_2 * X_3\hat{}2 - 1 / 2 * X_2 * X_3 * X_4 + 11 / 8 * X_2 * X_4\hat{}2$

$f5 = - 15 / 22 * X_1\hat{}2 * X_2 - 3 / 22 * X_1\hat{}2 * X_3 + 9 / 22 * X_1\hat{}2 * X_4 - 3 / 22 * X_1 * X_2\hat{}2 + 9 / 11 * X_1 * X_2 * X_3 + 9 / 11 * X_1 * X_2 * X_4 + 3 / 22 * X_1 * X_3\hat{}2 - 9 / 22 * X_1 * X_4\hat{}2 + 9 / 11 * X_2\hat{}3 - 3 / 11 * X_2\hat{}2 * X_3 - 9 / 11 * X_2\hat{}2 * X_4 - 6 / 11 * X_2 * X_3\hat{}2 + 9 / 11 * X_2 * X_3 * X_4$

$f6 = - X_1\hat{}2 * X_2 + X_1\hat{}2 * X_3 + X_1 * X_2\hat{}2 - X_1 * X_3\hat{}2 - X_2\hat{}2 * X_3 + X_2 * X_3\hat{}2$

$f7 = -\, 1 / 2 * X_1\hat{}3 + 1 / 4 * X_1\hat{}2 * X_2 + 1 / 4 * X_1\hat{}2 * X_3 + 1 / 4 * X_1\hat{}2 * X_4 + 1 / 4 * X_1 * X_2\hat{}2 - 1 / 2 * X_1 * X_2 * X_3 - 1 / 2 * X_1 * X_2 * X_4 + 1 / 4 * X_1 * X_3\hat{}2 - 1 / 2 * X_1 * X_3 * X_4 + 1 / 4 * X_1 * X_4\hat{}2$

Solution to Example 5:

$$\mathit{voronoi2}(a, \mathit{alph}, \mathit{beta}, X, Y) = f1\hat{}2{+}1/16{*}f2\hat{}2{+}f3\hat{}2{+}1/28{*}f4\hat{}2{+}7/27{*}f5\hat{}2 \tag{19}$$

where

$f1 = -\, a\hat{}5 * \mathit{alph}\hat{}3 * Y + a\hat{}5 * \mathit{alph}\hat{}2 * \mathit{beta} * X + a\hat{}6 * \mathit{alph}\hat{}2 + 2 * a\hat{}4 * \mathit{alph}\hat{}2 * Y\hat{}2 - 6 * a\hat{}4 * \mathit{alph} * \mathit{beta} * X * Y + 4 * a\hat{}4 * \mathit{beta}\hat{}2 * X\hat{}2 - 2 * a\hat{}5 * \mathit{alph} * Y + 4 * a\hat{}5 * \mathit{beta} * X + a\hat{}3 * \mathit{alph} * X\hat{}2 * Y - a\hat{}3 * \mathit{alph} * Y\hat{}3 - a\hat{}3 * \mathit{beta} * X\hat{}3 + a\hat{}3 * \mathit{beta} * X * Y\hat{}2 + a\hat{}4 * Y\hat{}2 + 4 * a\hat{}2 * \mathit{alph}\hat{}2 * Y\hat{}2 - 6 * a\hat{}2 * \mathit{alph} * \mathit{beta} * X * Y + 2 * a\hat{}2 * \mathit{beta}\hat{}2 * X\hat{}2 + a * \mathit{alph} * \mathit{beta}\hat{}2 * Y - a * \mathit{beta}\hat{}3 * X - 4 * a\hat{}4 + a\hat{}2 * X\hat{}2 + 4 * a * \mathit{alph} * Y - 2 * a * \mathit{beta} * X - 4 * a\hat{}2 + \mathit{beta}\hat{}2$

$f2 = -\, 4 * a\hat{}6 * \mathit{alph}\hat{}3 + 4 * a\hat{}5 * \mathit{alph}\hat{}2 * Y - 16 * a\hat{}5 * \mathit{alph} * \mathit{beta} * X - 4 * a\hat{}4 * \mathit{alph} * X\hat{}2 + 4 * a\hat{}4 * \mathit{alph} * Y\hat{}2 - 16 * a\hat{}4 * \mathit{beta} * X * Y + 16 * a\hat{}3 * \mathit{alph}\hat{}2 * Y - 24 * a\hat{}3 * \mathit{alph} * \mathit{beta} * X - 4 * a\hat{}3 * X\hat{}2 * Y - 4 * a\hat{}3 * Y\hat{}3 + 16 * a\hat{}4 * \mathit{alph} - 4 * a\hat{}2 * \mathit{alph} * \mathit{beta}\hat{}2 + 16 * a\hat{}2 * \mathit{alph} * Y\hat{}2 - 24 * a\hat{}2 * \mathit{beta} * X * Y + 16 * a\hat{}3 * Y - 4 * a * \mathit{beta}\hat{}2 * Y + 16 * a\hat{}2 * \mathit{alph} + 16 * a * Y$

$f3 = a\hat{}5 * \mathit{alph}\hat{}2 * X + a\hat{}4 * \mathit{alph}\hat{}2 * \mathit{beta} + 6 * a\hat{}4 * \mathit{alph} * X * Y - 4 * a\hat{}4 * \mathit{beta} * X\hat{}2 - 4 * a\hat{}5 * X + 6 * a\hat{}3 * \mathit{alph} * \mathit{beta} * Y - 4 * a\hat{}3 * \mathit{beta}\hat{}2 * X + a\hat{}3 * X\hat{}3 + a\hat{}3 * X * Y\hat{}2 - 4 * a\hat{}4 * \mathit{beta} + 4 * a\hat{}2 * \mathit{alph} * X * Y - a\hat{}2 * \mathit{beta} * X\hat{}2 + a\hat{}2 * \mathit{beta} * Y\hat{}2 - 4 * a\hat{}3 * X + 4 * a * \mathit{alph} * \mathit{beta} * Y - a * \mathit{beta}\hat{}2 * X - 4 * a\hat{}2 * \mathit{beta} + \mathit{beta}\hat{}3$

$f4 = 2 * a\hat{}4 * \mathit{alph}\hat{}2 * X * Y - 2 * a\hat{}4 * \mathit{alph} * \mathit{beta} * X\hat{}2 + 26 * a\hat{}5 * \mathit{alph} * X + 2 * a\hat{}3 * \mathit{alph}\hat{}2 * \mathit{beta} * Y - 2 * a\hat{}3 * \mathit{alph} * \mathit{beta}\hat{}2 * X + 2 * a\hat{}3 * \mathit{alph} * X * Y\hat{}2 - 2 * a\hat{}3 * \mathit{beta} * X\hat{}2 * Y + 26 * a\hat{}4 * \mathit{alph} * \mathit{beta} + 26 * a\hat{}4 * X * Y + 2 * a\hat{}2 * \mathit{alph} * \mathit{beta} * Y\hat{}2 - 2 * a\hat{}2 * \mathit{beta}\hat{}2 * X * Y + 28 * a\hat{}3 * \mathit{alph} * X + 26 * a\hat{}3 * \mathit{beta} * Y + 28 * a\hat{}2 * \mathit{alph} * \mathit{beta} + 28 * a\hat{}2 * X * Y + 28 * a * \mathit{beta} * Y$

$f5 = -\, 27 / 7 * a\hat{}4 * \mathit{alph}\hat{}2 * X * Y + 27 / 7 * a\hat{}4 * \mathit{alph} * \mathit{beta} * X\hat{}2 + 27 / 7 * a\hat{}5 * \mathit{alph} * X - 27 / 7 * a\hat{}3 * \mathit{alph}\hat{}2 * \mathit{beta} * Y + 27 / 7 * a\hat{}3 * \mathit{alph} * \mathit{beta}\hat{}2 * X - 27 / 7 * a\hat{}3 * \mathit{alph} * X * Y\hat{}2 + 27 / 7 * a\hat{}3 * \mathit{beta} * X\hat{}2 * Y + 27 / 7 * a\hat{}4 * \mathit{alph} * \mathit{beta} + 27 / 7 * a\hat{}4 * X * Y - 27 / 7 * a\hat{}2 * \mathit{alph} * \mathit{beta} * Y\hat{}2 + 27 / 7 * a\hat{}2 * \mathit{beta}\hat{}2 * X * Y + 27 / 7 * a\hat{}3 * \mathit{beta} * Y$