On Exact Polya and Putinar's Representations

Victor Magron[1,2] Mohab Safey El Din[2]

February 28, 2018

.

Abstract

We consider the problem of finding exact sums of squares (SOS) decompositions for certain classes of non-negative multivariate polynomials, relying on semidefinite programming (SDP) solvers.

We start by providing a hybrid numeric-symbolic algorithm computing exact rational SOS decompositions for polynomials lying in the interior of the SOS cone. It computes an approximate SOS decomposition for a perturbation of the input polynomial with an arbitrary-precision SDP solver. An exact SOS decomposition is obtained thanks to the perturbation terms. We prove that bit complexity estimates on output size and runtime are both polynomial in the degree of the input polynomial and simply exponential in the number of variables. Next, we apply this algorithm to compute exact Polya and Putinar's representations respectively for positive definite forms and positive polynomials over basic compact semi-algebraic sets. We also compare the implementation of our algorithms with existing methods in computer algebra including cylindrical algebraic decomposition and critical point method.

Keywords: Semidefinite programming, sums of squares decomposition, Polya's representation, Putinar's representation, hybrid numeric-symbolic algorithm, real algebraic geometry.

1 Introduction

Let $\mathbb{Q}$ (resp. $\mathbb{R}$) be the field of rational (resp. real) numbers and $X = (X_1, \ldots, X_n)$ be a sequence of variables. We consider the problem of deciding the non-negativity of $f \in \mathbb{Q}[X]$ either over $\mathbb{R}^n$ or over a semi-algebraic set S defined by some constraints $g_1 \geq 0, \ldots, g_m \geq 0$ (with $g_j \in \mathbb{Q}[X]$). Further, d denotes the maximum of the total degrees of these polynomials.

This problem is known to be NP hard [10]. The Cylindrical Algebraic Decomposition algorithm [13] allows to solve it in time doubly exponential in n (and polynomial in d). This complexity result has been improved later on, through the so-called critical point method, starting from [17] which culminates with [8] to establish that this decision problem can be solved in time $((m+1)d)^{O(n)}$. These latter ones have been developed to obtain implementations which reflect the complexity gain (see e.g. [3, 4, 40, 39, 6, 19, 5, 15, 16]) but still within a singly exponential complexity in n. Besides, these algorithms are "root finding" ones: they try to find a point at which f is negative over the considered domain. When f is positive, they return an empty list without a *certificate* that can be checked *a posteriori*.

To compute certificates of non-negativity, an approach based on *sums of squares* (SOS) decompositions (and their variants) has been popularized by Lasserre [26] and Parillo [33] (see also the survey [27] and references therein). In a nutshell, the idea is as follows.

A polynomial f is non-negative over $\mathbb{R}^n$ if it can be written as an SOS $s_1^2 + \cdots + s_r^2$ with $s_i \in \mathbb{R}[X]$ for $1 \leq i \leq r$. Also f is non-negative over the semi-algebraic set S if it can be written as $s_1^2 + \cdots + s_r^2 + \sum_{j=1}^m \sigma_j g_j$

[1]CNRS Verimag, 700 av Centrale, 38401 Saint-Martin d'Hères, France
[2]Sorbonne Université, CNRS, INRIA, Laboratoire d'Informatique de Paris 6, PolSys, Paris, France

where σ_i is a sum of squares in $\mathbb{R}[X]$ for $1 \leq j \leq m$. It turns out that, thanks to the "Gram matrix method" (see e.g. [26, 33]), computing such decompositions can be reduced to solving Linear Matrix Inequalities (LMI). This boils down to considering a semidefinite programming (SDP) problem.

For instance, on input $f \in \mathbb{Q}[X]$ of even degree $d = 2k$, the decomposition $f = s_1^2 + \cdots + s_r^2$ is a by-product of a decomposition of the form $f = v_k^T L^T D L v_k$ where v_k is the vector of all monomials of degree $\leq k$ in $\mathbb{Q}[X]$, L is a lower triangular matrix with non-negative real entries on the diagonal and D is a diagonal matrix with non-negative real entries. The matrices L and D are obtained after computing a symmetric matrix G (the Gram matrix), semidefinite positive, such that $f = v_k^T G v_k$. Such a matrix G is found using solvers for LMIs. Such inequalities can be solved symbolically (see [22]), but the degrees of the algebraic extensions needed to encode exactly the solutions are prohibitive on large examples [31]. Besides, there exist fast numerical solvers for solving LMIs implemented in double precision, e.g. SeDuMi [42], SDPA [43] as well as arbitrary-precision solvers, e.g. SDPA-GMP [30], successfully applied in many contexts, including bounds for kissing numbers [1] or computation of (real) radical ideals [23].

But using uniquely numerical solvers yields "approximate" non-negativity certificates. On our example, the matrices L and D (and consequently the polynomials $s_1, \ldots, s_r$) are not known exactly.

This raises topical questions. The first one is how to let interact symbolic computation with these numerical solvers to get *exact* certificates? Since not all positive polynomials are SOS, what to do when SOS certificates do not exist? Also, given inputs with rational coefficients, can we obtain certificates with rational coefficients?

For these questions, we inherit from previous contributions in the univariate case [11, 28] as well as in the multivariate case [34, 25]. Diophantine aspects are considered in [41, 20]. In the univariate (un)-constrained case, the algorithm from [11] computes an exact weighted SOS decomposition for a given positive polynomial $f \in \mathbb{Q}[X]$. The algorithm considers a perturbation of f, performs (complex) root isolation to get an approximate SOS decomposition of f. When the isolation is precise enough, the algorithm relies the perturbation terms to recover an exact rational decomposition. In the multivariate unconstrained case, Parillo and Peyrl designed a rounding-projection algorithm in [34] to compute a weighted rational SOS decompositon of a given polynomial f in the interior of the SOS cone. The algorithm computes an approximate Gram matrix of f, and rounds it to a rational matrix. With sufficient precision digits, the algorithm performs an orthogonal projection to recover an exact Gram matrix of f. The SOS decomposition is then obtained with an exact LDL^T procedure. This approach was significantly extended in [25] to handle rational functions.

Main contributions. This work provides an algorithmic framework to handle (un)-constrained polynomial problems with exact rational weighted SOS decompositions. The first contribution, given in Section 3, is a hybrid numeric-symbolic algorithm, called `intsos`, providing rational SOS decompositions for polynomials lying in the interior of the SOS cone. As for the algorithm from [11], the main idea is to perturbate the input polynomial, then to obtain an approximate Gram matrix of the perturbation by solving an SDP problem, and to recover an exact decomposition with the perturbation terms.

In Section 4, we rely on `intsos` to compute decompositions of positive definite forms into SOS of rational functions, based on Polya's representations, yielding a second algorithm, called `Polyasos`. In Section 5, we rely on `intsos` to compute weighted SOS decompositions for polynomials positive over compact semi-algebraic sets, yielding a third algorithm, called `Putinarsos`.

When the input is an n-variate polynomial of degree d with integer coefficients of maximum bit size τ, we prove in Section 3 that Algorithm `intsos` runs in boolean time $\tau^2 d^{\mathcal{O}(n)}$ and outputs SOS polynomials of bit size bounded by $\tau d^{\mathcal{O}(n)}$. This also yields bit complexity analysis for Algorithm `Polyasos` (see Section 4) and Algorithm `Putinarsos` (see Section 5). To the best of our knowledge, these are the first complexity estimates for the output of algorithms providing exact multivariate SOS decompositions.

The three algorithms are implemented within a Maple library, called `multivsos`. In Section 6, we provide numerical benchmarks to evaluate the performance of `multivsos` against existing methods based on CAD or critical point methods.

Acknowledgments. M. Safey El Din is supported by the ANR-17-CE40-0009 GALOP project and the GAMMA project funded by PGMO/FMJH.

2 Preliminaries

Let $\mathbb{Z}$ be the set of integers. For $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$, one has $|\alpha| := \alpha_1 + \cdots + \alpha_n$ and $X^\alpha := X_1^{\alpha_1} \ldots X_n^{\alpha_n}$. For all $k \in \mathbb{N}$, we let $\mathbb{N}^n_k := \{\alpha \in \mathbb{N}^n : |\alpha| \leq k\}$, whose cardinality is the binomial $\binom{n+k}{k}$. A polynomial $f \in \mathbb{R}[X]$ of degree $d = 2k$ is written as $f = \sum_{|\alpha| \leq d} f_\alpha X^\alpha$ and we identify f with its vector of coefficients $\mathbf{f} = (f_\alpha)$ in the basis (X^α), $\alpha \in \mathbb{N}^n_d$. Let $\Sigma[X]$ be the convex cone of sums of squares in $\mathbb{R}[X]$ and $\mathring{\Sigma}[X]$ be the interior of $\Sigma[X]$. We note $\Sigma_\mathbb{Z}(X) := \mathbb{Z}[X] \cap \Sigma[X]$ and $\mathring{\Sigma}_\mathbb{Z}[X]$ its interior. For instance, the polynomial $f = 4X_1^4 + 4X_1^3X_2 - 7X_1^2X_2^2 - 2X_1X_2^3 + 10X_2^4 = (2X_1X_2 + X_2^2)^2 + (2X_1^2 + X_1X_2 - 3X_2^2)^2$ belongs to $\Sigma_\mathbb{Z}(X)$.

We rely on the bit complexity model for complexity estimates. The bit size of an integer b is denoted by $\tau(b) := \log_2(|b|) + 1$ with $\tau(0) := 1$. For $f = \sum_{|\alpha| \leq d} f_\alpha X^\alpha \in \mathbb{Z}[X]$ of degree d, we note $\|f\|_\infty := \max_{|\alpha| \leq d} |f_\alpha|$ and $\tau(f) := \tau(\|f\|_\infty)$ with slight abuse of notation. Given $b \in \mathbb{Z}$ and $c \in \mathbb{Z}\backslash\{0\}$ with $\gcd(b, c) = 1$, we define $\tau(b/c) := \max\{\tau(b), \tau(c)\}$. For two mappings $g, h : \mathbb{N}^l \to \mathbb{R}$, we use the notation "$g(v) = \mathcal{O}(h(v))$" to state the existence of $b \in \mathbb{N}$ such that $g(v) \leq bh(v)$, for all $v \in \mathbb{N}^l$.

The *Newton polytope* or *cage* $\mathcal{C}(f)$ is the convex hull of the vectors of exponents of monomials that occur in $f \in \mathbb{R}[X]$. For the above example, $\mathcal{C}(f) = \{(4,0), (3,1), (2,2), (1,3), (0,4)\}$. For a symmetric real matrix G, we note $G \succeq 0$ (resp. $G \succ 0$) when G has only non-negative (resp. positive) eigenvalues and we say that G is *positive semidefinite* (SDP) (resp. *positive definite*).

With $f \in \mathbb{R}[X]$ of degree $d = 2k$, we consider the SDP program:

$$\inf_{G \succeq 0} \quad \mathrm{Tr}(G\,B_0) \quad \text{s.t.} \qquad \mathrm{Tr}(G\,B_\gamma) = f_\gamma\,, \quad \forall \gamma \in \mathbb{N}^n_d\,, \tag{1}$$

where B_γ has rows (resp. columns) indexed by $\mathbb{N}^n_k$ with (α, β) entry equal to 1 if $\alpha + \beta = \gamma$ and 0 otherwise.

Theorem 2.1. *[26, Theorem 3.2] Let $f \in \mathbb{R}[X]$ of degree $d = 2k$ and global minimum $f^\star := \inf_{\boldsymbol{x} \in \mathbb{R}^n} f(\boldsymbol{x})$. Assume that SDP (1) has a feasible solution $G^\star = \sum_{i=1}^r \lambda_i \mathbf{q}_i \mathbf{q}_i^T$, with the $\mathbf{q}_i$ being the eigenvectors of $G^\star$ corresponding to the non-negative eigenvalues λ_i, for all $i = 1, \ldots, r$. Then $f - f^\star = \sum_{i=1}^r \lambda_i q_i^2$.*

For the sake of efficiency, one reduces the size of matrix G indexing its rows and columns by half of $\mathcal{C}(f)$:

Theorem 2.2. *[37, Theorem 1] Let $f \in \Sigma[X]$ with $f = \sum_{i=1}^r s_i^2$ and $P := \mathcal{C}(f)$. Then for all $i = 1, \ldots, r$, $\mathcal{C}(s_i) \subseteq P/2$.*

Given $f \in \mathbb{R}[X]$, Theorem 2.1 states that one can theoretically certify that f lies in $\Sigma[X]$ by solving SDP (1). However, available SDP solvers are typically implemented in finite-precision and require the existence of a strictly feasible solution $G \succ 0$ to converge. This is equivalent for f to lie in $\mathring{\Sigma}[X]$ as stated in [12, Proposition 5.5]:

Theorem 2.3. *Let $f \in \mathbb{Z}[X]$ with $P := \mathcal{C}(f)$ and v_k be the vector of all monomials in $P/2$. Then $f \in \mathring{\Sigma}[X]$ if and only if there exists a positive definite matrix G such that $f = v_k^T G v_k$.*

3 Exact SOS representations

The aim of this section is to state and analyze a hybrid numeric-symbolic algorithm, called `intsos`, computing weighted SOS decompositions of polynomials in $\mathring{\Sigma}_\mathbb{Z}[X]$. This algorithm relies on perturbations of such polynomials.

Proposition 3.1. *Let $f \in \mathring{\Sigma}_\mathbb{Z}[X]$ of degree $d = 2k$, with $\tau = \tau(f)$ and $P = \mathcal{C}(f)$. Then, there exists $N \in \mathbb{N} - \{0\}$ such that for $\varepsilon := \frac{1}{2^N}$, $f - \varepsilon \sum_{\alpha \in P/2} X^{2\alpha} \in \mathring{\Sigma}[X]$. Moreover, $N = \tau(\varepsilon) \leq \tau d^{\mathcal{O}(n)}$.*

Proof. Let v_k be the vector of all monomials X^α in $P/2$. Note that each monomial in v_k has degree $\leq k$ and that $v_k^T v_k = \sum_{\alpha \in P/2} X^{2\alpha}$. Since $f \in \mathring{\Sigma}[X]$, there exists by Theorem 2.3 a matrix $G \succ 0$ such that $f = v_k^T G v_k$, with positive smallest eigenvalue λ. Let us define $N := \lceil \log_2 \frac{1}{\lambda} \rceil + 1$, i.e. the smallest integer such that $\varepsilon = \frac{1}{2^N} \leq \frac{\lambda}{2}$. Then, $\lambda > \varepsilon$ and the matrix $G - \varepsilon I$ has only positive eigenvalues. Hence, one has

$$f_\varepsilon := f - \varepsilon \sum_{\alpha \in P/2} X^{2\alpha} = v_k^T G v_k - \varepsilon v_k^T I v_k = v_k^T (G - \varepsilon I) v_k \,,$$

yielding $f_\varepsilon \in \mathring{\Sigma}[X]$.

For the second claim, let us consider the set $A := \{e \in \mathbb{R} : \forall \boldsymbol{x} \in \mathbb{R}^n, f(\boldsymbol{x}) - e \sum_{\alpha \in P/2} \boldsymbol{x}^{2\alpha} \geq 0\}$. Using [9, Thm 14.16], A is defined by univariate polynomials of degree in $d^{\mathcal{O}(n)}$ with coefficients of bit size bounded by $\tau d^{\mathcal{O}(n)}$. Hence the bit size of the mimimum absolute value of their non-zero real roots is below bounded by $\tau d^{\mathcal{O}(n)}$. □

The following can be found in [2, Lemma 2.1] and [2, Theorem 3.2].

Proposition 3.2. *Let $\tilde{G} \succ 0$ be a matrix with rational entries indexed on $\mathbb{N}^n_r$. Let L be the factor of $\tilde{G}$ computed using Cholesky's decomposition with finite precision δ_c. Then $LL^T = \tilde{G} + E$ where*

$$|E_{\alpha,\beta}| \leq (r+1) 2^{-\delta_c} |\tilde{G}_{\alpha,\alpha} \tilde{G}_{\beta,\beta}|^{\frac{1}{2}} / (1 - (r+1)2^{-\delta_c}) \,. \tag{2}$$

In addition, if the smallest eigenvalue $\tilde{\lambda}$ of $\tilde{G}$ satisfies the inequality

$$2^{-\delta_c} < \tilde{\lambda} / (r^2 + r + (r-1)\tilde{\lambda}) \,, \tag{3}$$

Cholesky's decomposition returns a rational nonsingular factor L.

3.1 Algorithm `intsos`

We present our algorithm `intsos` computing exact weighted rational SOS decompositions for polynomials in $\mathring{\Sigma}_{\mathbb{Z}}[X]$.

Given $f \in \mathbb{Z}[X]$ of degree $d = 2k$, one first computes its Newton polytope $P := \mathcal{C}(f)$ (see line 1) using standard algorithms such as quickhull [7]. The loop going from line 3 to line 4 finds a positive $\varepsilon \in \mathbb{Q}$ such that the perturbed polynomial $f_\varepsilon := f - \varepsilon \sum_{\alpha \in P/2} X^{2\alpha}$ is also in $\mathring{\Sigma}[X]$. This is done thanks to an oracle based on SDP or computer algebra procedures (e.g. CAD or critical points). If $f \in \mathring{\Sigma}_{\mathbb{Z}}[X]$, the existence of ε is ensured as in the proof of Theorem 3.1 if $A := \{e \in \mathbb{R} : \forall \boldsymbol{x} \in \mathbb{R}^n, f(\boldsymbol{x}) - e \sum_{\alpha \in P/2} \boldsymbol{x}^{2\alpha} \geq 0\}$ is non empty.

Next, we enter in the loop starting from line 6. Given $f_\varepsilon \in \mathbb{Z}[X]$, positive integers δ and R, the `sdp` function calls an SDP solver and tries to compute a rational approximation $\tilde{G}$ of the Gram matrix associated to f_ε together with a rational approximation $\tilde{\lambda}$ of its smallest eigenvalue. In practice, we use an arbitrary-precision SDP solver implemented with an interior-point method. However, in order to analyse the complexity of the procedure (see Remark 1), we assume that `sdp` relies on the ellipsoid algorithm [18].

Remark 1. In [14], the authors analyze the complexity of the short step, primal interior point method, used in SDP solvers. Within fixed accuracy, they obtain a polynomial complexity, as for the ellipsoid method, but the exact value of the exponents is not provided.

SDP problems are solved with this latter algorithm in polynomial-time within a given accuracy δ and a radius bound R on the Frobenius norm of $\tilde{G}$. The first step consists of solving SDP (1) by computing an approximate Gram matrix $\tilde{G} \succeq 2^{-\delta} I$ such that $|\operatorname{Tr}(\tilde{G} B_\gamma) - (f_\varepsilon)_\gamma| = |\sum_{\alpha+\beta=\gamma} \tilde{G}_{\alpha,\beta} - (f_\varepsilon)_\gamma| \leq 2^{-\delta}$ and $\sqrt{\operatorname{Tr}(\tilde{G}^2)} \leq R$. We pick large enough δ and R to obtain $\tilde{G} \succ 0$ and $\tilde{\lambda} > 0$ when $f_\varepsilon \in \mathring{\Sigma}[X]$.

The `cholesky` function computes the approximate Cholesky's decomposition LL^T of $\tilde{G}$ with precision δ_c. In order to guarantee that L will be a rational nonsingular matrix, a preliminary step consists of

Algorithm 1 `intsos`

Input: $f \in \mathbb{Z}[X]$, positive $\varepsilon \in \mathbb{Q}$, precision parameters $\delta, R \in \mathbb{N}$ for the SDP solver , precision $\delta_c \in \mathbb{N}$ for the Cholesky's decomposition

Output: list `c_list` of numbers in $\mathbb{Q}$ and list `s_list` of polynomials in $\mathbb{Q}[X]$

1: $P := \mathcal{C}(f)$
2: $t := \sum_{\alpha \in P/2} X^{2\alpha}$, $f_\varepsilon := f - \varepsilon t$
3: **while** $f_\varepsilon \notin \mathring{\Sigma}[X]$ **do** $\varepsilon := \frac{\varepsilon}{2}$, $f_\varepsilon := f - \varepsilon t$
4: **done**
5: ok := false
6: **while** not ok **do**
7: $(\tilde{G}, \tilde{\lambda}) := \texttt{sdp}(f_\varepsilon, \delta, R)$
8: $(s_1, \ldots, s_r) := \texttt{cholesky}(\tilde{G}, \tilde{\lambda}, \delta_c)$ ▷ $f_\varepsilon \simeq \sum_{i=1}^r s_i^2$
9: $u := f_\varepsilon - \sum_{i=1}^r s_i^2$
10: `c_list` $:= [1, \ldots, 1]$, `s_list` $:= [s_1, \ldots, s_r]$
11: **for** $\alpha \in P/2$ **do** $\varepsilon_\alpha := \varepsilon$
12: **done**
13: $\texttt{c\_list}, \texttt{s\_list}, (\varepsilon_\alpha) := \texttt{absorb}(u, P, (\varepsilon_\alpha), \texttt{c\_list}, \texttt{s\_list})$
14: **if** $\min_{\alpha \in P/2}\{\varepsilon_\alpha\} \geq 0$ **then** ok := true
15: **else** $\delta := 2\delta$, $R := 2R$, $\delta_c := 2\delta_c$
16: **end**
17: **done**
18: **for** $\alpha \in P/2$ **do**
19: `c_list` $:=$ `c_list` $\cup \{\varepsilon_\alpha\}$, `s_list` $:=$ `s_list` $\cup \{X^\alpha\}$
20: **done**
21: **return** `c_list`, `s_list`

verifying that the inequality from (3) holds, which happens when δ_c is large enough. Otherwise, `cholesky` selects the smallest δ_c such as (3) holds. Let v_k be the vector of all monomials X^α belonging to $P/2$ with size r. The output is a list of rational polynomials $[s_1, \ldots, s_r]$ such that for all $i = 1, \ldots, r$, s_i is the inner product of the i-th row of L by v_k. By Theorem 2.1, one would have $f_\varepsilon = \sum_{i=1}^r s_i^2$ with $s_i \in \mathbb{R}[X]$ after using exact SDP and Cholesky's decomposition. Here, we have to consider the remainder $u = f - \varepsilon \sum_{\alpha \in P/2} X^{2\alpha} - \sum_{i=1}^r s_i^2$, with $s_i \in \mathbb{Q}[X]$.

After these numeric steps, the algorithm starts to perform symbolic computation with the `absorb` subroutine at line 13. The loop from `absorb` is designed to obtain an exact weigthed SOS decomposition of $\varepsilon t + u = \varepsilon \sum_{\alpha \in P/2} X^{2\alpha} + \sum_\gamma u_\gamma X^\gamma$, yielding in turn an exact decomposition of f. Each term $u_\gamma X^\gamma$ can be written either $u_\gamma X^{2\alpha}$ or $u_\gamma X^{\alpha+\beta}$, for $\alpha, \beta \in P/2$. In the former case (line 2), one has

Algorithm 2 `absorb`

Input: $u \in \mathbb{Q}[X]$, multi-index set P, lists (ε_α) and `c_list` of numbers in $\mathbb{Q}$, list `s_list` of polynomials in $\mathbb{Q}[X]$

Output: lists (ε_α) and `c_list` of numbers in $\mathbb{Q}$, list `s_list` of polynomials in $\mathbb{Q}[X]$

1: **for** $\gamma \in \text{supp}(u)$ **do**
2: **if** $\gamma \in (2\mathbb{N})^n$ **then** $\alpha := \frac{\gamma}{2}$, $\varepsilon_\alpha := \varepsilon_\alpha + u_\gamma$
3: **else**
4: Find $\alpha, \beta \in P/2$ such that $\gamma = \alpha + \beta$
5: $\varepsilon_\alpha := \varepsilon_\alpha - \frac{|u_\gamma|}{2}$, $\varepsilon_\beta := \varepsilon_\beta - \frac{|u_\gamma|}{2}$
6: `c_list` $:=$ `c_list` $\cup \{\frac{|u_\gamma|}{2}\}$
7: `s_list` $:=$ `s_list` $\cup \{X^\alpha + \text{sgn}(u_\gamma) X^\beta\}$
8: **end**
9: **done**

$\varepsilon X^{2\alpha} + u_\gamma X^{2\alpha} = (\varepsilon + u_\gamma)X^{2\alpha}$. In the latter case (line 4), one has

$$\varepsilon(X^{2\alpha} + X^{2\beta}) + u_\gamma X^{\alpha+\beta} = |u_\gamma|/2(X^\alpha + \operatorname{sgn}(u_\gamma)X^\beta)^2 + (\varepsilon - |u_\gamma|/2)(X^{2\alpha} + X^{2\beta}).$$

If the positivity test of line 14 fails, then the coefficients of u are too large and one cannot ensure that $\varepsilon t + u$ is SOS. So we repeat the same procedure after increasing the precision of the SDP solver and Cholesky's decomposition.

In prior work [28], the authors and Schweighofer formalized and analyzed an algorithm called `univsos2`, initially provided in [11]. Given a univariate polynomial $f > 0$ of degree $d = 2k$, this algorithm computes weighted SOS decompositions of f. With $t := \sum_{i=0}^{k} X^{2i}$, the first numeric step of `univsos2` is to find ε such that the perturbed polynomial $f_\varepsilon := f - \varepsilon t > 0$ and to compute its complex roots, yielding an approximate SOS decomposition $s_1^2 + s_2^2$. The second symbolic step is very similar to the loop from line 1 to line 9 in `intsos`: one considers the remainder polynomial $u := f_\varepsilon - s_1^2 - s_2^2$ and tries to computes an exact SOS decomposition of $\varepsilon t + u$. This succeeds for large enough precision of the root isolation procedure. Therefore, `intsos` can be seen as an extension of `univsos2` in the multivariate case by replacing the numeric step of root isolation by SDP and keeping the same symbolic step.

Example 1. We apply Algorithm `intsos` on $f = 4X_1^4 + 4X_1^3X_2 - 7X_1^2X_2^2 - 2X_1X_2^3 + 10X_2^4$, with $\varepsilon = 1$, $\delta = R = 60$ and $\delta_c = 10$. Then $P/2 := \mathcal{C}(f)/2 = \{(2,0),(1,1),(0,2)\}$ (line 1). The loop from line 3 to line 4 ends and we get $f - \varepsilon t = f - (X_1^4 + X_1^2X_2^2 + X_2^2) \in \mathring{\Sigma}[X]$. The `sdp` (line 7) and `cholesky` (line 8) procedures yield $s_1 = 2X_1^2 + X_1X_2 - \frac{8}{3}X_2^2$, $s_2 = \frac{4}{3}X_1X_2 + \frac{3}{2}X_2^2$ and $s_3 = \frac{2}{7}X_2^2$. The remainder polynomial is $u = f - \varepsilon t - s_1^2 - s_2^2 - s_3^2 = -X_1^4 - \frac{1}{9}X_1^2X_2^2 - \frac{2}{3}X_1X_2^3 - \frac{781}{1764}X_2^4$.

At the end of the loop from line 1 to line 9, we obtain $\varepsilon_{(2,0)} = (\varepsilon - X_1^4 = 0$, which is the coefficient of X_1^4 in $\varepsilon t + u$. Then, $\varepsilon(X_1^2X_2^2 + X_2^4) - \frac{2}{3}X_1X_2^3 = \frac{1}{3}(X_1X_2 - X_2^2)^2 + (\varepsilon - \frac{1}{3})(X_1^2X_2^2 + X_2^4)$. In the polynomial $\varepsilon t + u$, the coefficient of $X_1^2X_2^2$ is $\varepsilon_{(1,1)} = \varepsilon - \frac{1}{3} - \frac{1}{9} = \frac{5}{9}$ and the coefficient of X_4^4 is $\varepsilon_{(0,2)} = \varepsilon - \frac{1}{3} - \frac{781}{1764} = \frac{395}{1764}$.

Eventually, we obtain the weighted rational SOS decomposition: $4X_1^4 + 4X_1^3X_2 - 7X_1^2X_2^2 - 2X_1X_2^3 + 10X_2^4 = \frac{1}{3}(X_1X_2 - X_2^2)^2 + \frac{5}{9}(X_1X_2)^2 + \frac{395}{1764}X_2^4 + (2X_1^2 + X_1X_2 - \frac{8}{3}X_2^2)^2 + (\frac{4}{3}X_1X_2 + \frac{3}{2}X_2^2)^2 + (\frac{2}{7}X_2^2)^2)$.

3.2 Correctness and bit size of the output

Let $f \in \mathring{\Sigma}_{\mathbb{Z}}[X]$ of degree $d = 2k$, $\tau = \tau(f)$ and $P = \mathcal{C}(f)$.

Proposition 3.3. *Let G be a positive definite Gram matrix associated to f and $0 < \epsilon \in \mathbb{Q}$ be such that $f_\varepsilon = f - \varepsilon \sum_{\alpha \in P/2} X^{2\alpha} \in \mathring{\Sigma}[X]$. Then, there exist positive integers δ, R such that $G - \varepsilon I$ is a Gram matrix associated to f_ε, satisfies $G - \varepsilon I \succeq 2^{-\delta} I$ and $\sqrt{\operatorname{Tr}(G - \varepsilon I^2)} \leq R$. Also, the maximal bit sizes of δ and R are upper bounded by $\tau d^{\mathcal{O}(n)}$.*

Proof. Let λ be the smallest eigenvalue of G. By Proposition 3.1, $G \succeq \varepsilon I$ for $\varepsilon = \frac{1}{2^N} \leq \frac{\lambda}{2}$. With $\delta = N + 1$, $2^{-\delta} = \frac{1}{2^{N+1}} \leq \frac{\lambda}{4} < \frac{\lambda}{2}$, yielding $G - \varepsilon \succeq \frac{\lambda}{2} I \succeq 2^{-\delta} I$. As $N \leq \tau d^{\mathcal{O}(n)}$, one has $\delta \leq \tau d^{\mathcal{O}(n)}$.

As in the proof of Proposition 3.1, we consider the largest eigenvalue λ' of the Gram matrix G of f and prove that the set $A' := \{e' \in \mathbb{R} : \forall \boldsymbol{x} \in \mathbb{R}^n, -f(\boldsymbol{x}) + e' \sum_{\alpha \in P/2} \boldsymbol{x}^{2\alpha} \geq 0\}$ is not empty. We use again [9, Thm 14.16] to prove that A' contains an interval $]0, \frac{1}{2^N}[$ with $N \leq \tau d^{\mathcal{O}(n)}$. This allows in turn to obtain a rational upper bound ε' of λ' with bit size $\tau d^{\mathcal{O}(n)}$). The size of G is bounded by $\binom{n+k}{n}$, thus the trace of G^2 is less than $\binom{n+k}{n}\varepsilon'^2$. Using that for all $k \geq 2$,

$$\binom{n+k}{n} = \frac{(n+k)\cdots(k+1)}{n!} = (1 + \frac{k}{n})(1 + \frac{k}{n-1})\cdots(1+k) \leq k^{n-1}(1+k) \leq 2k^n \leq d^n,$$

one has $\sqrt{\operatorname{Tr}(G - \varepsilon I)^2} \leq d^{\frac{n}{2}}\varepsilon' = \tau d^{\mathcal{O}(n)}$. □

Proposition 3.4. *Let f be as above. When applying Algorithm `intsos` to f, the procedure always terminates and outputs a weighted rational SOS decompositon of f. The maximum bit size of the coefficients involved in this SOS decomposition is upper bounded by $\tau d^{\mathcal{O}(n)}$.*

Proof. Let us first consider the loop of Algorithm `intsos` defined from line 3 to line 4. From Proposition 3.1, this loop terminates when $f_\varepsilon \in \mathring{\Sigma}[X]$ for $\varepsilon = \frac{1}{2^N}$ and $N \leq \tau d^{\mathcal{O}(n)}$.

When calling the `sdp` function at line 7 to solve SDP (1) with precision parameters δ and R, we compute an approximate Gram matrix $\tilde{G}$ of f_ε such that $\tilde{G} \succeq 2^\delta I$ and $\operatorname{Tr}(\tilde{G}^2) \leq R^2$. From Proposition 3.3, this procedure succeeds for large enough values of δ and R of bitisze upper bounded by $\tau d^{\mathcal{O}(n)}$. In this case, we obtain a positive rational approximation $\tilde{\lambda} \geq 2^{-\delta}$ of the smallest eigenvalue of $\tilde{G}$.

Then the Cholesky's decomposition of $\tilde{G}$ is computed when calling the `cholesky` function at line 8. The decomposition is guaranteed to succeed by selecting a large enough δ_c such that (3) holds. Let r be the size of $\tilde{G}$ and δ_c be the smallest integer such that $2^{-\delta_c} < \frac{2^{-\delta}}{r^2+r+(r-1)2^{-\delta}}$. Since the function $x \mapsto \frac{x}{r^2+r+(r-1)x}$ is increasing on $[0,\infty)$ and $\tilde{\lambda} \geq 2^{-\delta}$, (3) holds. We obtain an approximate weighted SOS decomposition $\sum_{i=1}^r s_i^2$ of f_ε with rational coefficients.

Let us now consider the remainder polynomial $u = f_\varepsilon - \sum_{i=1}^r s_i^2$. The second loop of Algorithm `intsos` defined from line 6 to line 17 terminates when for all $\alpha \in P/2$, $\varepsilon_\alpha \geq 0$. This condition is fulfilled when for all $\alpha \in P/2$, $\varepsilon - \sum_{\beta \in P/2} |u_{\alpha+\beta}|/2 + u_\alpha \geq 0$. This latter condition holds when for all $\gamma \in \operatorname{supp}(u)$, $|u_\gamma| \leq \frac{\varepsilon}{r}$.

Next, we show that this happens when the precisions δ of `sdp` and δ_c of `cholesky` are both large enough. From the definition of u, one has for all $\gamma \in \operatorname{supp}(u)$, $u_\gamma = f_\gamma - \varepsilon_\gamma - (\sum_{i=1}^r s_i^2)_\gamma$, where $\varepsilon_\gamma = \varepsilon$ when $\gamma \in (2\mathbb{N})^n$ and $\varepsilon_\gamma = 0$ otherwise. The positive definite matrix $\tilde{G}$ computed by the SDP solver is an approximation of an exact Gram matrix of f_ε. At precision δ, one has for all $\gamma \in \operatorname{supp}(f)$, $\tilde{G} \succeq 2^{-\delta} I$ such that

$$|f_\gamma - \varepsilon_\gamma - \operatorname{Tr}(\tilde{G} B_\gamma)| = |f_\gamma - \varepsilon_\gamma - \sum_{\alpha+\beta=\gamma} \tilde{G}_{\alpha,\beta}| \leq 2^{-\delta}\,.$$

In addition, it follows from (2) that the approximated Cholesky decomposition LL^T of $\tilde{G}$ performed at precision δ satisfies $LL^T = \tilde{G} + E$ with $|E_{\alpha,\beta}| \leq \frac{(r+1)2^{-\delta_c}}{1-(r+1)2^{-\delta_c}} |\tilde{G}_{\alpha,\alpha}\, \tilde{G}_{\beta,\beta}|^{\frac{1}{2}}$, for all $\alpha, \beta \in P/2$. Moreover, by using Cauchy-Schwartz inequality, one has

$$\sum_{\alpha \in P/2} \tilde{G}_{\alpha,\alpha} = \operatorname{Tr} \tilde{G} \leq \sqrt{\operatorname{Tr} I}\sqrt{\operatorname{Tr} \tilde{G}^2} \leq \sqrt{r} R\,.$$

For all $\gamma \in \operatorname{supp}(u)$, this yields

$$\Big|\sum_{\alpha+\beta=\gamma} \tilde{G}_{\alpha,\alpha}\, \tilde{G}_{\beta,\beta}\Big|^{\frac{1}{2}} \leq \sum_{\alpha+\beta=\gamma} \frac{\tilde{G}_{\alpha,\alpha} + \tilde{G}_{\beta,\beta}}{2} \leq \operatorname{Tr} \tilde{G} \leq \sqrt{r} R\,,$$

the first inequality coming again from Cauchy-Schwartz inequality.

Thus, for all $\gamma \in \operatorname{supp}(u)$, one has

$$\Big|\sum_{\alpha+\beta=\gamma} \tilde{G}_{\alpha,\beta} - (\sum_{i=1}^r s_i^2)_\gamma\Big| = \Big|\sum_{\alpha+\beta=\gamma} \tilde{G}_{\alpha,\beta} - \sum_{\alpha+\beta=\gamma} (LL^T)_{\alpha,\beta}\Big| = \Big|\sum_{\alpha+\beta=\gamma} E_{\alpha,\beta}\Big|\,,$$

which is bounded by

$$\frac{(r+1)2^{-\delta_c}}{1-(r+1)2^{-\delta_c}} \sum_{\alpha+\beta=\gamma} |\tilde{G}_{\alpha,\alpha}\, \tilde{G}_{\beta,\beta}|^{\frac{1}{2}} \leq \frac{\sqrt{r}(r+1)2^{-\delta_c} R}{1-(r+1)2^{-\delta_c}}\,.$$

Now, let us take the smallest δ such that $2^{-\delta} \leq \frac{\varepsilon}{2r} = \frac{1}{2^{N+1} r}$ as well as the smallest δ_c such that

$$\frac{\sqrt{r}(r+1)2^{-\delta_c} R}{1-(r+1)2^{-\delta_c}} \leq \frac{\varepsilon}{2r}\,,$$

that is $\delta = \lceil N + 1 + \log_2 r \rceil$ and $\delta_c = \lceil \log_2 R + \log_2(r+1) + \log_2(2^{N+1} r\sqrt{r} + 1) \rceil$.

From the previous inequalities, for all $\gamma \in \mathrm{supp}(u)$, it holds that

$$|u_\gamma| = |f_\gamma - \varepsilon_\gamma - (\sum_{i=1}^r s_i^2)_\gamma| \leq |f_\gamma - \varepsilon_\gamma - \sum_{\alpha+\beta=\gamma} \tilde{G}_{\alpha,\beta}| + |\sum_{\alpha+\beta=\gamma} \tilde{G}_{\alpha,\beta} - (\sum_{i=1}^r s_i^2)_\gamma| \leq \frac{\varepsilon}{2r} + \frac{\varepsilon}{2r} = \frac{\varepsilon}{r}\,.$$

This ensures that Algorithm `intsos` terminates.

Let us note

$$\Delta(u) := \{(\alpha,\beta) : \alpha+\beta \in \mathrm{supp}(u)\,, \alpha,\beta \in P/2\,, \alpha \neq \beta\}\,.$$

When terminating, the first output `c_list` of Algorithm `intsos` is a list of non-negative rational numbers containing the list $[1,\ldots,1]$ of length r, the list $\{\frac{|u_{\alpha+\beta}|}{2} : (\alpha,\beta) \in \Delta(u)\}$ and the list $\{\varepsilon_\alpha : \alpha \in \frac{P}{2}\}$. The second output `s_list` of Algorithm `intsos` is a list of monomials containing the list $[s_1,\ldots,s_r]$, the list $\{X^\alpha + \mathrm{sgn}\,(u_{\alpha+\beta})X^\beta : (\alpha,\beta) \in \Delta(u)\}$ and the list $\{X^\alpha : \alpha \in P/2\}$. From the output, we obtain the following weigthed SOS decomposition

$$f = \sum_{i=1}^r s_i^2 + \sum_{(\alpha,\,\beta)\,\in\,\Delta(u)} \frac{|u_{\alpha+\beta}|}{2}(X^\alpha + \mathrm{sgn}\,(u_{\alpha+\beta})X^\beta)^2 + \sum_{\alpha\,\in\,\frac{P}{2}} \varepsilon_\alpha X^{2\alpha}\,.$$

Now, we bound the bit size of the coefficients. Since $r \leq \binom{n+k}{n} \leq d^n$ and $N \leq \tau d^{\mathcal{O}(n)}$, one has $\delta \leq \tau d^{\mathcal{O}(n)}$. Similarly, $R, \delta_c \leq \tau d^{\mathcal{O}(n)}$. This bounds also the maximal bit size of the coefficients involved in the approximate decomposition $\sum_{i=1}^r s_i^2$ as well the coefficients of u. In the worst case, the coefficient ε_α involved in the exact SOS decomposition is equal to $\varepsilon - \sum_{\beta\in P/2} |u_{\alpha+\beta}|/2 + u_\alpha$ for some $\alpha \in P/2$. Using again that the cardinal r of $P/2$ is less than $\binom{n+k}{n} \leq d^n$, we obtain a maximum bit size upper bounded by $\tau d^{\mathcal{O}(n)}$. □

3.3 Bit complexity analysis

Theorem 3.5. *For f as above, there exist ε, δ, R, δ_c of bit sizes $\leq \tau d^{\mathcal{O}(n)}$ such that `intsos`$(f,\varepsilon,\delta,R,\delta_c)$ runs in boolean time $\tau^2 d^{\mathcal{O}(n)}$.*

Proof. We consider ε, δ, R and δ_c as in the proof of Proposition 3.4, so that Algorithm `intsos` only performs a single iteration within the two while loops before terminating. Thus, the bit size of each input parameter is upper bounded by $\tau d^{\mathcal{O}(n)}$.

Computing $\mathcal{C}(f)$ with the quickhull algorithm runs in boolean time $\mathcal{O}(V^2)$ for a polytope with V vertices. In our case $V \leq \binom{n+d}{n} \leq 2d^n$, so that this procedure runs in boolean time $\mathcal{O}(d^{2n})$. Next, we investigate the computational cost of the call to `sdp` at line 7. Let us note $n_{\mathrm{sdp}} = r$ (resp. m_{sdp}) the size (resp. number of entries) of $\tilde{G}$. This step consists of solving SDP (1), which is performed in $\mathcal{O}(n_{\mathrm{sdp}}^4 \log_2(2^\tau n_{\mathrm{sdp}}\, R\, 2^\delta))$ iterations of the ellipsoid method, where each iteration requires $\mathcal{O}(n_{\mathrm{sdp}}^2(m_{\mathrm{sdp}} + n_{\mathrm{sdp}}))$ arithmetic operations over $\log_2(2^\tau n_{\mathrm{sdp}}\, R\, 2^\delta)$-bit numbers (see e.g. [18]). Since $m_{\mathrm{sdp}}, n_{\mathrm{sdp}} \leq \binom{n+d}{n} \leq 2d^n$, one has $\log_2(2^\tau n_{\mathrm{sdp}}\, R\, 2^\delta) \leq \tau d^{\mathcal{O}(n)}$, $n_{\mathrm{sdp}}^2(m_{\mathrm{sdp}} + n_{\mathrm{sdp}}) \leq \mathcal{O}(\tau d^{3n})$ and $n_{\mathrm{sdp}}^4 \log_2(2^\tau n_{\mathrm{sdp}}\, R\, 2^\delta) \leq \tau d^{\mathcal{O}(n)}$. Overall, the ellipsoid algorithm runs in boolean time $\tau^2 d^{\mathcal{O}(n)}$ to compute the approximate Gram matrix $\tilde{G}$. We end with the cost of the call to `cholesky` at line 8. Cholesky's decomposition is performed in $\mathcal{O}(n_{\mathrm{sdp}}^3)$ arithmetic operations over δ_c-bit numbers. Since $\delta_c \leq \tau d^{\mathcal{O}(n)}$, the function runs in boolean time $\tau d^{\mathcal{O}(n)}$. The other elementary arithmetic operations performed while running Algorithm `intsos` have a negligable cost w.r.t. to the `sdp` procedure. □

4 Exact Polya's representations

Next, we show how to apply Algorithm `intsos` to decompose positive definite forms into SOS of rational functions.

Let $G_n := \sum_{i=1}^n X_i^2$ and $\mathbb{S}^{n-1} := \{\boldsymbol{x} \in \mathbb{R}^n : G_n(\boldsymbol{x}) = 1\}$ be the unit $(n-1)$-sphere. A positive definite form $f \in \mathbb{R}[X]$ is a homogeneous polynomial which is positive over $\mathbb{S}^{n-1}$. For such a form, we set

$$\varepsilon(f) := \frac{\min_{\boldsymbol{x}\in\mathbb{S}^{n-1}} f(\boldsymbol{x})}{\max_{\boldsymbol{x}\in\mathbb{S}^{n-1}} f(\boldsymbol{x})},$$

which measures how close f is to having a zero in $\mathbb{S}^{n-1}$. While there is no guarantee that $f \in \Sigma[X]$, Reznick proved in [38] that for large enough $D \in \mathbb{N}$, $fG_n^D \in \Sigma[X]$. The proof being based on prior work by Polya [35], such SOS decompositions are called *Polya's representations* and D is called the Polya's degree. Our next result states that for large enough $D \in \mathbb{N}$, $fG_n^D \in \mathring{\Sigma}[X]$.

Lemma 4.1. *Let f be a positive definite form of degree d in $\mathbb{Z}[X]$ and $D \geq \frac{nd(d-1)}{4\log 2\,\varepsilon(f)} - \frac{n+d}{2}$. Then $f\,G_n^{D+1} \in \mathring{\Sigma}[X]$.*

Proof. Let $P := \mathcal{C}(f)$ and $t := \sum_{\alpha\in P/2} X^{2\alpha}$. Since f is a form, then each term $X^{2\alpha}$ has degree d, for all $\alpha \in P/2$, thus t is a form. First, we show that for any positive $e < \frac{\min_{\boldsymbol{x}\in\mathbb{S}^{n-1}} f(\boldsymbol{x})}{\max_{\boldsymbol{x}\in\mathbb{S}^{n-1}} t(\boldsymbol{x})}$, the form $(f - et)$ is positive definite: for any nonzero $\boldsymbol{x} \in \mathbb{R}^n$, one has

$$f(\boldsymbol{x}) - et(\boldsymbol{x}) = G_n(\boldsymbol{x})^d[f\big(\frac{\boldsymbol{x}}{G_n(\boldsymbol{x})}\big) - et\big(\frac{\boldsymbol{x}}{G_n(\boldsymbol{x})}\big)] > 0$$

since $(f - et)$ is positive on $\mathbb{S}^{n-1}$. Next, [38, Theorem 3.12] implies that for any positive integer D_e such that

$$D_e \geq \underline{D_e} := \frac{nd(d-1)}{4\log 2\,\varepsilon(f-et)} - \frac{n+d}{2},$$

one has $(f - et)\,G_n^{D_e} \in \Sigma[X]$. As in the proof of Proposition 3.1, this yields $f\,G_n^{D_e} \in \mathring{\Sigma}[X]$.

Next, with $\underline{D} = \frac{nd(d-1)}{4\log 2\,\varepsilon(f)} - \frac{n+d}{2}$, we prove that there exists $N \in \mathbb{N}$ such that for $e = \frac{\min_{\boldsymbol{x}\in\mathbb{S}^{n-1}} f(\boldsymbol{x})}{N \max_{\boldsymbol{x}\in\mathbb{S}^{n-1}} t(\boldsymbol{x})}$, $\underline{D_e} \leq \underline{D} + 1$. Since $f\,G_n^{D_e} \in \mathring{\Sigma}[X]$ for all $D_e \geq \underline{D_e}$, this will yield the desired result. For any $\boldsymbol{x} \in \mathbb{S}^{n-1}$, one has

$$\min_{\boldsymbol{x}\in\mathbb{S}^{n-1}} f(\boldsymbol{x}) - e \max_{\boldsymbol{x}\in\mathbb{S}^{n-1}} t(\boldsymbol{x}) \leq f(\boldsymbol{x}) - et(\boldsymbol{x}) \leq \max_{\boldsymbol{x}\in\mathbb{S}^{n-1}} f(\boldsymbol{x}).$$

Hence we obtain the following:

$$\varepsilon(f - et) \geq \frac{\min_{\boldsymbol{x}\in\mathbb{S}^{n-1}} f(\boldsymbol{x}) - e\max_{\boldsymbol{x}\in\mathbb{S}^{n-1}} t(\boldsymbol{x})}{\max_{\boldsymbol{x}\in\mathbb{S}^{n-1}} f(\boldsymbol{x})} = \varepsilon(f)\frac{N-1}{N}.$$

Therefore, one has $\underline{D_e} \leq \frac{N}{N-1}\frac{nd(d-1)}{4\log 2\,\varepsilon(f)} - \frac{n+d}{2}$, yielding $\underline{D_e} - \underline{D} \leq \frac{1}{N-1}\frac{nd(d-1)}{4\log 2\,\varepsilon(f)}$. By choosing $N := \lfloor \frac{nd(d-1)}{4\log 2\,\varepsilon(f)} - 1 \rfloor$, one ensures that $\underline{D_e} - \underline{D} \leq 1$, which concludes the proof. □

Algorithm `Polyasos` takes as input $f \in \mathbb{Z}[X]$, finds the smallest $D \in \mathbb{N}$ such that $f\,G_n^D \in \mathring{\Sigma}[X]$, thanks to an oracle as in `intsos`. Then, `intsos` is applied on $f\,G_n^D$.

Algorithm 3 `Polyasos`

Input: $f \in \mathbb{Z}[X]$, positive $\varepsilon \in \mathbb{Q}$, precision parameters $\delta, R \in \mathbb{N}$ for the SDP solver, precision $\delta_c \in \mathbb{N}$ for the Cholesky's decomposition

Output: list `c_list` of numbers in $\mathbb{Q}$ and list `s_list` of polynomials in $\mathbb{Q}[X]$

1: $D := 0$
2: **while** $f\,G_n^D \notin \mathring{\Sigma}[X]$ **do** $D := D + 1$
3: **done**
4: **return** `intsos`$(f\,G_n^D, \varepsilon, \delta, R, \delta_c)$

Example 2. Let us apply `Polyasos` on the perturbed Motzkin polynomial $f = (1+2^{-20})(X_3^6 + X_1^4X_2^2 + X_1^2X_2^4) - 3X_1^2X_2^2X_3^2$. With $D = 1$, one has $f\,G_n = (X_1^2 + X_2^2 + X_3^2)\,f \in \mathring{\Sigma}[X]$ and `intsos` yields an SOS decomposition of $f\,G_n$ with $\varepsilon = 2^{-20}$, $\delta = R = 60$, $\delta_c = 10$.

Theorem 4.2. *Let $f \in \mathbb{Z}[X]$ be a positive definite form of degree d, coefficients of bit size at most τ. On input f, Algorithm* `Polyasos` *terminates and outputs a weighted SOS decomposition for f. The maximum bit size of its coefficients involved and the boolean running time of the procedure are both upper bounded by $2^{\tau d^{\mathcal{O}(n)}}$.*

Proof. By Lemma 4.1, the while loop from line 2 to 3 is ensured to terminate for a positive integer $D \geq \frac{nd(d-1)}{4\log 2\,\varepsilon(f)} - \frac{n+d}{2} + 1$. By Proposition 3.4, when applying `intsos` to $f\,G_n^D$, the procedure always terminates. The outputs are a list of non-negative rational numbers $[c_1, \ldots, c_r]$ and a list of rational polynomials $[s_1, \ldots, s_r]$ providing the weighted SOS decompositon $f\,G_n^D = \sum_{i=1}^r c_i s_i^2$. Thus, we obtain $f = \sum_{i=1}^r c_i \frac{s_i^2}{G_n^D}$, yielding the first claim.

Since, $(X_1^2 + \cdots + X_n^2)^D = \sum_{|\alpha|=D} \frac{D!}{\alpha_1!\cdots\alpha_n!} X^{2\alpha}$, each coefficient of G_n^D is upper bounded by $\sum_{|\alpha|=D} \frac{D!}{\alpha_1!\cdots\alpha_n!} = n^D$. Thus $\tau(f\,G_n^D) \leq \tau + D\log n$. Using again Proposition 3.4, the maximum bit size of the coefficients involved in the weighted SOS decomposition of $f\,G_n^D$ is upper bounded by $(\tau + D\log n)(d+2D)^{\mathcal{O}(n)}$. Now, we derive an upper bound of D. Since f is a positive form of degree d, one has

$$\min_{\boldsymbol{x}\in\mathbb{S}^{n-1}} f(\boldsymbol{x}) = \max\{e : \forall \boldsymbol{x} \in \mathbb{R}^n, f(\boldsymbol{x}) - eG_n(\boldsymbol{x})^d \geq 0\}\,.$$

Again, we rely on [9, Theorem 14.16] to show that $\min_{\boldsymbol{x}\in\mathbb{S}^{n-1}} f(\boldsymbol{x}) \geq 2^{-\tau d^{\mathcal{O}(n)}}$. Similarly, we obtain $\max_{\boldsymbol{x}\in\mathbb{S}^{n-1}} f(\boldsymbol{x}) \leq 2^{\tau d^{\mathcal{O}(n)}}$ and thus $\frac{1}{\varepsilon(f)} \leq 2^{\tau d^{\mathcal{O}(n)}}$. We obtain $\frac{nd(d-1)}{4\log 2\,\varepsilon(f)} - \frac{n+d}{2} + 1 \leq 2^{\tau d^{\mathcal{O}(n)}}$. This implies that $(\tau + D\log n)(d+2D)^{\mathcal{O}(n)} \leq 2^{\tau d^{\mathcal{O}(n)}}$. From Theorem 3.5, the boolean running time is upper bounded by $(\tau + D\log n)^2(d+2D)^{\mathcal{O}(n)}$, which ends the proof. □

5 Exact Putinar's representations

We let $f, g_1, \ldots, g_m$ in $\mathbb{Z}[X]$ of degree $\leq d$ and τ be a bound on the bit size of their coefficients. Assume that f is positive over $S := \{\boldsymbol{x} \in \mathbb{R}^n : g_1(\boldsymbol{x}) \geq 0, \ldots, g_m(\boldsymbol{x}) \geq 0\}$ and reaches its infimum with $f^\star := \min_{\boldsymbol{x}\in S} f(\boldsymbol{x}) > 0$. With $f = \sum_{|\alpha|\leq d} f_\alpha \boldsymbol{x}^\alpha$, we set $\|f\| := \max_{|\alpha|\leq d} \frac{f_\alpha \alpha_1!\cdots\alpha_n!}{|\alpha|!}$ and $g_0 := 1$.

We consider the quadratic module $\mathcal{Q}(S) := \left\{\sum_{j=0}^m \sigma_j g_j : \sigma_j \in \Sigma[\boldsymbol{x}]\right\}$ and, for $D \in \mathbb{N}$, the D-truncated quadratic module $\mathcal{Q}_D(S) := \left\{\sum_{j=0}^m \sigma_j g_j : \sigma_j \in \Sigma[\boldsymbol{x}],\ \deg(\sigma_j g_j) \leq D\right\}$ generated by $g_1, \ldots, g_m$. We say that $\mathcal{Q}(S)$ is *archimedean* if $N - G_n \in \mathcal{Q}(S)$ for some $N \in \mathbb{N}$. We also assume in this section:

Assumption 5.1. The set S is a basic compact semi-algebraic set with nonempty interior, included in $[-1,1]^n$ and $\mathcal{Q}(S)$ is archimedean.

Under Assumption 5.1, f is positive over S only if $f \in \mathcal{Q}_D(S)$ for some $D \in 2\mathbb{N}$ (see [36]). In this case, there exists a *Putinar's representation* $f = \sum_{i=0}^m \sigma_j g_j$ with $\sigma_j \in \Sigma[X]$ for $0 \leq j \leq m$. Let $w_j := \lceil \deg g_j/2 \rceil$, for all $1 \leq j \leq m$.

One can certify that $f \in \mathcal{Q}_D(S)$ for $D = 2k$ by solving the next SDP with $k \geq \max\{\lceil d/2 \rceil, w_1, \ldots, w_m\}$:

$$\begin{aligned} \inf_{G_0, G_1, \ldots, G_m \succeq 0} \quad & \operatorname{Tr}(G_0\, B_0) + \sum_{i=1}^m g_j(0) \operatorname{Tr}(G_j\, C_{j0}) \\ \text{s.t.} \quad & \operatorname{Tr}(G_0\, B_\gamma) + \sum_{j=1}^m \operatorname{Tr}(G_j\, C_{j\gamma}) = f_\gamma\,, \quad \forall \gamma \in \mathbb{N}^n_D\,, \end{aligned} \tag{4}$$

where B_γ is as for SDP (1) and $C_{j\gamma}$ has rows (resp. columns) indexed by $\mathbb{N}^n_{k-w_j}$ with (α,β) entry equal to $\sum_{\alpha+\beta+\delta=\gamma} g_{j\delta}$. SDP (4) is a reformulation of the problem $\sup\{b : f - b \in \mathcal{Q}_D(S)\}$, with optimal value denoted by $f_D^\star$. Next result follows from [26, Theorem 4.2].

Theorem 5.2. *We use the notation and assumptions introduced above. For $D \in 2\mathbb{N}$ large enough, one has $0 < f_D^\star \leq f^\star$. In addition, SDP (4) has an optimal solution $(G_0, G_1, \ldots, G_m)$, yielding the following*

Putinar's representation: $f - f_D^\star = \sum_{i=1}^{r} \lambda_{i0} q_{i0}^2 + \sum_{i=1}^{m} g_j \sum_{i=1}^{r_j} \lambda_{ij} q_{ij}^2$ *where the vectors of coefficients of the polynomials* q_{ij} *are the eigenvectors of* G_j *with respective eigenvalues* λ_{ij}*, for all* $j = 0, \ldots, m$.

The complexity of Putinar's Positivstellensätz was analyzed in [32]:

Theorem 5.3. *With the notation and assumptions introduced above, there exists a real* $\chi_S > 0$ *depending on* S *such that*

(i) for all even $D \geq \chi_S \exp\big(d^2 n^d \frac{\|f\|}{f^\star}\big)^{\chi_S}$, $f \in \mathcal{Q}_D(S)$.

(ii) for all even $D \geq \chi_S \exp\big(2d^2 n^d\big)^{\chi_S}$, $0 \leq f^\star - f_D^\star \leq \frac{6d^3 n^{2d} \|f\|}{\sqrt[\chi_S]{\log \frac{D}{\chi_S}}}$.

In theory, one can certify that f belongs to $\mathcal{Q}_D(S)$ for $D = 2k$ large enough, by solving SDP (4). Next, we show how to ensure the existence of a strictly feasible solution for SDP (4) after replacing the initial set of constraints S by S', defined as follows:

$$S' := \{\boldsymbol{x} \in S : 1 - \boldsymbol{x}^{2\alpha} \geq 0\,, \forall \alpha \in \mathbb{N}^n_k\}\,.$$

We first give a lower bound for $f^\star$.

Proposition 5.4. *With the above notation and assumptions, one has:*

$$f^\star \geq 2^{-(\tau + d + d\log_2 n + 1) d^{n+1}} d^{-(n+1)d^{n+1}} = 2^{-\tau d^{\mathcal{O}(n)}}\,.$$

Proof. Let $Y = (Y_1, \ldots, Y_n)$ and $\tilde{f} \in \mathbb{Z}[Y]$ be the polynomial obtained by replacing Y_i by $2nY_i - 1$ in f. Note that if $\boldsymbol{x} = (x_1, \ldots, x_n) \in S \subseteq [-1, 1]^n$, then $\boldsymbol{y} = \big(\big(\frac{x_i+1}{2n}\big)\big)_{1 \leq i \leq n}$ lies in the standard simplex Δ_n, so the polynomial $\tilde{f}$ takes only positive values over Δ_n. Since $x_i = 2ny_i - 1$ and $(2n-1)^d \leq (2n)^d$, the polynomial $\tilde{f}$ has coefficients of bit size at most $\tau + d + d\log_2 n$. Then, the desired result follows from [24, Theorem 1], stating that $\min_{\boldsymbol{y} \in \Delta_n} \tilde{f}(\boldsymbol{y}) > 2^{-(\tau(\tilde{f})+1)d^{n+1}} d^{-(n+1)d^{n+1}}$. □

Theorem 5.5. *We use the notation and assumptions introduced above. There exists* $D \in 2\mathbb{N}$ *such that:*
(i) $f \in \mathcal{Q}_D(S)$ *with the representation*

$$f = f_D^\star + \sum_{j=0}^{m} \sigma_j g_j$$

for $f_D^\star > 0$, $\sigma_j \in \Sigma[X]$ *with* $\deg(\sigma_j g_j) \leq D$ *for all* $j = 0, \ldots, m$.
(ii) $f \in \mathcal{Q}_D(S')$ *with the representation*

$$f = \sum_{j=0}^{m} \mathring{\sigma}_j g_j + \sum_{|\alpha| \leq k} c_\alpha (1 - X^{2\alpha})$$

for $\mathring{\sigma}_j \in \mathring{\Sigma}[X]$ *with* $\deg(\mathring{\sigma}_j g_j) \leq D$*, for all* $j = 0, \ldots, m$*, and some sequence of positive numbers* $(c_\alpha)_{|\alpha| \leq k}$.
(iii) There exists a real $C_S > 0$ *depending on* S *and* $\varepsilon = \frac{1}{2^N}$ *with positive* $N \in \mathbb{N}$ *such that* $f - \varepsilon \sum_{|\alpha| \leq k} X^{2\alpha} \in \mathcal{Q}_D(S')$ *and* $N \leq 2^{\tau d^{n C_S}}$*, where* τ *is the maximal bit size of the coefficients of* $f, g_1, \ldots, g_m$.

Proof. Let χ_S be as in Theorem 5.3 and $D = 2k$ be the smallest integer larger than $\underline{D}$ given by:

$$\underline{D} := \max\{\chi_S \exp\big(\frac{12 d^3 n^{2d} \|f\|}{f^\star}\big)^{\chi_S}, \chi_S \exp(2d^2 n^d)^{\chi_S}\}\,.$$

Theorem 5.3 implies $f \in \mathcal{Q}_D(S)$ and $f^\star - f_D^\star \leq \frac{6d^3 n^{2d}\|f\|}{\sqrt[\chi_S]{\log \frac{D}{\chi_S}}} \leq \frac{f^\star}{2}$.

(i) This yields the representation $f - f_D^\star = \sum_{j=0}^{m} \sigma_j g_j$, with $f_D^\star \geq \frac{f^\star}{2} > 0$, $\sigma_j \in \Sigma[X]$ and $\deg(\sigma_j g_j) \leq D$ for all $j = 0, \ldots, m$.

(ii) For $1 \leq j \leq m$, let us define $t_j := \sum_{|\alpha| \leq k-w_j} X^{2\alpha}$, $t_0 := \sum_{|\alpha| \leq k} X^{2\alpha}$ and $t := \sum_{j=0}^m t_j g_j$. For a given $\nu > 0$, we use the perturbation polynomial $-\nu t = -\nu \sum_{|\gamma| \leq D} t_\gamma X^\gamma$. For each term $-t_\gamma X^\gamma$, one has $\gamma = \alpha + \beta$ with $\alpha, \beta \in \mathbb{N}^n_k$, thus $-t_\gamma X^\gamma = |t_\gamma|(-1 + \frac{1}{2}(1 - X^{2\alpha}) + \frac{1}{2}(1 - X^{2\beta}) + \frac{1}{2}(X^\alpha - \operatorname{sgn}(t_\gamma) X^\beta)^2)$. As in the proof of Proposition 3.4, let us note $\Delta(t) := \{(\alpha, \beta) : \alpha + \beta \in \operatorname{supp}(t), \alpha, \beta \in \mathbb{N}^n_k, \alpha \neq \beta\}$. Hence, there exist $d_\alpha \geq 0$ for all $\alpha \in \mathbb{N}^n_k$ such that

$$f = f - \nu t + \nu t = f^\star_D - \sum_{|\gamma| \leq D} \nu |t_\gamma| + \sum_{j=0}^m \sigma_j g_j + \nu t + \sum_{|\alpha| \leq k} d_\alpha (1 - X^{2\alpha}) + \nu \sum_{(\alpha,\beta) \in \Delta(t)} \frac{|t_{\alpha+\beta}|}{2} (X^\alpha - \operatorname{sgn}(t_{\alpha+\beta}) X^\beta)^2 .$$

Since one has not necessarily $d_\alpha > 0$ for all $\alpha \in \mathbb{N}^n_k$, we now explain how to handle the case when $d_\alpha = 0$ for $\alpha \in \mathbb{N}^n_k$. We write

$$\begin{aligned} -\sum_{|\gamma| \leq D} \nu |t_\gamma| + \sum_{|\alpha| \leq k} d_\alpha (1 - X^{2\alpha}) = &- \sum_{|\gamma| \leq D} \nu |t_\gamma| - \sum_{\alpha : d_\alpha = 0} \nu + \sum_{\alpha : d_\alpha = 0} \nu (1 - X^{2\alpha}) + \sum_{\alpha : d_\alpha = 0} \nu X^{2\alpha} \\ &+ \sum_{|\alpha| : d_\alpha = 0} d_\alpha (1 - X^{2\alpha}) + \sum_{|\alpha| : d_\alpha > 0} d_\alpha (1 - X^{2\alpha}) . \end{aligned}$$

For $\alpha \in \mathbb{N}^n_k$, we define $c_\alpha := \nu$ if $d_\alpha = 0$ and $c_\alpha := d_\alpha$ otherwise, $a := \sum_{|\gamma| \leq D} |t_\gamma| + \sum_{\alpha : d_\alpha = 0} 1$, $\mathring{\sigma}_j := \sigma_j + \nu t_j$, for each $j = 1, \ldots, m$ and

$$\mathring{\sigma}_0 := f^\star_D - \nu a + \sigma_0 + \nu t_0 + \nu \sum_{(\alpha,\beta) \in \Delta(t)} \frac{|t_{\alpha+\beta}|}{2} (X^\alpha - \operatorname{sgn}(t_{\alpha+\beta}) X^\beta)^2 + \sum_{\alpha : d_\alpha = 0} \nu X^{2\alpha} .$$

So, there exists a sequence of positive numbers $(c_\alpha)_{|\alpha| \leq k}$ such that

$$f = \sum_{j=0}^m \mathring{\sigma}_j g_j + \sum_{|\alpha| \leq k} c_\alpha (1 - X^{2\alpha}) .$$

Now, let us select $\nu := \frac{1}{2^M}$ with M being the smallest positive integer such that $0 < \nu \leq \frac{f^\star_D}{2a}$. This implies the existence of a positive definite Gram matrix for $\mathring{\sigma}_0$, thus by Theorem 2.3, $\mathring{\sigma}_0 \in \mathring{\Sigma}[X]$. Similarly, for $1 \leq j \leq m$, $\mathring{\sigma}_j$ belongs to $\mathring{\Sigma}[X]$, which proves the second claim.

(iii) Let $N := M + 1$ and $\varepsilon := \frac{1}{2^N} = \frac{\nu}{2}$. One has

$$f - \varepsilon \sum_{|\alpha| \leq k} X^{2\alpha} = f - \varepsilon t_0 = \mathring{\sigma}_0 - \varepsilon t_0 + \sum_{j=1}^m \mathring{\sigma}_j g_j + \sum_{|\alpha| \leq k} c_\alpha (1 - X^{2\alpha}) .$$

Thus, $\sigma_0 + (\nu - \varepsilon) t_0 \in \mathring{\Sigma}[X]$. This implies that $\mathring{\sigma}_0 - \varepsilon t_0 \in \mathring{\Sigma}[X]$ and $f - \varepsilon t_0 \in \mathcal{Q}_D(S')$. Next, we derive a lower bound of $\frac{f^\star_D}{a}$. Since $t = \sum_{|\alpha| \leq k} X^{2\alpha} + \sum_{j=1}^m g_j \sum_{|\alpha| \leq k - w_j} X^{2\alpha}$, one has $\sum_{|\gamma| \leq D} |t_\gamma| \leq 2^\tau (m+1) \binom{n+D}{n}$. This implies that

$$a \leq 2^\tau (m+1) \binom{n+D}{n} + \binom{n+k}{k} \leq 2^\tau (m+2) \binom{n+D}{n} .$$

Recall that $\frac{f^\star}{2} \leq f^\star_D$, implying

$$\frac{f^\star_D}{a} \geq \frac{f^\star}{2^{\tau+1}(m+2)\binom{n+D}{n}} \geq \frac{1}{(m+2) 2^{\tau d^{\mathcal{O}(n)}} D^n} ,$$

where the last inequality follows from Theorem 5.4. Let us now give an upper bound of $\log_2 D$. First, note that for all $\alpha \in \mathbb{N}^n$, $\frac{|\alpha|!}{\alpha_1! \cdots \alpha_n!} \geq 1$, thus $\|f\| \leq 2^\tau$. Since D is the smallest even integer larger than $\underline{D}$, one has

$$\log_2 D \leq 1 + \log_2 \underline{D} \leq 1 + \log \chi_S + (12 d^3 n^{2d} 2^\tau 2^{\tau d^{\mathcal{O}(n)}})^{\chi_S} .$$

Next, since N is the smallest integer such that $\varepsilon = \frac{1}{2^N} = \frac{\nu}{2} \leq \frac{f^\star_D}{2a}$, it is enough to take

$$N \leq 1 + \log_2(m+2) + \tau d^{\mathcal{O}(n)} + n \log_2 D \leq 2^{\tau d^{n C_S}}$$

for some real $C_S > 0$ depending on S, the desired result. □

Algorithm 4 `Putinarsos`.

Input: $f \in \mathbb{Z}[X]$, $S := \{\boldsymbol{x} \in \mathbb{R}^n : g_1(\boldsymbol{x}) \geq 0, \ldots, g_m(\boldsymbol{x}) \geq 0\}$ with $g_1, \ldots, g_m \in \mathbb{Z}[X]$, positive $\varepsilon \in \mathbb{Q}$, precision parameters $\delta, R \in \mathbb{N}$ for the SDP solver , precision $\delta_c \in \mathbb{N}$ for the Cholesky's decomposition

Output: lists $\texttt{c\_list}_0, \ldots, \texttt{c\_list}_m, \texttt{c\_alpha}$ of numbers in $\mathbb{Q}$ and lists $\texttt{s\_list}_0, \ldots, \texttt{s\_list}_m$ of polynomials in $\mathbb{Q}[X]$

$k := \max\{\lceil d/2 \rceil, w_1, \ldots, w_m\}$, $D := 2k$, $g_0 := 1$
while $f \notin \mathcal{Q}_D(S)$ **do** $k := k + 1$, $D := D + 2$
done
$P := \mathbb{N}^n_D$, $S' := \{\boldsymbol{x} \in S : 1 - \boldsymbol{x}^{2\alpha} \geq 0\,, \forall \alpha \in \mathbb{N}^n_k\}$
$t := \sum_{\alpha \in P/2} X^{2\alpha}$, $f_\varepsilon := f - \varepsilon t$
while $f_\varepsilon \notin \mathcal{Q}_D(S')$ **do** $\varepsilon := \frac{\varepsilon}{2}$, $f_\varepsilon := f - \varepsilon t$
done
ok := false
while not ok **do**
  $[\tilde{G}_0, \ldots, \tilde{G}_m, \tilde{\lambda}_0, \ldots, \tilde{\lambda}_m, (\tilde{c}_\alpha)_{|\alpha| \leq k}], := \texttt{sdp}(f_\varepsilon, \delta, R, S')$
  $\texttt{c\_alpha} := (\tilde{c}_\alpha)_{|\alpha| \leq k}$
  for $j \in \{0, \ldots, m\}$ **do**
    $(s_{1j}, \ldots, s_{r_j j}) := \texttt{cholesky}(\tilde{G}_j, \tilde{\lambda}_j, \delta_c)$, $\tilde{\sigma}_j := \sum_{i=1}^{r_j} s_{ij}^2$
    $\texttt{c\_list}_j := [1, \ldots, 1]$, $\texttt{s\_list}_j := [s_{1j}, \ldots, s_{r_j j}]$
  done
  $u := f_\varepsilon - \sum_{j=0}^{m} \tilde{\sigma}_j\, g_j - \sum_{|\alpha| \leq k} \tilde{c}_\alpha (1 - X^{2\alpha})$
  for $\alpha \in P/2$ **do** $\varepsilon_\alpha := \varepsilon$
  done
  $\texttt{c\_list}, \texttt{s\_list}, (\varepsilon_\alpha) := \texttt{absorb}(u, P, (\varepsilon_\alpha), \texttt{c\_list}, \texttt{s\_list})$
  if $\min_{\alpha \in P/2}\{\varepsilon_\alpha\} \geq 0$ **then** ok := true
  else $\delta := 2\delta$, $R := 2R$, $\delta_c := 2\delta_c$
  end
done
for $\alpha \in P/2$ **do**
  $\texttt{c\_list}_0 := \texttt{c\_list}_0 \cup \{\varepsilon_\alpha\}$, $\texttt{s\_list}_0 := \texttt{s\_list}_0 \cup \{\boldsymbol{x}^\alpha\}$
done
return $\texttt{c\_list}_0, \ldots, \texttt{c\_list}_m, \texttt{c\_alpha}, \texttt{s\_list}_0, \ldots, \texttt{s\_list}_m$

We can now present Algorithm `Putinarsos`. For $f \in \mathbb{Z}[X]$ positive over a basic compact semi-algebraic set S satisfying Assumption 5.1, the first loop outputs the smallest positive integer $D = 2k$ such that $f \in \mathcal{Q}_D(S)$. Then the procedure is similar to `intsos`. As for the first loop of `intsos`, the loop from line 6 to line 7 allows to obtain a perturbed polynomial $f_\varepsilon \in \mathcal{Q}_D(S')$, with $S' := \{\boldsymbol{x} \in S : 1 - \boldsymbol{x}^{2\alpha} \geq 0\,, \forall \alpha \in \mathbb{N}^n_k\}$. Then one solves SDP (4) with the `sdp` procedure and performs Cholesky's decomposition to obtain an approximate Putinar's representation of $f_\varepsilon = f - \varepsilon t$ and a remainder u. Next, we apply the `absorb` subroutine as in `intsos`. The rationale is that with large enough precision parameters for the procedures `sdp` and `cholesky`, one finds an exact weighted SOS decomposition of $u + \varepsilon t$, which yields in turn an exact Putinar's representation of f in $\mathcal{Q}_D(S')$ with rational coefficients.

Example 3. Let us apply `Putinarsos` to $f = -X_1^2 - 2X_1X_2 - 2X_2^2 + 6$, $S := \{(x_1, x_2) \in \mathbb{R}^2 : 1 - x_1^2 \geq 0, 1 - x_2^2 \geq 0\}$ and the same precision parameters as in Example 1. The first and second loop yield $D = 2$ and $\varepsilon = 1$. After running `absorb`, we obtain the exact Putinar's representation $f = \frac{23853407}{292204836} + \frac{23}{49}X_1^2 + \frac{130657269}{291009481}X_2^2 + \frac{1}{2442^2} + (X_1 - X_2)^2 + (\frac{X_2}{2437})^2 + (\frac{11}{7})^2(1 - X_1^2) + (\frac{13}{7})^2(1 - X_2^2)$.

Theorem 5.6. *We use the notation and assumptions introduced above. For some $C_S > 0$ depending on S, there exist ε, δ, R, δ_c and $D = 2k$ of bit sizes less than $\mathcal{O}\left(2^{\tau d^{nC_S}}\right)$ for which `Putinarsos`$(f, S, \varepsilon, \delta, R, \delta_c)$ terminates and outputs an exact Putinar's representation with rational coefficients of $f \in \mathcal{Q}(S')$, with $S' := \{\boldsymbol{x} \in S : 1 - \boldsymbol{x}^{2\alpha} \geq 0\,, \forall \alpha \in \mathbb{N}^n_k\}$. The maximum bit size of these coefficients is bounded by $\mathcal{O}\left(2^{\tau d^{nC_S}}\right)$ and the procedure runs in boolean time $\mathcal{O}\left(2^{2^{\tau d^{nC_S}}}\right)$.*

Proof. The loops going from line 2 to line 3 and from line 6 to line 7 always terminate as respective consequences of Theorem 5.5 (i) and Theorem 5.5 (iii) with $D \leq 2^{\tau d^{nC_S}}$, $\varepsilon = \frac{1}{2^N}$, $N \leq 2^{\tau d^{nC_S}}$, for some real $C_S > 0$ depending on S.

What remains to prove is similar to Proposition 3.4 and Theorem 3.5. Let ν, $\mathring{\sigma}_0, \ldots, \mathring{\sigma}_m, (c_\alpha)_{|\alpha| \leq k}$ be as in the proof of Theorem 5.5. Note that ν (resp. $\varepsilon - \nu$) is a lower bound of the smallest eigenvalues of any Gram matrix associated to $\mathring{\sigma}_j$ (resp. $\mathring{\sigma}_0$) for $1 \leq j \leq m$. In addition, $c_\alpha \geq \nu$ for all $\alpha \in \mathbb{N}^n_k$. When the `sdp` procedure at line 10 succeeds, the matrix $\tilde{G}_j$ is an approximate Gram matrix of the polynomial $\mathring{\sigma}_j$ with $\tilde{G}_j \succeq 2^\delta I$, $\sqrt{\operatorname{Tr}(\tilde{G}_j^2)} \leq R$, we obtain a positive rational approximation $\tilde{\lambda}_j \geq 2^{-\delta}$ of the smallest eigenvalue of $\tilde{G}_j$, $\tilde{c_\alpha}$ is a rational approximation of c_α with $\tilde{c_\alpha} \geq 2^{-\delta}$, and $\tilde{c_\alpha} \leq R$, for all $j = 0, \ldots, m$ and $\alpha \in \mathbb{N}^n_k$. This happens when $2^{-\delta} \leq \varepsilon$ and $2^{-\delta} \leq \varepsilon - \nu$, thus for $\delta = \mathcal{O}\left(2^{\tau d^{nC_S}}\right)$. As in the proof of Proposition 3.3, we derive a similar upper bound of R by a symmetric argument while considering a Putinar representation of $\overline{f}_D - f \in \mathcal{Q}_D(S')$, where $\overline{f}_D := \inf\{b : b - f \in \mathcal{Q}_D(S)\}$. As for the second loop of Algorithm `intsos`, the third loop of `Putinarsos` terminates when the remainder polynomial $u = f_\varepsilon - \sum_{j=0}^m \tilde{\sigma_j}\, g_j - \sum_{|\alpha| \leq k} \tilde{c}_\alpha (1 - X^{2\alpha})$ satisfies $|u_\gamma| \leq \frac{\varepsilon}{r_0}$, where $r_0 = \binom{n+k}{n}$ is the size of $P/2 = \mathbb{N}^n_k$. As in the proof of Proposition 3.4, one can show that this happens when δ and δ_c are large enough.

To bound the precision δ_c required for Cholesky's decomposition, we do as in the proof of Proposition 3.4. The difference now is that there are $m + \binom{n+k}{k} = m + r_0$ additional terms in each equality constraint of SDP (4), by comparison with SDP (1). Thus, we need to bound for all $j = 1 \ldots, m$, $\alpha \in \mathbb{N}^n_k$ and $\gamma \in \operatorname{supp}(u)$ each term $|\operatorname{Tr}(\tilde{G}_j C_{j\gamma}) - (g_j \tilde{\sigma})_\gamma|$ related to the constraint $g_j \geq 0$ as well as each term (omitted for conciseness) involving $\tilde{c}_\alpha$ related to the constraint $1 - X^{2\alpha} \geq 0$. By using the fact that

$$\operatorname{Tr}(\tilde{G}_j C_{j\gamma}) = \sum_\delta g_{j\delta} \sum_{\alpha+\beta+\delta=\gamma} \tilde{G}_{j\alpha,\beta}\,,$$

we obtain

$$|\operatorname{Tr}(\tilde{G}_j C_{j\gamma}) - (g_j \tilde{\sigma})_\gamma| \leq \sum_\delta |g_{j\delta}| \frac{\sqrt{r_j}(r_j + 1)2^{-\delta_c} R}{1 - (r_j + 1)2^{-\delta_c}}\,,$$

where r_j is the size of $\tilde{G}_j$. Note that the size r_0 of the matrix $\tilde{G}_0$ satisfies $r_0 \geq r_j$ for all $j = 1, \ldots, m$. In addition, $\deg g_j \leq D$ implies

$$\sum_\delta |g_{j\delta}| \leq \binom{n + \deg g_j}{n} 2^\tau \leq \binom{n + D}{n} 2^\tau \leq D^n 2^{\tau+1}\,.$$

This yields an upper bound of $D^n 2^{\tau+1} \frac{\sqrt{r_0}(r_0+1)2^{-\delta_c} R}{1-(r_0+1)2^{-\delta_c}}$. We obtain a similar bound (omitted for conciseness) for each term involving $\tilde{c}_\alpha$.

Then, we take the smallest δ such that $2^{-\delta} \leq \frac{\epsilon}{2r_0}$ and the smallest δ_c such that

$$D^n 2^\tau \frac{\sqrt{r_0}(r_0+1)2^{-\delta_c} R}{1-(r_0+1)2^{-\delta_c}} \leq \frac{\varepsilon}{2r_0((m+1)+r_0)},$$

which ensures that `Putinarsos` terminates. One proves that the procedure outputs an exact Putinar's representation of $f \in \mathcal{Q}(S')$ with rational coefficients of maximum bit size bounded by $\mathcal{O}\left(2^{\tau d^{nC_S}}\right)$.

As in the proof of Theorem 3.5, let n_{sdp} be the sum of the sizes of the matrices involved in SDP (4) and m_{sdp} be the number of entries. Note that

$$n_{\text{sdp}} \leq (m+1)r_0 + r_0 \leq (m+2)\binom{n+D}{n}$$

and $m_{\text{sdp}} := \binom{n+D}{n}$. To bound the boolean running time, we consider the cost of solving SDP (4), which is performed in $\mathcal{O}\left(n_{\text{sdp}}^4 \log_2(2^\tau n_{\text{sdp}} R\, 2^\delta)\right)$ iterations of the ellipsoid method, where each iteration requires $\mathcal{O}\left(n_{\text{sdp}}^2(m_{\text{sdp}} + n_{\text{sdp}})\right)$ arithmetic operations over $\log_2(2^\tau n_{\text{sdp}} R\, 2^\delta)$-bit numbers. Since m_{sdp} is bounded by $\binom{n+D}{n} \leq 2D^n$ and $\log_2 D = \mathcal{O}\left(2^{\tau d^{nC_S}}\right)$, one has $m_{\text{sdp}} = \mathcal{O}\left(2^{2^{\tau d^{nC_S}}}\right)$. We obtain the same bound for n_{sdp}, which ends the proof. □

The complexity is polynomial in the degree D of the representation, often close in practice to the degrees of the involved polynomials, as shown in Section 6.

6 Practical experiments

We provide practical performance results for Algorithms `intsos`, `Polyasos` and `Putinarsos`. These are implemented in a library, called `multivsos`, written in Maple. More details about installation and benchmark execution are given on the two webpages dedicated to univariate[1] and multivariate[2] polynomials. This tool is available within the RAGlib Maple package[3]. All results were obtained on an Intel Core i7-5600U CPU (2.60 GHz) with 16Gb of RAM. We use the Maple `Convex` package[4] to compute Newton polytopes. Our subroutine `sdp` relies on the arbitrary-precision solver SDPA-GMP [30] and the `cholesky` procedure is implemented with the function `LUDecomposition` available within Maple. Most of the time is spent in the `sdp` procedure for all benchmarks.

In Table 1, we compare the performance of `multivsos` for nine univariate polynomials being positive over compact intervals. More details about these benchmarks are given in [11, Section 6] and [28, Section 5]. In this case, we use `Putinarsos`. The main difference is that we use SDP in `multivsos` instead of complex root isolation in `univsos2`. The results emphasize that `univsos2` performs better and provides more concise SOS certificates, especially for high degrees (see e.g. # 5). For # 3, we were not able to obtain a decomposition within a day of computation, as meant by the symbol − in the corresponding column entries. Large values of d and τ require more precision. The values of ε, δ and δ_c are respectively between 2^{-80} and 2^{-240}, 30 and 100, 200 and 2000.

Next, we compare the performance of `multivsos` with other tools in Table 2. The two first benchmarks are built from the polynomial $f = (X_1^2+1)^2 + (X_2^2+1)^2 + 2(X_1+X_2+1)^2 - 268849736/10^8$ from [26, Example 1], with $f_{12} := f^3$ and $f_{20} := f^5$. For these two benchmarks, we apply `intsos`. We use `Polyasos` to handle M_{20} (resp. M_{100}), obtained as in Example 2 by adding 2^{-20} (resp. 2^{-100}) to the positive coefficients of the Motzkin polynomial and r_i, which is a randomly generated positive definite quartic with i variables. We implemented in Maple the projection and rounding algorithm from [34] also

[1] `https://github.com/magronv/univsos`
[2] `https://github.com/magronv/multivsos`
[3] `http://www-polsys.lip6.fr/~safey/RAGLib/`
[4] `http://www-home.math.uwo.ca/~mfranz/convex`

Table 1: `multivsos` vs `univsos2` [28] for benchmarks from [11].

Id	d	τ (bits)	`multivsos`		`univsos2`	
			τ_1 (bits)	t_1 (s)	τ_2 (bits)	t_2 (s)
# 1	13	22 682	387 178	0.84	51 992	0.83
# 3	32	269 958	–	–	580 335	2.64
# 4	22	47 019	1 229 036	2.08	106 797	1.78
# 5	34	117 307	10 271 899	69.3	265 330	5.21
# 6	17	26 438	713 865	1.15	59 926	1.03
# 7	43	67 399	10 360 440	16.3	152 277	11.2
# 8	22	27 581	1 123 152	1.95	63 630	1.86
# 9	20	30 414	896 342	1.54	68 664	1.61
# 10	25	42 749	2 436 703	3.02	98 926	2.76

relying on SDP, denoted by `RoundProject`. For `multivsos`, the values of ε, δ and δ_c lie between 2^{-100} and 2^{-10}, 60 and 200, 10 and 60. We compare with `RAGLib` based on critical points and the `SamplePoints` procedure (abbreviated as `CAD`) based on CAD, both available in Maple. While these methods outperform the two SDP-based algorithms for examples with $n \leq 3$, they are less efficient for larger examples such as r_6^2 and suffer from a severe computational burden when $n \geq 8$. An additional drawback is that they do not provide non-negativity certificates. However, note that they can solve less restrictive problems, involving positive semidefinite forms or non-negative polynomials.

As shown in [25], SDP-based methods may provide exact certificates even in such cases and can be extended to rational functions. The algorithms we developed in this paper are unable to handle such cases. In most cases, `multivsos` is more efficient than `RoundProject` and outputs more concise representations. The reason is that `multivsos` performs approximate Cholesky's decompositions while `RoundProject` computes exact LDL^T decompositions of Gram matrices obtained after the two steps of rounding and projection. Note that we could not solve the examples of Table 2 with less precision.

Table 2: `multivsos` vs `RoundProject` [34] vs `RAGLib` vs `CAD` for n-variate polynomials of degree d (Polya).

Id	n	d	`multivsos`		`RoundProject`		`RAGLib`	`CAD`
			τ_1 (bits)	t_1 (s)	τ_2 (bits)	t_2 (s)	t_3 (s)	t_4 (s)
f_{12}	2	12	162 861	5.96	5 185 020	6.92	0.15	0.07
f_{20}	2	20	745 419	110.	78 949 497	141.	0.16	0.03
M_{20}	3	8	4 695	0.18	3 996	0.15	0.13	0.05
M_{100}	3	8	17 232	0.35	18 831	0.29	0.15	0.03
r_2	2	4	1 866	0.03	1 031	0.04	0.09	0.01
r_4	4	4	14 571	0.15	47 133	0.25	0.32	–
r_6	6	4	56 890	0.34	475 359	0.54	623.	–
r_8	8	4	157 583	0.96	2 251 511	1.41	–	–
r_{10}	10	4	344 347	2.45	8 374 082	4.59	–	–
r_6^2	6	8	1 283 982	13.8	146 103 466	106.	10.9	–

Finally, we compare the performance of `multivsos` (`Putinarsos`) on positive polynomials on basic compact semi-algebraic sets in Table 3. The first benchmark is from [26, Problem 4.6]. Each benchmark f_i comes from an inequality of the Flyspeck project [21]. The three last benchmarks are from [29]. The maximal degree of the polynomials involved in each system is denoted by d. We emphasize that the degree $D = 2k$ of each Putinar representation obtained in practice with `Putinarsos` is very close to d, which is in contrast with the theoretical complexity estimates obtained in Section 5. The values of ε, δ and δ_c lie between 2^{-30} and 2^{-10}, 60 and 200, 10 and 30. As for Table 2, `RAGLib` performs better for problems with $d \leq 3$ and $n \leq 4$. Larger problems (e.g. magnetism, f_{859}) are handled more efficiently with `multivsos` and `CAD` can only solve 3 benchmarks out of 10. We plan to extend the procedure `RoundProject` and the algorithm from [25] to the case of such constrained problems.

Table 3: `multivsos` vs `RAGLib` vs `CAD` for positive polynomials over basic compact semialgebraic sets (Putinar).

Id	n	d	`multivsos`			`RAGLib`	`CAD`
			k	τ_1 (bits)	t_1 (s)	t_2 (s)	t_3 (s)
p_{46}	2	4	3	21 723	0.83	0.15	0.81
f_{260}	6	3	2	114 642	2.72	0.12	–
f_{491}	6	3	2	108 359	9.65	0.01	0.05
f_{752}	6	2	2	10 204	0.26	0.07	–
f_{859}	6	7	4	6 355 724	303.	5896.	–
f_{863}	4	2	1	5 492	0.14	0.01	0.01
f_{884}	4	4	3	300 784	25.1	0.21	–
f_{890}	4	4	2	60 787	0.59	0.08	–
butcher	6	3	2	247 623	1.32	47.2	–
heart	8	4	2	618 847	2.94	0.54	–
magnetism	7	2	1	9 622	0.29	434.	–

References

[1] C. Bachoc and F. Vallentin. New upper bounds for kissing numbers from semidefinite programming. *Journal of the AMS*, 21(3):909–924, 2008.

[2] Z. Bai, J. Demmel, and A. McKenney. On Floating Point Errors in Cholesky. Technical report, 1989. (LAPACK Working Note 14).

[3] B. Bank, M. Giusti, J. Heintz, and G.-M. Mbakop. Polar varieties and efficient real elimination. *Mathematische Zeitschrift*, 238(1):115–144, 2001.

[4] B. Bank, M. Giusti, J. Heintz, and L.-M. Pardo. Generalized polar varieties: Geometry and algorithms. *Journal of complexity*, 2005.

[5] B. Bank, M. Giusti, J. Heintz, and M. Safey El Din. Intrinsic complexity estimates in polynomial optimization. *Journal of Complexity*, 30(4):430–443, 2014.

[6] B. Bank, M. Giusti, J. Heintz, M. Safey El Din, and É. Schost. On the geometry of polar varieties. *Applicable Algebra in Engineering, Communication and Computing*, 21(1):33–83, 2010.

[7] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa. The Quickhull Algorithm for Convex Hulls. *ACM Trans. Math. Softw.*, 22(4):469–483, 1996.

[8] S. Basu, R. Pollack, and M.-F. Roy. A new algorithm to find a point in every cell defined by a family of polynomials. In *Quantifier elimination and cylindrical algebraic decomposition*. Springer-Verlag, 1998.

[9] S. Basu, R. Pollack, and M.-F. Roy. *Algorithms in Real Algebraic Geometry (Algorithms and Computation in Mathematics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[10] L. Blum, F. Cucker, M. Shub, and S. Smale. *Complexity and real computation*. Springer Science & Business Media, 2012.

[11] S. Chevillard, J. Harrison, M. Joldes, and C. Lauter. Efficient and accurate computation of upper bounds of approximation errors. *Theoretical Computer Science*, 412(16):1523 – 1543, 2011.

[12] M. D. Choi, T. Y. Lam, and B. Reznick. Sums of squares of real polynomials. volume 58 of *Proc. Sympos. Pure Math.*, pages 103–126. Amer. Math. Soc., 1995.

[13] G. E Collins. Quantifier elimination for real closed fields by cylindrical algebraic decompostion. In *ATFL 2nd GI Conf. Kaiserslautern*, pages 134–183, 1975.

[14] E. de Klerk and F. Vallentin. On the Turing Model Complexity of Interior Point Methods for Semidefinite Programming. *SIAM Journal on Optimization*, 26(3):1944–1961, 2016.

[15] A. Greuet and M. Safey El Din. Probabilistic Algorithm for Polynomial Optimization over a Real Algebraic Set. *SIAM Journal on Optimization*, 24(3):1313–1343, 2014.

[16] A. Greuet, F. Guo, M. Safey El Din, and Lihong Zhi. Global optimization of polynomials restricted to a smooth variety using sums of squares. *Journal of Symbolic Computation*, 47(5):503 – 518, 2012.

[17] D. Grigoriev and N. Vorobjov. Solving systems of polynomials inequalities in subexponential time. *Journal of Symbolic Computation*, 5:37–64, 1988.

[18] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, second corrected edition edition, 1993.

[19] F. Guo, M. Safey El Din, and L. Zhi. Global optimization of polynomials using generalized critical values and sums of squares. In *Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation*, ISSAC '10, pages 107–114, New York, NY, USA, 2010. ACM.

[20] Q. Guo, M. Safey El Din, and L. Zhi. Computing rational solutions of linear matrix inequalities. In *Proceedings of the 38th International Symposium on Symbolic and Algebraic Computation*, pages 197–204. ACM, 2013.

[21] Thomas C. Hales. The flyspeck project, 2013.

[22] D. Henrion, S. Naldi, and M. Safey El Din. Exact Algorithms for Linear Matrix Inequalities. *SIAM Journal on Optimization*, 26(4):2512–2539, 2016.

[23] J.B. Lasserre, M. Laurent, B. Mourrain, P. Rostalski and P. TréBuchet. Moment Matrices, Border Bases and Real Radical Computation. *J. Symb. Computation.*, 51:63–85, 2013.

[24] G. Jeronimo and D. Perrucci. On the minimum of a positive polynomial over the standard simplex. *Journal of Symbolic Computation*, 45(4):434 – 442, 2010.

[25] E. Kaltofen, B. Li, Z. Yang, and L. Zhi. Exact certification of global optimality of approximate factorizations via rationalizing sums-of-squares with floating point scalars. In *Proceedings of the twenty-first international symposium on Symbolic and algebraic computation*, pages 155–164. ACM, 2008.

[26] J.-B. Lasserre. Global Optimization with Polynomials and the Problem of Moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001.

[27] M. Laurent. *Sums of squares, moment matrices and optimization over polynomials*. Springer, 2009.

[28] V. Magron, M. Safey El Din, and M. Schweighofer. Algorithms for Sums of Squares Decompositions of Non-negative Univariate Polynomials, 2017. Submitted.

[29] C. Muñoz and A. Narkawicz. Formalization of Bernstein Polynomials and Applications to Global Optimization. *J. Aut. Reasoning*, 51(2):151–196, 2013.

[30] M. Nakata. A numerical evaluation of highly accurate multiple-precision arithmetic version of semidefinite programming solver: SDPA-GMP, -QD and -DD. In *CACSD*, pages 29–34, 2010.

[31] J. Nie, K. Ranestad, and B. Sturmfels. The algebraic degree of semidefinite programming. *Mathematical Programming*, 122(2):379–405, 2010.

[32] J. Nie and M. Schweighofer. On the complexity of Putinar's Positivstellensatz. *Journal of Complexity*, 23(1):135 – 150, 2007.

[33] P. A. Parrilo. *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*. PhD thesis, California Inst. Tech., 2000.

[34] H. Peyrl and P.A. Parrilo. Computing sum of squares decompositions with rational coefficients. *Theoretical Computer Science*, 409(2):269–281, 2008.

[35] G. Pólya. Über positive Darstellung von Polynomen. *Naturforsch. Ges. Zürich*, 73:141–145, 1928.

[36] M. Putinar. Positive polynomials on compact semi-algebraic sets. *Indiana University Mathematics Journal*, 42(3):969–984, 1993.

[37] B. Reznick. Extremal PSD forms with few terms. *Duke Mathematical Journal*, 45(2):363–374, 1978.

[38] B. Reznick. Uniform denominators in Hilbert's seventeenth problem. *Mathematische Zeitschrift*, 220(1):75–97, Dec 1995.

[39] M. Safey El Din. Testing sign conditions on a multivariate polynomial and applications. *Mathematics in Computer Science*, 1(1):177–207, 2007.

[40] M. Safey El Din and É. Schost. Polar varieties and computation of one point in each connected component of a smooth real algebraic set. In *ISSAC'03*, pages 224–231. ACM, 2003.

[41] M. Safey El Din and L. Zhi. Computing Rational Points in Convex Semialgebraic Sets and Sum of Squares Decompositions. *SIAM J. on Optimization*, 20(6):2876–2889, September 2010.

[42] J. F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones, 1998.

[43] M. Yamashita, K. Fujisawa, K. Nakata, M. Nakata, M. Fukuda, K. Kobayashi, and K. Goto. A high-performance software package for semidefinite programs : SDPA7. Technical report, Dept. of Information Sciences, Tokyo Inst. Tech., 2010.