On Exact Reznick, Hilbert-Artin and Putinar's Representations

Victor Magron

CNRS; LAAS; 7 avenue du colonel Roche, F-31400 Toulouse; France

Mohab Safey El Din

Sorbonne Université, CNRS, Laboratoire d'Informatique de Paris 6 (UMR 7606), PolSys, Paris, France.

Abstract

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

We provide a hybrid numeric-symbolic algorithm computing *exact* rational SOS decompositions with rational coefficients for polynomials lying in the interior of the SOS cone. The first step of this algorithm computes an approximate SOS decomposition for a perturbation of the input polynomial with an arbitrary-precision SDP solver. Next, an exact SOS decomposition is obtained thanks to the perturbation terms and a compensation phenomenon. We prove that bit complexity estimates on output size and runtime are both polynomial in the degree of the input polynomial and singly exponential in the number of variables. Next, we apply this algorithm to compute exact Reznick, Hilbert-Artin's representation and Putinar's representations respectively for positive definite forms and positive polynomials over basic compact semi-algebraic sets. We also report on practical experiments done with the implementation of these algorithms and existing alternatives such as the critical point method and cylindrical algebraic decomposition.

Keywords: Real algebraic geometry, Semidefinite programming, sums of squares decomposition, Reznick's representation, Hilbert-Artin's representation, Putinar's representation, hybrid numeric-symbolic algorithm.

---

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 closed 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 co-NP hard (Blum et al., 2012). The Cylindrical Algebraic Decomposition algorithm due to Collins (1975) and Wütrich (1976) allows one to solve it in time

Email addresses: vmagron@laas.fr (Victor Magron), mohab.safey@lip6.fr (Mohab Safey El Din)

URL: https://homepages.laas.fr/vmagron (Victor Magron), http://www-polsys.lip6.fr/~safey (Mohab Safey El Din)

doubly exponential in n (and polynomial in d). This has been significantly improved, through the so-called critical point method, starting from Grigoriev and Vorobjov (1988) which culminates with Basu et al. (1998) to establish that this decision problem can be solved in time $((m+1)d)^{O(n)}$. These latter ones have been developed to obtain practically fast implementations which reflect the complexity gain (see e.g. Bank et al. (2001, 2005); Safey El Din and Schost (2003); Safey El Din (2007b); Bank et al. (2010); Guo et al. (2010); Bank et al. (2014); Greuet and Safey El Din (2014); Greuet et al. (2012)). These algorithms are "root finding" ones: they are designed to compute at least one point in each connected component of the set defined by $f < 0$. This is done by solving polynomial systems defining critical points of some well-chosen polynomial maps restricted to $f = -\varepsilon$ for ε small enough. Hence the complexity of these algorithms depends on the difficulty of solving these polynomial systems (which can be exponential in n as the Bézout bound on the number of their solutions is). Moreover, when f is non-negative, they return an empty list without a *certificate* that can be checked *a posteriori*. This paper focuses on the computation of such certificates under some favorable situations.

To compute certificates of non-negativity, an approach based on *sums of squares* (SOS) decompositions of polynomials (see Lasserre (2001) and P. A. Parrilo (2000)). Many positive polynomials are not sums of squares of polynomials following Blekherman (2006). However, some variants have been designed to make this approach more general ; see e.g. the survey by Laurent (2009) and references therein. In a nutshell, the core and initial 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 $1 \le i \le r$. Also f is non-negative over the semi-algebraic set S if it can be written as $s_1^2 + \cdots + s_r^2 + \sum_{j=1}^m \sigma_j g_j$ where σ_i is a sum of squares in $\mathbb{R}[X]$ for $1 \le j \le m$. It turns out that, thanks to the "Gram matrix method" (see e.g. Choi et al. (1995); Lasserre (2001); P. A. Parrilo (2000)), 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 $\le k$ 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 Henrion et al. (2016)), but the degrees of the algebraic extensions needed to encode exactly the solutions are prohibitive on large examples Nie et al. (2010). Besides, there exist fast numerical solvers for solving LMIs implemented in double precision, e.g. SeDuMi by Sturm (1998), SDPA by Yamashita et al. (2010) as well as arbitrary-precision solvers, e.g. SDPA-GMP by Nakata (2010), successfully applied in many contexts, including bounds for kissing numbers by Bachoc and Vallentin (2008) or computation of (real) radical ideals by J.B. Lasserre, M. Laurent, B. Mourrain, P. Rostalski and P. Trébuchet (2013).

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

This raises topical questions. The first one is how to use symbolic computation jointly 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 contributions in the univariate case by Chevillard et al. (2011); Magron et al. (2018) as well as in the multivariate case by Peyrl and Parrilo (2008);

Kaltofen et al. (2008). Note that Kaltofen et al. (2008, 2012) allow us to compute SOS with rational coefficients on some degenerate examples. Moreover, Kaltofen et al. (2012) allows to compute decompositions into sums of squares of rational fractions. Diophantine aspects are considered in Safey El Din and Zhi (2010); Guo et al. (2013). When an SOS decomposition exists with coefficients in a totally real Galois field, Hillar (2009) and Quarez (2010) provide bounds on the total number of squares.

In the univariate (un)-constrained case, given $f \in \mathbb{Q}[X]$, the algorithm by Chevillard et al. (2011) computes an exact (weighted) SOS decomposition $f = \sum_{i=1}^{t} c_i g_i^2$ with $c_i \in \mathbb{Q}$ and $g_i \in \mathbb{Q}[X]$. We call such SOS decompositions *weighted* because the coefficients c_i are considered outside the square, which helps when one wants to output data with rational coefficients only. To do that, the algorithm considers first 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 Peyrl and Parrilo (2008) 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 Kaltofen et al. (2008) to handle rational functions and in Guo et al. (2012) to derive certificates of impossibility for Hilbert-Artin representations of a given degree. In a recent work by Laplagne (2018), the author derives an algorithm based on facial reduction techniques to obtain exact rational decompositions for some sub-classes of non-negative polynomials lying in the border of the SOS cone. Among such degenerate sub-classes, he considers polynomials that can be written as sums of squares of polynomials with coefficients in an algebraic extension of $\mathbb{Q}$ of odd degree.

Main contributions. This work provides an algorithmic framework for computing exact rational weighted SOS decompositions in some favourable situations. 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 by Chevillard et al. (2011), the main idea is to perturb the input polynomial, then to obtain an approximate SOS decomposition (through some Gram matrix of the perturbation by solving an SDP problem), and to recover an exact decomposition using the perturbation terms.

In Section 4.1, we rely on `intsos` to compute decompositions of positive definite forms into SOS of rational functions, based on Reznick's representations, yielding an algorithm, called `Reznicksos`. In Section 4.2, we provide another algorithm, called `Hilbertsos`, to decompose non-negative polynomials into SOS of rational functions, under the assumption that the numerator belongs to the interior of the SOS cone. In Section 5, we rely on `intsos` to compute weighted SOS decompositions for polynomials positive over basic compact semi-algebraic sets, yielding the `Putinarsos` algorithm.

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 $\tau^2 d^{O(n)}$ and outputs SOS polynomials of bit size bounded by $\tau d^{O(n)}$. This also yields bit complexity analysis for Algorithm `Reznicksos` (see Section 4.1) 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 constants in the exponents are explicitely given in the sequel.

The three algorithms are implemented within a Maple procedure, called `multivsos`, integrated

in the RealCertify Maple library by Magron and Safey El Din (2018b). In Section 6, we provide benchmarks to evaluate the performance of multivsos. We compare it with previous approaches in Peyrl and Parrilo (2008) as well as with the more general methods based on the critical point method and Cylindrical Algebraic Decomposition.
This paper is the follow-up of our previous contribution (Magron and Safey El Din, 2018a), published in the proceedings of ISSAC'18. The main theoretical and practical novelties are the following: we provide explicit bounds for the bit complexity analyzes of our algorithms. In Section 3.4, we state formally the rounding-projection algorithm from Peyrl and Parrilo (2008), analyze its bit complexity and compare it with our algorithm intsos. We show that both algorithms have the same bit complexity. Another novelty is in Section 4.2, where we explain how to handle the sub-class of non-negative polynomials admitting an Hilbert-Artin's representation, for which the numerator belongs to the interior of the SOS cone. In Section 5.4, we state a constrained version of the rounding-projection algorithm. Again, this algorithm has the same bit complexity as Putinarsos. We have updated accordingly Section 6 by providing some related numerical comparisons. We also consider benchmarks involving non-negative polynomials which do not belong to the interior of the SOS cone.

Acknowledgments.. M. Safey El Din is supported by the ANR-18-CE33-0011 Sesame, the ANR-19-CE40-0018 De Rerum Natura projects funded by ANR (Agence Nationale de la Recherche), the ANR-19-CE48-0015 Ecarp project funded jointly by ANR and FWF, and the CAMiSAdo project funded by PGMO/FMJH. V. Magron benefited from the support of the FMJH Program PGMO (EPICS project) and from the support of EDF, Thales, Orange et Criteo, as well as from the Tremplin ERC Stg Grant ANR-18-ERC2-0004-01 (T-COPS project). Both authors are supported by European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement 813211 (POEMA).

2. Preliminaries

Let $\mathbb{Z}$ be the ring of integers and $X = (X_1, \ldots, X_n)$. 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), $\alpha \in \mathbb{N}^n_d$. When referring to univariate polynomials, we use the indeterminate E and we denote by $\mathbb{Z}[E]$ the set of univariate polynomials with integer coefficients. 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 will be interested in those polynomials which lie in $\mathbb{Z}[X] \cap \Sigma[X]$. For instance, the polynomial

$$f = 4X_1^4 + 4X_1^3 X_2 - 7X_1^2 X_2^2 - 2X_1 X_2^3 + 10X_2^4 = (2X_1X_2 + X_2^2)^2 + (2X_1^2 + X_1X_2 - 3X_2^2)^2$$

lies in $\mathbb{Z}[X] \cap \Sigma[X]$.

The complexity estimates in this paper rely on the bit complexity model. The bit size of an integer b is denoted by $\tau(b) := \lfloor \log_2(|b|) \rfloor + 1$ with $\tau(0) := 1$. For $f = \sum_{|\alpha| \leq d} f_\alpha X^\alpha \in \mathbb{Z}[X]$ of degree d, we denote $\|f\|_\infty := \max_{|\alpha| \leq d} |f_\alpha|$ 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 "$g(v) = O(h(v))$" 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* $C(f)$ is the convex hull of the vectors of exponents of monomials that occur in $f \in \mathbb{R}[X]$. For the above example, $C(f) = \{(4,0), (3,1), (2,2), (1,3), (0,4)\}$. 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*). For a given Newton polytope P, let $\Sigma_P[X]$ be the convex cone of sums of squares whose Newton polytope is contained in P. Since the Newton polytope P is often clear from the context, we suppress the index P.

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

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

where $\mathrm{Tr}(M)$ (for a given matrix M) denotes the trace of M, 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 1. *(Lasserre, 2001, Theorem 3.2) Let $f \in \mathbb{R}[X]$ of degree $d = 2k$ and global infimum $f^\star := \inf_{\boldsymbol{x} \in \mathbb{R}^n} f(\boldsymbol{x})$. 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 $i = 1, \dots, r$. Then $f - f^\star = \sum_{i=1}^r \lambda_i q_i^2$.*

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

Theorem 2. *(Reznick, 1978, Theorem 1) Let $f \in \Sigma[X]$ with $f = \sum_{i=1}^r s_i^2$, $P := C(f)$ and $Q := P/2 \cap \mathbb{N}^n$. Then for all $i = 1, \dots, r$, $C(s_i) \subseteq Q$.*

Given $f \in \mathbb{R}[X]$, Theorem 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 (Choi et al., 1995, Proposition 5.5):

Theorem 3. *Let $f \in \mathbb{Z}[X]$ with $P := C(f)$, $Q := P/2 \cap \mathbb{N}^n$ and v_k be the vector of all monomials with support in Q. Then $f \in \mathring{\Sigma}[X]$ if and only if there exists a positive definite matrix G such that $f = v_k^T G v_k$.*

Eventually, we will rely on the following bound for the roots of polynomials with integer coefficients:

Lemma 4. *(Mignotte, 1992, Theorem 4.2 (ii)) Let $f \in \mathbb{Z}[E]$ of degree d, with coefficient bit size bounded from above by τ. If $f(e) = 0$ and $e \neq 0$, then $\frac{1}{2^\tau + 1} \leq |e| \leq 2^\tau + 1$.*

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 $\mathbb{Z}[X] \cap \mathring{\Sigma}[X]$. This algorithm relies on perturbations of such polynomials. We first establish the following preliminary result.

Proposition 5. *Let $f \in \mathbb{Z}[X] \cap \mathring{\Sigma}[X]$ of degree $d = 2k$, with $\tau = \tau(f)$, $P = C(f)$ and $Q := P/2 \cap \mathbb{N}^n$. Then, there exists $N \in \mathbb{N} - \{0\}$ such that for $\varepsilon := \frac{1}{2^N}$, $f - \varepsilon \sum_{\alpha \in Q} X^{2\alpha} \in \mathring{\Sigma}[X]$, with $N \leq \tau(\varepsilon) \leq O(\tau \cdot (4d+2)^{3n+3})$.*

The proof of this result relies on the following technical statement whose proof is postponed in the Appendix.

Proposition 6. *Let $g \in \mathbb{Z}[X]$ of degree d and let $\tau = \tau(g)$. Assume that the algebraic set $V(g) \subset \mathbb{C}^n$ defined by $g = 0$ is smooth. Then, there exists a polynomial $w \in \mathbb{Z}[X_1]$ of degree $\leq d^n$ with coefficients of bit size $\leq \tau \cdot (4d+2)^{3n}$ such that its set of real roots contains the critical values of the restriction of the projection on the X_1-axis to $V(g)$.*

Proof of Proposition 5. Let v_k be the vector of all monomials X^α, with α in Q. Note that each monomial in v_k has degree $\leq k$ and that $v_k^T v_k = \sum_{\alpha \in Q} X^{2\alpha}$. Since $f \in \mathring{\Sigma}[X]$, there exists by Theorem 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

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

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

For the second claim, let us consider the algebraic set V defined by

$$f(X) - E = \frac{\partial f}{\partial X_1} = \cdots = \frac{\partial f}{\partial X_n} \,.$$

Let us note A the projection of $V \cap \mathbb{R}^n$ on the E-axis. Note that A contains the minimizers of $f(\boldsymbol{x})$.

Note that the algebraic set defined by $f - E = 0$ is smooth (since it is a graph). Applying Proposition 6 to $f - E$, we deduce that there exists a polynomial in $\mathbb{Z}[E]$ of degree less than d^{n+1} with coefficients of bit size less than $\tau \cdot (4d+2)^{3n+3}$ such that its set of real roots contains A. By Lemma 4, it follows that it is enough to select $N \leq O(\tau \cdot (4d+2)^{3n+3})$.

□

The following can be found in (Bai et al., 1989, Lemma 2.1) and (Bai et al., 1989, Theorem 3.2).

Proposition 7. *Let $G \succ 0$ be a matrix with rational entries indexed on $\mathbb{N}^n_r$. Let L be the factor of G computed using Cholesky's decomposition with finite precision δ_c. Then $LL^T = G + F$ where*

$$|F_{\alpha,\beta}| \leq \frac{(r+1)2^{-\delta_c} |G_{\alpha,\alpha}\, G_{\beta,\beta}|^{\frac{1}{2}}}{1 - (r+1)2^{-\delta_c}} \,. \tag{2}$$

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

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

Cholesky's decomposition returns a rational nonsingular factor L.

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 $\mathbb{Q}[X]$

$P := C(f)$, $Q := P/2 \cap \mathbb{N}^n$
$t := \sum_{\alpha \in Q} X^{2\alpha}$, $f_\varepsilon := f - \varepsilon t$
while $f_\varepsilon \notin \mathring{\Sigma}[X]$ **do** $\varepsilon := \frac{\varepsilon}{2}$, $f_\varepsilon := f - \varepsilon t$
done
ok := false
while not ok **do**
  $(\tilde{G}, \tilde{\lambda})$:= `sdp`$(f_\varepsilon, \delta, R)$
  $(s_1, \ldots, s_r)$:= `cholesky`$(\tilde{G}, \tilde{\lambda}, \delta_c)$ ▷ $f_\varepsilon \simeq \sum_{i=1}^r s_i^2$
  $u := f_\varepsilon - \sum_{i=1}^r s_i^2$
  `c_list` := $[1, \ldots, 1]$, `s_list` := $[s_1, \ldots, s_r]$
  for $\alpha \in Q$ **do** $\varepsilon_\alpha := \varepsilon$
  done
  `c_list`, `s_list`, (ε_α) := `absorb`$(u, Q, (\varepsilon_\alpha)$, `c_list`, `s_list`)
  if $\min_{\alpha \in Q}\{\varepsilon_\alpha\} \geq 0$ **then** ok := true
  else $\delta := 2\delta$, $R := 2R$, $\delta_c := 2\delta_c$
  end
done
for $\alpha \in Q$ **do** `c_list` := `c_list` $\cup \{\varepsilon_\alpha\}$, `s_list` := `s_list` $\cup \{X^\alpha\}$
done
return `c_list`, `s_list`

Algorithm 2 `absorb`

Input: $u \in \mathbb{Q}[X]$, multi-index set Q, 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 $\mathbb{Q}[X]$

for $\gamma \in \operatorname{supp}(u)$ **do**
  if $\gamma \in (2\mathbb{N})^n$ **then** $\alpha := \frac{\gamma}{2}$, $\varepsilon_\alpha := \varepsilon_\alpha + u_\gamma$
  else
    Find $\alpha, \beta \in Q$ such that $\gamma = \alpha + \beta$
    $\varepsilon_\alpha := \varepsilon_\alpha - \frac{|u_\gamma|}{2}$, $\varepsilon_\beta := \varepsilon_\beta - \frac{|u_\gamma|}{2}$
    `c_list` := `c_list` $\cup \{\frac{|u_\gamma|}{2}\}$
    `s_list` := `s_list` $\cup \{X^\alpha + \operatorname{sgn}(u_\gamma) X^\beta\}$
  end
done

3.1. *Algorithm* `intsos`

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

Given $f \in \mathbb{Z}[X]$ of degree $d = 2k$, one first computes its Newton polytope $P := C(f)$ (see line 1) and $Q := P/2 \cap \mathbb{N}^n$ using standard algorithms such as quickhull by Barber et al. (1996). The loop going from line 3 to line 4 finds a positive $\varepsilon \in \mathbb{Q}$ such that the perturbed polynomial $f_\varepsilon := f - \varepsilon \sum_{\alpha \in Q} X^{2\alpha}$ is also in $\mathring{\Sigma}[X]$. This is done thanks to any external *oracle* deciding the non-negativity of a polynomial. Even if this oracle is able to *decide* non-negativity, we would like to emphasize that our algorithm outputs an SOS certificate in order to *certify* the non-negativity of the input. In practice, we often choose the value of ε while relying on a heuristic technique rather than this external oracle, for the sake of efficiency (see Section 6 for more details).

If $f \in \mathbb{Z}[X] \cap \mathring{\Sigma}[X]$, then the set $\{e \in \mathbb{R}^{>0} : \forall \boldsymbol{x} \in \mathbb{R}^n, f(\boldsymbol{x}) - e \sum_{\alpha \in Q} \boldsymbol{x}^{2\alpha} \geq 0\}$ is non empty (see the proof of Proposition 5). If the oracle asserts that $\boldsymbol{x} \mapsto f(\boldsymbol{x}) - e \sum_{\alpha \in Q} \boldsymbol{x}^{2\alpha}$ is non-negative on $\mathbb{R}^n$, then e belongs to this set and it is enough to select $\varepsilon = e/2$ to ensure that $f_\varepsilon := f - \varepsilon \sum_{\alpha \in Q} X^{2\alpha} \in \mathring{\Sigma}[X]$.

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 order to analyse the complexity of the procedure (see Remark 8), we assume that `sdp` relies on the ellipsoid algorithm by Grötschel et al. (1993).

Remark 8. *In de Klerk and Vallentin (2016), the authors analyze the complexity of the short step, primal interior point method, used in SDP solvers. Within fixed accuracy, they obtain a polynomial complexity, as for the ellipsoid method, but the exact value of the exponents is not provided.*

Also, in practice, we use an arbitrary-precision SDP solver implemented with an interior-point method.

SDP problems are solved with this latter algorithm in polynomial-time within a given accuracy 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 integers 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 verifying that the inequality (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 size r vector of all monomials X^α with α belonging to Q. 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 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 $u = f - \varepsilon \sum_{\alpha \in Q} X^{2\alpha} - \sum_{i=1}^r s_i^2$, with $s_i \in \mathbb{Q}[X]$.

After these steps, the algorithm starts to perform symbolic computation with the `absorb` subroutine at line 13. The loop from `absorb` is designed to obtain an exact weigthed SOS decomposition of $\varepsilon t + u = \varepsilon \sum_{\alpha \in Q} X^{2\alpha} + \sum_\gamma u_\gamma X^\gamma$, 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 $\alpha, \beta \in Q$. In the former case (line 2), one has

$$\varepsilon X^{2\alpha} + u_\gamma X^{2\alpha} = (\varepsilon + u_\gamma) X^{2\alpha} \,.$$

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 Magron et al. (2018), the authors and Schweighofer formalized and analyzed an algorithm called `univsos2`, initially provided in Chevillard et al. (2011). 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 9. *We apply Algorithm* `intsos` *on*

$$f = 4X_1^4 + 4X_1^3 X_2 - 7X_1^2 X_2^2 - 2X_1 X_2^3 + 10 X_2^4,$$

with $\varepsilon = 1$, $\delta = R = 60$ *and* $\delta_c = 10$. *Then*

$$Q := C(f)/2 \cap \mathbb{N}^n = \{(2,0), (1,1), (0,2)\}$$

(line 1). The loop from line 3 to line 4 ends and we get $f - \varepsilon t = f - (X_1^4 + X_1^2 X_2^2 + X_2^2) \in \mathring{\Sigma}[X]$. *The* `sdp` *(line 7) and* `cholesky` *(line 8) procedures yield*

$$s_1 = 2X_1^2 + X_1 X_2 - \frac{8}{3} X_2^2, \quad s_2 = \frac{4}{3} X_1 X_2 + \frac{3}{2} X_2^2 \quad \textit{and} \quad s_3 = \frac{2}{7} X_2^2.$$

The remainder polynomial is $u = f - \varepsilon t - s_1^2 - s_2^2 - s_3^2 = -X_1^4 - \frac{1}{9} X_1^2 X_2^2 - \frac{2}{3} X_1 X_2^3 - \frac{781}{1764} X_2^4$.

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,*

$$\varepsilon (X_1^2 X_2^2 + X_2^4) - \frac{2}{3} X_1 X_2^3 = \frac{1}{3} (X_1 X_2 - X_2^2)^2 + (\varepsilon - \frac{1}{3})(X_1^2 X_2^2 + X_2^4).$$

In the polynomial $\varepsilon t + u$, *the coefficient of* $X_1^2 X_2^2$ *is* $\varepsilon_{(1,1)} = \varepsilon - \frac{1}{3} - \frac{1}{9} = \frac{5}{9}$ *and the coefficient of* X_4^4 *is* $\varepsilon_{(0,2)} = \varepsilon - \frac{1}{3} - \frac{781}{1764} = \frac{395}{1764}$.

Eventually, we obtain the weighted rational SOS decomposition:

$$\begin{aligned} 4X_1^4 + 4X_1^3 X_2 - 7X_1^2 X_2^2 - 2X_1 X_2^3 + 10X_2^4 = &\frac{1}{3}(X_1 X_2 - X_2^2)^2 + \frac{5}{9}(X_1 X_2)^2 + \frac{395}{1764} X_2^4 \\ &+ (2X_1^2 + X_1 X_2 - \frac{8}{3} X_2^2)^2 + (\frac{4}{3} X_1 X_2 + \frac{3}{2} X_2^2)^2 + (\frac{2}{7} X_2^2)^2 \,. \end{aligned}$$

3.2. Correctness and bit size of the output

Let $f \in \mathbb{Z}[X] \cap \mathring{\Sigma}[X]$ of degree $d = 2k$, $\tau := \tau(f)$ and $Q := C(f)/2 \cap \mathbb{N}^n$.

Proposition 10. *Let G be a positive definite Gram matrix associated to f and take $0 < \varepsilon \in \mathbb{Q}$ as in Proposition 5 so that $f_\varepsilon = f - \varepsilon \sum_{\alpha \in Q} X^{2\alpha} \in \mathring{\Sigma}[X]$. 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 $\sqrt{\mathrm{Tr}\,((G - \varepsilon I)^2)} \leq R$. Also, the maximal bit sizes of δ and R are upper bounded by $O(\tau \cdot (4d + 2)^{3n+3})$ and $O(\tau \cdot (4d + 2)^{4n+3})$, respectively.*

Proof. Let λ be the smallest eigenvalue of G. By Proposition 5, $G \succeq \varepsilon I$ for $\varepsilon = \frac{1}{2^N} \leq \frac{\lambda}{2}$ with $N \leq O(\tau \cdot (4d + 2)^{3n+3})$. By defining $\delta := N+1$, $2^{-\delta} = \frac{1}{2^{N+1}} \leq \frac{\lambda}{4} < \frac{\lambda}{2}$, yielding $G - \varepsilon \succeq \frac{\lambda}{2} I \succeq 2^{-\delta} I$. As $N \leq O(\tau \cdot (4d + 2)^{3n+3})$, one has $\delta \leq O(\tau \cdot (4d + 2)^{3n+3})$.

As in the proof of Proposition 5, we consider the largest eigenvalue λ' of the Gram matrix G of f and prove that the set $\{e' \in \mathbb{R} : \forall \boldsymbol{x} \in \mathbb{R}^n, -f(\boldsymbol{x}) + e' \sum_{\alpha \in Q} \boldsymbol{x}^{2\alpha} \geq 0\}$ is not empty. We apply again Proposition 6 as in the proof of Proposition 5 to establish that this set contains an interval $]0, \frac{1}{2^N}[$ with $N \leq O(\tau \cdot (4d + 2)^{3n+3})$. This allows in turn to obtain a rational upper bound ε' of λ' with bit size $O(\tau \cdot (4d + 2)^{3n+3}))$. 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 $\sqrt{\mathrm{Tr}\,((G - \varepsilon I)^2)} \leq \sqrt{\mathrm{Tr}\,(G^2)} \leq d^{\frac{n}{2}} \varepsilon' = O(\tau \cdot (4d + 2)^{4n+3})$. □

Proposition 11. *Let f be as above. When applying Algorithm* `intsos` *to f, the procedure always terminates and outputs a weighted SOS decompositon of f with rational coefficients. The maximum bit size of the coefficients involved in this SOS decomposition is upper bounded by $O(\tau \cdot (4d + 2)^{4n+3})$.*

Proof. Let us first consider the loop of Algorithm `intsos` defined from line 3 to line 4. From Proposition 5, this loop terminates when $f_\varepsilon \in \mathring{\Sigma}[X]$ for $\varepsilon = \frac{1}{2^N}$ and $N \leq O(\tau \cdot (4d + 2)^{3n+3})$.

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 $\mathrm{Tr}\,(\tilde{G}^2) \leq R^2$. From Proposition 10, this procedure succeeds for large enough values of δ and R of bit size upper bounded by $O(\tau \cdot (4d + 2)^{4n+3})$. In this case, we obtain a positive rational approximation $\tilde{\lambda} \geq 2^{-\delta}$ of the smallest eigenvalue of $\tilde{G}$.

Then the Cholesky 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 $\alpha \in Q$, $\varepsilon_\alpha \geq 0$. This condition is fulfilled when for all $\alpha \in Q$, $\varepsilon - \sum_{\beta \in Q} |u_{\alpha+\beta}|/2 + u_\alpha \geq 0$. This latter condition holds when for all $\gamma \in \mathrm{supp}(u)$, $|u_\gamma| \leq \frac{\varepsilon}{r}$.

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 $\gamma \in \mathrm{supp}(u)$, $u_\gamma = f_\gamma - \varepsilon_\gamma - (\sum_{i=1}^r s_i^2)_\gamma$, 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 $\gamma \in \operatorname{supp}(f)$, $\tilde{G} \succeq 2^{-\delta} I$ and

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

In addition, it follows from (2) that the approximated Cholesky decomposition LL^T of $\tilde{G}$ performed at precision δ satisfies $LL^T = \tilde{G} + F$ with

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

for all $\alpha, \beta \in Q$. Moreover, by using Cauchy-Schwartz inequality, one has

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

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

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

where the first inequality comes again from Cauchy-Schwartz inequality.

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

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

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 $\gamma \in \operatorname{supp}(u)$, 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

$$\Delta(u) := \{(\alpha, \beta) : \alpha + \beta \in \operatorname{supp}(u), \alpha, \beta \in Q, \alpha \neq \beta\} .$$

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 $\{\varepsilon_\alpha : \alpha \in Q\}$. The second output `s_list` of Algorithm `intsos` is a list of polynomials containing the list $[s_1, \dots, s_r]$, the list $\{X^\alpha + \operatorname{sgn}(u_{\alpha+\beta})X^\beta : (\alpha,\beta) \in \Delta(u)\}$ and the list $\{X^\alpha : \alpha \in Q\}$. From the output, we obtain the following weigthed SOS decomposition

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

Now, we bound the bit size of the coefficients. Since $r \leq \binom{n+k}{n} \leq d^n$ and $N \leq O(\tau \cdot (4d+2)^{3n+3})$, one has $\delta \leq O(\tau \cdot (4d+2)^{3n+3})$. Similarly, $\delta_c \leq O(\tau \cdot (4d+2)^{4n+3})$. This bounds also the maximal bit size of the coefficients involved in the approximate decomposition $\sum_{i=1}^r s_i^2$ as well as the coefficients of u. In the worst case, the coefficient ε_α involved in the exact SOS decomposition is equal to $\varepsilon - \sum_{\beta\in Q} |u_{\alpha+\beta}|/2 + u_\alpha$ for some $\alpha \in Q$. Using again that the cardinal r of Q is less than $\binom{n+k}{n} \leq d^n$, we obtain a maximum bit size upper bounded by $O(\tau \cdot (4d+2)^{3n+3})$. □

3.3. Bit complexity analysis

Theorem 12. *For f as above, there exist ε, δ, R, δ_c of bit sizes upper bounded by $O(\tau \cdot (4d+2)^{4n+3})$ such that* `intsos`$(f, \varepsilon, \delta, R, \delta_c)$ *runs in boolean time $O(\tau^2 \cdot (4d+2)^{15n+6})$.*

Proof. We consider ε, δ, R and δ_c as in the proof of Proposition 11, 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 $O(\tau \cdot (4d+2)^{4n+3})$.

Computing $C(f)$ with the quickhull algorithm runs in boolean time $O(V^2)$ 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 $O(d^{n+1})$. Next, we investigate the computational cost of the call to `sdp` at line 7. Let us note $n_{\text{sdp}} = r$ (resp. m_{sdp}) the size (resp. number of entries) of $\tilde{G}$. This step consists of solving SDP (1), which is performed in $O(n_{\text{sdp}}^4 \log_2(2^\tau n_{\text{sdp}} R\, 2^\delta))$ iterations of the ellipsoid method, where each iteration requires $O(n_{\text{sdp}}^2(m_{\text{sdp}} + n_{\text{sdp}}))$ arithmetic operations over $\log_2(2^\tau n_{\text{sdp}} R\, 2^\delta)$-bit numbers (see e.g. Grötschel et al. (1993)). Since $m_{\text{sdp}}, n_{\text{sdp}}$ are both bounded above by $\binom{n+d}{n} \leq 2d^n$, one has

$$\begin{aligned}
\log_2(2^\tau n_{\text{sdp}} R\, 2^\delta) &\leq O(\tau \cdot (4d+2)^{4n+3}), \\
n_{\text{sdp}}^2(m_{\text{sdp}} + n_{\text{sdp}}) &\leq O(d^{3n}), \\
n_{\text{sdp}}^4 \log_2(2^\tau n_{\text{sdp}} R\, 2^\delta) &\leq O(\tau \cdot (4d+2)^{8n+3}).
\end{aligned}$$

Overall, the ellipsoid algorithm runs in boolean time $O(\tau^2 \cdot (4d+2)^{15n+6})$ 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 $O(n_{\text{sdp}}^3)$ arithmetic operations over δ_c-bit numbers. Since $\delta_c \leq O(\tau \cdot (4d+2)^{4n+3})$, the function runs in boolean time $O(\tau \cdot (4d+2)^{7n+3})$. The other elementary arithmetic operations performed while running Algorithm `intsos` have a negligible cost w.r.t. to the `sdp` procedure. □

3.4. Comparison with the rounding-projection algorithm of Peyrl and Parrilo

We recall the algorithm designed in Peyrl and Parrilo (2008). We denote this rounding-projection algorithm by `RoundProject`.

The first main step in Line 5 consists of rounding the approximation $\tilde{G}$ of a Gram matrix associated to f up to precision δ_i. The second main step in Line 8 consists of computing the

Algorithm 3 `RoundProject`

Input: $f \in \mathbb{Z}[X]$, rounding precision $\delta_i \in \mathbb{N}$, precision parameters $\delta, R \in \mathbb{N}$ for the SDP solver
Output: list `c_list` of numbers in $\mathbb{Q}$ and list `s_list` of polynomials in $\mathbb{Q}[X]$

1: $P := C(f)$, $Q := P/2 \cap \mathbb{N}^n$
2: ok := false
3: **while** not ok **do**
4: $\quad(\tilde{G}, \tilde{\lambda}) :=$ `sdp`(f, δ, R)
5: $\quad G' :=$ `round`$(\tilde{G}, \delta_i)$
6: $\quad$**for** $\alpha, \beta \in Q$ **do**
7: $\quad\quad \eta(\alpha + \beta) := \#\{(\alpha', \beta') \in Q^2 \mid \alpha' + \beta' = \alpha + \beta\}$
8: $\quad\quad G(\alpha, \beta) := G'(\alpha, \beta) - \frac{1}{\eta(\alpha+\beta)}\Big(\sum_{\alpha'+\beta'=\alpha+\beta} G'(\alpha', \beta') - f_{\alpha+\beta}\Big)$
9: $\quad$**done**
10: $\quad(c_1, \ldots, c_r, s_1, \ldots, s_r) :=$ `ldl`(G) $\qquad \triangleright\ f = \sum_{i=1}^r c_i s_i^2$
11: $\quad$**if** $c_1, \ldots, c_r \in \mathbb{Q}^{>0}, s_1, \ldots, s_r \in \mathbb{Q}[X]$ **then** ok := true
12: $\quad$**else** $\delta := 2\delta$, $R := 2R$, $\delta_c := 2\delta_c$
13: $\quad$**end**
14: **done**
15: `c_list` $:= [c_1, \ldots, c_r]$, `s_list` $:= [s_1, \ldots, s_r]$
16: **return** `c_list`, `s_list`

orthogonal projection G of G' on an adequate affine subspace in such a way that $\sum_{\alpha+\beta=\gamma} G_{\alpha,\beta} = f_\gamma$, for all $\gamma \in \operatorname{supp}(f)$. For more details on this orthogonal projection, we refer to (Peyrl and Parrilo, 2008, Proposition 7). The algorithm then performs in (10) an exact diagonalization of the matrix G via the LDL^T decomposition (see e.g. (Golub and Loan, 1996, § 4.1)). It is proved in (Peyrl and Parrilo, 2008, Proposition 8) that for $f \in \mathring{\Sigma}[X]$, Algorithm `RoundProject` returns a weighted SOS decomposition of f with rational coefficients when the precision of the rounding and SDP solving steps are large enough.

The main differences w.r.t. Algorithm `intsos` are that `RoundProject` does not perform a perturbation of the input polynomial f and computes an exact LDL^T decomposition of a Gram matrix G. In our case, we compute an approximate Cholesky's decomposition of $\tilde{G}$ instead of a projection, then perform an exact compensation of the error terms, thanks to the initial perturbation.

The next result gives upper bounds on the bit size of the coefficients involved in the SOS decomposition returned by `RoundProject` as well as upper bounds on the boolean running time. Even though `intsos` and `RoundProject` have the same exponential bit complexity, the upper bound estimates are larger in the case of `RoundProject`. It would be worth investigating whether these bounds are tight in general.

Theorem 13. *For f as above, there exist δ_i, δ, R of bit sizes $\leq O(\tau \cdot (4d+2)^{4n+3})$ such that `RoundProject`(f, δ_i, δ, R) outputs a rational SOS decomposition of f with rational coefficients. The maximum bit size of the coefficients involved in this SOS decomposition is upper bounded by $O(\tau \cdot (4d+2)^{6n+3})$ and the boolean running time is $O(\tau^2 \cdot (4d+2)^{15n+6})$.*

Proof. Let us assume that Algorithm `RoundProject` returns a matrix $G \succ 0$ associated to f with smallest eigenvalue λ and let $N \in \mathbb{N}$ be the smallest integer such that $2^{-N} \leq \lambda$. As in Proposition 10, one proves that the bit size of N is upper bounded by $O(\tau \cdot (4d+2)^{3n+3})$. By (Peyrl and

Parrilo, 2008, Proposition 8), Algorithm `RoundProject` terminates and outputs such a matrix G together with a weighted rational SOS decomposition of f if $2^{-\delta_i} + 2^{-\delta'} \leq 2^{-N}$, where δ' stands for the euclidean distance between G' and G, yielding

$$\sqrt{\sum_{\alpha,\beta \in Q} (G_{\alpha,\beta} - G'_{\alpha,\beta})^2} = 2^{-\delta'} .$$

For all $\alpha, \beta \in Q$, one has $|G'_{\alpha,\beta} - \tilde{G}_{\alpha,\beta}| \leq 2^{-\delta_i}$. As in the proof of Proposition 11, at SDP precision δ, one has for all $\gamma \in \mathrm{supp}(f)$, $\tilde{G} \succeq 2^{-\delta} I$ and

$$|f_\gamma - \sum_{\alpha+\beta=\gamma} \tilde{G}_{\alpha,\beta}| \leq 2^{-\delta} .$$

For all $\alpha, \beta \in Q$, let us define $e_{\alpha,\beta} := \sum_{\alpha'+\beta'=\alpha+\beta} G'(\alpha',\beta') - f_{\alpha+\beta}$ and note that

$$|e_{\alpha,\beta}| \leq \sum_{\alpha'+\beta'=\alpha+\beta} \left| G'(\alpha',\beta') - \tilde{G}(\alpha',\beta') \right| + \left| \sum_{\alpha'+\beta'=\alpha+\beta} \tilde{G}(\alpha',\beta') - f_{\alpha+\beta} \right| \leq \eta(\alpha+\beta) 2^{-\delta_i} + 2^{-\delta} .$$

For all $\alpha, \beta \in Q$, we use the fact that $\eta(\alpha+\beta) \geq 1$ and that the cardinal of Q is less than the size r of G, with $r \leq d^n$, to obtain

$$2^{-\delta'} = \sum_{\alpha,\beta \in Q} \frac{e_{\alpha,\beta}}{\eta(\alpha+\beta)} \leq d^{2n} (2^{-\delta_i} + 2^{-\delta}).$$

To ensure that $2^{-\delta_i} + 2^{-\delta'} \leq 2^{-N}$, it is sufficient to have $(d^{2n} + 1) 2^{-\delta_i} + d^{2n} 2^{-\delta} \leq 2^{-N}$, which is obtained with δ_i and δ with bit size upper bounded by $O(\tau \cdot (4d+2)^{3n+3})$. The bit size of the coefficients involved in the weighted SOS decomposition is upper bounded by the output bit size of the LDL^T decomposition of the matrix G, that is $O(\delta_i r^3) = O(\tau \cdot (4d+2)^{6n+3})$.
The bound on the running time is obtained exactly as in Theorem 11. □

4. Exact Reznick and Hilbert-Artin's representations

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

4.1. Exact Reznick's representations

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 (1995) proved that for large enough $D \in \mathbb{N}$, $f G_n^D \in \Sigma[X]$. Such SOS decompositions are called *Reznick's representations* and D is called the *Reznick's degree*. The next result states that for large enough $D \in \mathbb{N}$, $f G_n^D \in \mathring{\Sigma}[X]$, as a direct consequence of Reznick (1995).

Lemma 14. *Let f be a positive definite form of degree $d = 2k$ in $\mathbb{Z}[X]$ and $D \geq \frac{nd(d-1)}{4\log 2\,\varepsilon(f)} - \frac{n+d}{2} + 1$. Then $f\,G_n^D \in \mathring{\Sigma}[X]$.*

Proof. First, for any positive $e < \min_{\boldsymbol{x}\in\mathbb{S}^{n-1}} f(\boldsymbol{x})$, the form $(f - eG_n^k)$ is positive on $\mathbb{S}^{n-1}$. Then, for any nonzero $\boldsymbol{x} \in \mathbb{R}^n$, one has

$$f(\boldsymbol{x}) - eG_n(\boldsymbol{x})^k = G_n(\boldsymbol{x})^k \Big(f\Big(\frac{\boldsymbol{x}}{G_n(\boldsymbol{x})}\Big) - e \Big) > 0\,,$$

implying that $(f - eG_n^k)$ is positive definite. Next, (Reznick, 1995, Theorem 3.12) implies that for any positive integer D_e such that

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

one has $(f - eG_n^k)\,G_n^{D_e} \in \Sigma[X]$. One has $G_n^{k+D_e} = \sum_{|\alpha|=k+D_e} \binom{k+D_e}{\alpha_1!\ldots\alpha_n!} X^{2\alpha}$. Let $v_{k+D_e}(X)$ be the vector of monomials with exponents in $\mathbb{N}^n_{k+D_e}$. Then, one can write $G_n^{k+D_e} = v_{k+D_e}^T A v_{k+D_e}$ with A being a diagonal matrix with positive entries $\binom{k+D_e}{\alpha_1!\ldots\alpha_n!}$, thus $A \succ 0$. Next we select e small enough so that there is no term cancellation in $(f - eG_n^k)$, ensuring that the Newton polytope of $(f - eG_n^k)$ is equal to $\mathbb{N}^n_{2k}$. This in turn implies that the Newton polytope of $(f - eG_n^k)\,G_n^{D_e}$ is equal to $\mathbb{N}^n_{2(k+D_e)}$. Since $(f - eG_n^k)\,G_n^{D_e} \in \Sigma[X]$, there exists $A' \succeq 0$ indexed by $\mathbb{N}^n_{k+D_e}$ such that $fG_n^{D_e} - eG_n^{k+D_e} = v_{k+D_e}^T A' v_{k+D_e}$. This yields $fG_n^{D_e} = v_{k+D_e}^T (eA + A') v_{k+D_e}$. Since $eA + A' \succ 0$, Theorem 3 implies that $f\,G_n^{D_e} \in \mathring{\Sigma}[X]$.

Next, with $\underline{D} := \frac{nd(d-1)}{4\log 2\,\varepsilon(f)} - \frac{n+d}{2}$, we prove that there exists a large enough $N \in \mathbb{N}$ such that for $e = \frac{\min_{\boldsymbol{x}\in\mathbb{S}^{n-1}} f(\boldsymbol{x})}{N}$, $\underline{D_e} \leq \underline{D} + 1$. 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 $G_n(\boldsymbol{x})^k = 1$, thus

$$\min_{\boldsymbol{x}\in\mathbb{S}^{n-1}} (f(\boldsymbol{x}) - eG_n(\boldsymbol{x})^k) = \min_{\boldsymbol{x}\in\mathbb{S}^{n-1}} f(\boldsymbol{x}) - e\,, \quad \max_{\boldsymbol{x}\in\mathbb{S}^{n-1}} (f(\boldsymbol{x}) - eG_n(\boldsymbol{x})^k) = \max_{\boldsymbol{x}\in\mathbb{S}^{n-1}} f(\boldsymbol{x}) - e\,.$$

Hence,

$$\varepsilon(f - eG_n^k) = \frac{\min_{\boldsymbol{x}\in\mathbb{S}^{n-1}} f(\boldsymbol{x})[1 - 1/N]}{\min_{\boldsymbol{x}\in\mathbb{S}^{n-1}} f(\boldsymbol{x})[1/\varepsilon(f) - 1/N]} = \frac{\varepsilon(f)(N-1)}{N - \varepsilon(f)}\,.$$

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

Algorithm `Reznicksos` 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 which decides if some given polynomial is a positive definite form. Further, we denote by `interiorSOScone` a routine which takes as input f, G_n and D and returns true if and only if $f\,G_n^D \in \mathring{\Sigma}[X]$, else it returns false. Then, `intsos` is applied on $f\,G_n^D$.

Example 15. *Let us apply `Reznicksos` on the perturbed Motzkin polynomial*

$$f = (1 + 2^{-20})(X_3^6 + X_1^4 X_2^2 + X_1^2 X_2^4) - 3X_1^2 X_2^2 X_3^2.$$

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 $\varepsilon = 2^{-20}$, $\delta = R = 60$, $\delta_c = 10$.

Algorithm 4 `Reznicksos`

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** `interiorSOScone`$(f\,G_n, D) =$ false **do** $D := D + 1$
3: **done**
4: **return** `intsos`$(f\,G_n^D, \varepsilon, \delta, R, \delta_c)$

Theorem 16. *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* `Reznicksos` *terminates and outputs a weighted SOS decomposition for f. The maximum bit size of the coefficients involved in the decomposition and the boolean running time of the procedure are both upper bounded by $2^{O(\tau\cdot(4d+2)^{4n+3})}$.*

Proof. By Lemma 14, the while loop from line 2 to 3 is ensured to terminate for a positive integer $D \geq \frac{nd(d-1)}{4\log 2\,\varepsilon(f)} - \frac{n+d}{2}$. By Proposition 11, when applying `intsos` to $f\,G_n^D$, the procedure always terminates. The outputs are a list of non-negative rational numbers $[c_1, \dots, c_r]$ and a list of rational polynomials $[s_1, \dots, s_r]$ 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 11, the maximum bit size of the coefficients involved in the weighted SOS decomposition of $f\,G_n^D$ is upper bounded by $O((\tau + D\log n)(4d + 8D + 2)^{3n+3})$. Now, we derive an upper bound on D. One has $\min_{\boldsymbol{x}\in\mathbb{S}^{n-1}} f(\boldsymbol{x}) := \min\{e \in \mathbb{R}^{>0} : f(\boldsymbol{x}) - e = 0\,, \boldsymbol{x} \in \mathbb{S}^{n-1}\}$.
Again, we rely on Proposition 6 to show that $\min_{\boldsymbol{x}\in\mathbb{S}^{n-1}} f(\boldsymbol{x}) \geq 2^{-O(\tau\cdot(4d+2)^{3n+3})}$. Similarly, we obtain $\max_{\boldsymbol{x}\in\mathbb{S}^{n-1}} f(\boldsymbol{x}) \leq 2^{O(\tau\cdot(4d+2)^{3n+3})}$ and thus $\frac{1}{\varepsilon(f)} \leq 2^{O(\tau\cdot(4d+2)^{3n+3})}$. Overall, we obtain

$$\frac{nd(d-1)}{4\log 2\,\varepsilon(f)} - \frac{n+d}{2} + 1 \leq D \leq 2^{O(\tau\cdot(4d+2)^{3n+3})}.$$

This implies that

$$O((\tau + D\log n)(4d + 8D + 2)^{3n+3}) \leq 2^{(3n+3)\,O(\tau\cdot(4d+2)^{3n+3})} \leq 2^{O(\tau\cdot(4d+2)^{4n+3})}.$$

From Theorem 12, the running time is upper bounded by $O((\tau + D\log n)^2(4d + 8D + 2)^{15n+6})$, which ends the proof. ☐

The bit complexity of `Reznicksos` is polynomial in the Reznick's degree D of the representation. In all the examples we tackled, this degree was rather small as shown in Section 6.

4.2. Exact Hilbert-Artin's representations

Here, we focus on the subclass of non-negative polynomials in $\mathbb{Z}[X]$ which admit an Hilbert-Artin's representation of the form $f = \frac{\mathring{\sigma}}{h^2}$, with h being a nonzero polynomial in $\mathbb{R}[X]$ and $\mathring{\sigma} \in \mathring{\Sigma}[X]$.
We start to recall the famous result by Artin, providing a general solution to Hilbert's 17th problem:

Theorem 17. *(Artin, 1927, Theorem 4) Let $f \in \mathbb{R}[X]$ be a polynomial non-negative over the reals. Then, f can be decomposed as a sum of squares of rational functions with rational coefficients and there exist a nonzero $h \in \mathbb{Q}[X]$ and $\sigma \in \Sigma[X]$ such that $f = \frac{\sigma}{h^2}$.*

Given $f \in \mathbb{R}[X]$ non-negative over the reals, let us note $\deg f = d = 2k$, and $\tau = \tau(f)$. Given $D \in \mathbb{N}$, we denote by S_D the convex hull of the set

$$\operatorname{supp}(f) + \mathbb{N}^n_{2D} = \{\alpha + \beta \mid \alpha \in \operatorname{supp}(f), \beta \in \mathbb{N}^n_{2D}\} \subseteq \mathbb{N}^n_{d+2D}.$$

Finally, we set $Q_D := S_D/2 \cap \mathbb{N}^n_{k+D}$.
To compute Hilbert-Artin's representation, one can solve the following SDP program:

$$\begin{aligned} \sup_{G,H \succeq 0} \quad & \operatorname{Tr} G \\ \text{s.t.} \quad & \operatorname{Tr}(H F_\gamma) = \operatorname{Tr}(G B_\gamma), \quad \forall \gamma \in Q_D, \\ & \operatorname{Tr}(H) = 1. \end{aligned} \tag{4}$$

where B_γ is as for SDP (1), with rows (resp. columns) indexed by Q_D, and F_γ has rows (resp. columns) indexed by $\mathbb{N}^n_D$ with (α,β) entry equal to $\sum_{\alpha+\beta+\delta=\gamma} f_\delta$. Let us now provide the rationale behind SDP (4). The first set of trace equality constraints allows one to find a Gram matrix H associated to h^2, with rows (resp. columns) indexed by $\mathbb{N}^n_D$, as well as a Gram matrix G associated to σ, with rows (resp. columns) indexed by Q_D. The last trace equality constraint allows one to ensure that H is not the zero matrix. Note that we are only interested in finding a stricly feasible solution for SDP (4), thus we can choose any objective function. Here, we maximize the trace, as we would like to obtain a full rank matrix for G.

Proposition 18. *Let $f \in \mathbb{Z}[X]$ be a polynomial non-negative over the reals, with* $\deg f = d = 2k$. *Let us assume that f admits the Hilbert-Artin's representation $f = \frac{\sigma}{h^2}$, with $\sigma \in \mathring{\Sigma}[X]$, $h \in \mathbb{Q}[X]$,* $\deg h = D \in \mathbb{N}$ *and* $\deg \sigma = 2(D+k)$. *Let Q_D be defined as above. Then, there exist $\mathring{\sigma}_D, \mathring{\sigma} \in \mathring{\Sigma}[X]$ such that*

$$\mathring{\sigma}_D f = \mathring{\sigma},$$

ensuring the existence of a strictly feasible solution $G, H \succ 0$ for SDP (4).

Proof. By applying Proposition 5 to $h^2 f$, there exists $\varepsilon > 0$ such that $\tilde{\sigma} := h^2 f - \varepsilon \sum_{\alpha \in Q_D} X^{2\alpha} \in \mathring{\Sigma}[X]$. In addition, for all $\lambda > 0$, one has

$$h^2 f = h^2 f + \lambda f \sum_{\alpha \in \mathbb{N}^n_D} X^{2\alpha} - \lambda f \sum_{\alpha \in \mathbb{N}^n_D} X^{2\alpha} = \Big(h^2 + \lambda \sum_{\alpha \in \mathbb{N}^n_D} X^{2\alpha}\Big) f - \lambda f \sum_{\alpha \in \mathbb{N}^n_D} X^{2\alpha} = \tilde{\sigma} + \varepsilon \sum_{\alpha \in Q_D} X^{2\alpha}.$$

Let us define $u_\lambda := \lambda f \sum_{\alpha \in \mathbb{N}^n_D} X^{2\alpha}$. As in the proof of Proposition 11, we show that for small enough λ, the polynomial $\varepsilon \sum_{\alpha \in Q_D} X^{2\alpha} + u_\lambda$ belongs to $\Sigma[X]$. Fix such a λ, and define $\mathring{\sigma} := \tilde{\sigma} + \varepsilon \sum_{\alpha \in Q_D} X^{2\alpha} + u_\lambda$ and $\mathring{\sigma}_D := h^2 + \lambda \sum_{\alpha \in \mathbb{N}^n_D} X^{2\alpha}$. Since $\tilde{\sigma} \in \mathring{\Sigma}[X]$, there exists a positive definite Gram matrix G associated to $\mathring{\sigma}$. Similarly, there exists a positive definite Gram matrix H associated to $\mathring{\sigma}_D$. By Theorem 3, this implies that $\mathring{\sigma}, \mathring{\sigma}_D \in \mathring{\Sigma}[X]$, showing the claim. □

To find such representations in practice, we consider a perturbation of the trace equality constraints of SDP (4) where we replace the matrix G by the matrix $G - \epsilon \mathbf{I}$:

$$\begin{aligned} \mathbf{P}^\varepsilon \quad : \quad \sup_{G,H \succeq 0} \quad & \operatorname{Tr} G \\ \text{s.t.} \quad & \operatorname{Tr}(H F_\gamma) = \operatorname{Tr}(G B_\gamma) - \varepsilon \operatorname{Tr}(B_\gamma), \quad \forall \gamma \in Q_D, \\ & \operatorname{Tr}(H) = 1. \end{aligned}$$

For $D \in \mathbb{N}$, let us note $\mathring{\Sigma}_D(X) := \{\frac{\sigma}{\sigma_D} : \sigma \in \mathring{\Sigma}[X], \sigma_D \in \Sigma[X] \text{ with } \deg \sigma_D \le 2D\}$.
Algorithm `Hilbertsos` takes as input $f \in \mathbb{Z}[X]$, finds $\sigma_D \in \Sigma[X]$ of smallest degree $2D$ such that $f\sigma_D \in \mathring{\Sigma}[X]$, thanks to an oracle as in `intsos` (i.e., the smallest D for which $f \in \mathring{\Sigma}_D(X)$). Then, the algorithm finds the largest rational $\varepsilon > 0$ such that Problem $\mathbf{P}^\varepsilon$ has a strictly feasible solution. Problem $\mathbf{P}^\varepsilon$ is solved by calling the `sdp` function, relying on an SDP solver. Eventually, the algorithm calls the procedure `absorb`, as in `intsos`, to recover an exact rational SOS decomposition.

Algorithm 5 `Hilbertsos`

Input: $f \in \mathbb{Z}[X]$ of degree $d = 2k$, 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 lists $\texttt{c\_list}_1, \texttt{c\_list}_2$ of numbers in $\mathbb{Q}$ and lists $\texttt{s\_list}_1, \texttt{s\_list}_2$ of polynomials in $\mathbb{Q}[X]$

$D := 1$
while $f \notin \mathring{\Sigma}[X]/\Sigma_D[X]$ **do** $D := D + 1$
done
Compute the convex hull S_D of $\mathrm{supp}(f) + \mathbb{N}^n_{d+2D}$
$Q_D := S_D/2 \cap \mathbb{N}^n_{k+D}$
$t := \sum_{\alpha \in Q_D} X^{2\alpha}$
while Problem $\mathbf{P}^\varepsilon$ has no strictly feasible solution **do** $\varepsilon := \frac{\varepsilon}{2}$
done
ok := false
while not ok **do**
  $(\tilde{G}, \tilde{H}, \tilde{\lambda}_1, \tilde{\lambda}_2) := \texttt{sdp}(f, \varepsilon, \delta, R)$
  $(s_{11}, \dots, s_{1r_1}) := \texttt{cholesky}(\tilde{G}, \tilde{\lambda}_1, \delta_c)$
  $(s_{21}, \dots, s_{2r_2}) := \texttt{cholesky}(\tilde{H}, \tilde{\lambda}_2, \delta_c)$
  $\tilde{\sigma} := \sum_{i=1}^{r_1} s_{1i}^2$, $\tilde{\sigma}_D := \sum_{i=1}^{r_2} s_{2i}^2$
  $u := \tilde{\sigma}_D f - \tilde{\sigma} - \varepsilon t$
  $\texttt{c\_list}_1 := [1, \dots, 1]$, $\texttt{s\_list}_1 := [s_{11}, \dots, s_{r_1 1}]$
  $\texttt{c\_list}_2 := [1, \dots, 1]$, $\texttt{s\_list}_2 := [s_{12}, \dots, s_{r_2 2}]$
  for $\alpha \in Q_D$ **do** $\varepsilon_\alpha := \varepsilon$
  done
  $\texttt{c\_list}_1, \texttt{s\_list}_1, (\varepsilon_\alpha) := \texttt{absorb}(u, Q_D, (\varepsilon_\alpha), \texttt{c\_list}_1, \texttt{s\_list}_1)$
  if $\min_{\alpha \in Q_D}\{\varepsilon_\alpha\} \ge 0$ **then** ok := true
  else $\delta := 2\delta$, $R := 2R$, $\delta_c := 2\delta_c$
  end
done
for $\alpha \in Q_D$ **do** $\texttt{c\_list}_1 := \texttt{c\_list}_1 \cup \{\varepsilon_\alpha\}$, $\texttt{s\_list}_1 := \texttt{s\_list}_1 \cup \{X^\alpha\}$
done
return $\texttt{c\_list}_1, \texttt{c\_list}_2, \texttt{s\_list}_1, \texttt{s\_list}_2$

Theorem 19. *Let $f \in \mathbb{Z}[X] \cap \mathring{\Sigma}_D(X)$ and assume that the SOS polynomials involved in the denominator of f have coefficients of bit size at most $\tau_D \ge \tau$. On input f, Algorithm `Hilbertsos` terminates and outputs a weighted SOS decomposition for f. There exist $\varepsilon, \delta, R, \delta_c$ of bit sizes upper bounded by $O(\tau_D \cdot (4d + 4D + 2)^{3n+3})$ such that `Hilbertsos`$(f, \varepsilon, \delta, R, \delta_c)$ runs in boolean running time $O(\tau_D^2 \cdot (4d + 4D + 2)^{15n+6})$.*

Proof. Since $f \in \mathring{\Sigma}_D(X)$, the first loop of Algorithm `Hilbertsos` terminates and there exists a strictly feasible solution for SDP (4), by Proposition (18). Thus, there exists a small enough $\varepsilon > 0$ such that Problem $\mathbf{P}^{\varepsilon}$ has also a strictly feasible solution. This ensures that the second loop of Algorithm `Hilbertsos` terminates. Then, one shows as for Algorithm `intsos` that the absorption procedure succeeds, yielding termination of the third loop. Let us note

$$\Delta_D(u) := \{(\alpha,\beta) : \alpha+\beta \in \operatorname{supp}(u)\,, \alpha,\beta \in Q_D\,, \alpha \neq \beta\}\,.$$

The first output `c_list`$_1$ of Algorithm `Hilbertsos` is a list of non-negative rational numbers containing the list $[1,\ldots,1]$ of length r_1, the list $\{\frac{|u_{\alpha+\beta}|}{2} : (\alpha,\beta) \in \Delta_D(u)\}$ and the list $\{\varepsilon_\alpha : \alpha \in Q_D\}$. The second output `s_list`$_1$ of Algorithm `intsos` is a list of polynomials containing the list $[s_{11},\ldots,s_{r_1 1}]$, the list $\{X^\alpha + \operatorname{sgn}(u_{\alpha+\beta})X^\beta : (\alpha,\beta) \in \Delta_D(u)\}$ and the list $\{X^\alpha : \alpha \in Q_D\}$. From these two outputs, one reconstructs the weighted SOS decomposition of the numerator σ of f. The third output `c_list`$_2$ is a list of non-negative rational numbers containing the list $[1,\ldots,1]$ of length r_2 and the fourth output is a list of polynomials $[s_{12},\ldots,s_{r_2 2}]$. From these two outputs, one reconstructs the weighted SOS decomposition of the denominator σ_D of f. At the end, we obtain the weighted SOS decomposition $f = \frac{\sigma}{\sigma_D}$ with

$$\sigma_D := \sum_{i=1}^{r_2} s_{2i}^2\,, \quad \sigma := \sum_{i=1}^{r_1} s_{1i}^2 + \sum_{(\alpha,\beta)\in\Delta_D(u)} \frac{|u_{\alpha+\beta}|}{2}(X^\alpha + \operatorname{sgn}(u_{\alpha+\beta})X^\beta)^2 + \sum_{\alpha\in Q_D} \varepsilon_\alpha X^{2\alpha}\,.$$

Writing $f = \frac{\sigma}{\sigma_D}$, one shows as in Proposition 5 that the largest rational number belonging to the set $\{\varepsilon \in \mathbb{R}^{>0} : \forall \boldsymbol{x} \in \mathbb{R}^n, \sigma_D(\boldsymbol{x}) f(\boldsymbol{x}) - \varepsilon \sum_{\alpha\in Q_D} \boldsymbol{x}^{2\alpha} \geq 0\}$ has bit size upper bounded by $O(\tau_D \cdot (4d+4D+2)^{3n+3})$. We conclude our bit complexity analysis as in Proposition 11 and Theorem 12. □

Remark 20. *Note that even if the bit complexity of* `Hilbertsos` *is polynomial in the degree D of the denominator, this degree can be rather large. In Lombardi et al. (2018), the authors provide an upper bound expressed with a tower of five exponentials for the degrees of denominators involved in Hilbert-Artin's representations.*

5. Exact Putinar's representations

We let $f, g_1, \ldots, g_m$ in $\mathbb{Z}[X]$ of degrees less than $d \in \mathbb{N}$ and $\tau \in \mathbb{N}$ 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 $Q(S) := \{\sum_{j=0}^m \sigma_j g_j : \sigma_j \in \Sigma[\boldsymbol{x}]\}$ and, for $D \in \mathbb{N}$, the D-truncated quadratic module $Q_D(S) := \{\sum_{j=0}^m \sigma_j g_j : \sigma_j \in \Sigma[\boldsymbol{x}]\,, \deg(\sigma_j g_j) \leq D\}$ generated by $g_1, \ldots, g_m$. We say that $Q(S)$ is *archimedean* if $N - G_n \in Q(S)$ for some $N \in \mathbb{N}$ (recall that $G_n := \sum_{i=1}^n X_i^2$). We also assume in this section:

Assumption 21. *The set S is a basic compact semi-algebraic set with nonempty interior, included in $[-1,1]^n$ and $Q(S)$ is archimedean.*

Under Assumption 21, f is positive over S only if $f \in Q_D(S)$ for some $D \in 2\mathbb{N}$ (see Putinar (1993)). 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

$0 \le j \le m$. One can certify that $f \in Q_D(S)$ for $D = 2k$ by solving the next SDP with $k \ge \lceil d/2 \rceil$:

$$\begin{aligned} \inf_{G_0, G_1, \dots, G_m \succeq 0} \quad & \mathrm{Tr}\,(G_0\, B_0) + \sum_{i=1}^{m} g_j(0)\, \mathrm{Tr}\,(G_j\, C_{j0}) \\ \text{s.t.} \quad & \mathrm{Tr}\,(G_0\, B_\gamma) + \sum_{j=1}^{m} \mathrm{Tr}\,(G_j\, C_{j\gamma}) = f_\gamma\,, \quad \forall \gamma \in \mathbb{N}_D^n - \{0\}\,, \end{aligned} \tag{5}$$

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 (5) is a reformulation of the problem

$$f_D^\star := \sup\{b : f - b \in Q_D(S)\}\,.$$

Thus $f_D^\star$ is also the optimal value of SDP (5). The next result follows from (Lasserre, 2001, Theorem 4.2):

Theorem 22. *We use the notation and assumptions introduced above. For $D \in 2\mathbb{N}$ large enough, one has*

$$0 < f_D^\star \le f^\star\,.$$

In addition, SDP (5) has an optimal solution $(G_0, G_1, \dots, G_m)$, 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 $j = 0, \dots, m$.

The complexity of Putinar's Positivstellensätz was analyzed by Nie and Schweighofer (2007):

Theorem 23. *With the notation and assumptions introduced above, there exists a real $\chi_S > 0$ depending on S such that*

(i) for all even $D \ge \chi_S \exp(d^2 n^d \frac{\|f\|}{f^\star})^{\chi_S}$, $f \in Q_D(S)$.

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

From a computational viewpoint, one can certify that f lies in $Q_D(S)$ for $D = 2k$ large enough, by solving SDP (5). Next, we show how to ensure the existence of a strictly feasible solution for SDP (5) after replacing the set defined by our initial constraints S by the following one

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

5.1. Preliminary results

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

Proposition 24. *With the above notation and assumptions, one has*

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

Proof. Let $Y = (Y_1, \dots, Y_n)$ and $\tilde{f} \in \mathbb{Z}[Y]$ be the polynomial obtained by replacing Y_i by $2nY_i - 1$ in f. Note that if $\boldsymbol{x} = (x_1, \dots, x_n) \in S \subseteq [-1,1]^n$, then $\boldsymbol{y} = \left(\left(\frac{x_i+1}{2n}\right)\right)_{1\le i\le n}$ 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 $(2n-1)^d \le (2n)^d$, the polynomial $\tilde{f}$ has coefficients of bit size at most $\tau + d + d\log_2 n$. Then, the inequality follows from (Jeronimo and Perrucci, 2010, Theorem 1), stating that

$$\min_{\boldsymbol{y}\in\Delta_n} \tilde{f}(\boldsymbol{y}) > 2^{-(\tau(\tilde{f})+1)d^{n+1}} d^{-(n+1)d^{n+1}} .$$

We obtain the second inequality after noticing that for all $d \ge 2$, one has $d\log_2 nd^{n+1} \le d^{2n+2}$, $nd^{n+1} \le d^{2n+1}$, $d \le 2^d$, and $2^{d^{2n+2}} d^{d^{2n+1}} \le 2^{2d^{2n+2}}$. □

Theorem 25. *We use the notation and assumptions introduced above. There exists $D \in 2\mathbb{N}$ such that:*
(i) $f \in Q_D(S)$ 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 $\deg(\sigma_j g_j) \le D$ for all $j = 0, \dots, m$.
(ii) $f \in Q_D(S')$ with the representation

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

for $\mathring{\sigma}_j \in \mathring{\Sigma}[X]$ with $\deg(\mathring{\sigma}_j g_j) \le D$, for all $j = 0, \dots, m$, and some sequence of positive numbers $(c_\alpha)_{|\alpha|\le k}$.
(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

$$f - \varepsilon \sum_{|\alpha|\le k} X^{2\alpha} \in Q_D(S') , \quad N \le 2^{C_S \tau d^{2n+2}} ,$$

where τ is the maximal bit size of the coefficients of $f, g_1, \dots, g_m$.

Proof. Let χ_S be as in Theorem 23 and $D = 2k$ be the smallest integer larger than

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

Theorem 23 implies that $f \in Q_D(S)$ and $f^\star - f_D^\star \le \frac{6d^3 n^{2d}\|f\|}{\chi_S\sqrt{\log \frac{D}{\chi_S}}} \le \frac{f^\star}{2}$.

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

(ii) For $1 \le j \le m$, let us define

$$t_j := \sum_{|\alpha|\le k-w_j} X^{2\alpha} , \quad t_0 := \sum_{|\alpha|\le k} X^{2\alpha} , \quad t := \sum_{j=0}^{m} t_j g_j .$$

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 11, let us note

$$\Delta(t) := \{(\alpha, \beta) : \alpha + \beta \in \operatorname{supp}(t) \,, \alpha, \beta \in \mathbb{N}^n_k \,, \alpha \neq \beta\} \,.$$

Hence, for all $\alpha \in \mathbb{N}^n_k$, there exists $d_\alpha \geq 0$ such that

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

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 $j = 1, \dots, m$ and

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

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 $0 < \nu \leq \frac{f_D^\star}{2a}$. This implies the existence of a positive definite Gram matrix for $\mathring{\sigma}_0$, thus by Theorem 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 $\frac{f_D^\star}{a}$. 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 $\frac{f^\star}{2} \leq f_D^\star$, implying

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

where the last inequality follows from Theorem 24. 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

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

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

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

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

5.2. Algorithm `Putinarsos`

We can now present Algorithm `Putinarsos`.

For $f \in \mathbb{Z}[X]$ positive over a basic compact semi-algebraic set S satisfying Assumption 21, the first loop outputs the smallest positive integer $D = 2k$ such that $f \in Q_D(S)$.
Then the procedure is similar to `intsos`. As for the first loop of `intsos`, the loop from line 6 to line 7 allows us to obtain a perturbed polynomial $f_\varepsilon \in Q_D(S')$, with $S' := \{\boldsymbol{x} \in S : 1 - \boldsymbol{x}^{2\alpha} \geq 0, \forall \alpha \in \mathbb{N}_k^n\}$.
Then one solves SDP (5) 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 $Q_D(S')$ with rational coefficients.

Example 26. *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 9. The first and second loop yield* $D = 2$ *and* $\varepsilon = 1$. *After running* `absorb`, *we obtain the exact Putinar's representation*

$$f = \frac{23853407}{292204836} + \frac{23}{49}X_1^2 + \frac{130657269}{291009481}X_2^2 + \frac{1}{2442^2} + (X_1 - X_2)^2 + (\frac{X_2}{2437})^2 + (\frac{11}{7})^2(1 - X_1^2) + (\frac{13}{7})^2(1 - X_2^2).$$

Algorithm 6 `Putinarsos`.

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

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 $\mathbb{Q}[X]$

$k := \lceil d/2 \rceil$, $D := 2k$, $g_0 := 1$
while $f \notin Q_D(S)$ **do** $k := k + 1$, $D := D + 2$
done
$P := \mathbb{N}^n_D$, $Q := \mathbb{N}^n_k$, $S' := \{\boldsymbol{x} \in S : 1 - \boldsymbol{x}^{2\alpha} \geq 0\,, \forall \alpha \in \mathbb{N}^n_k\}$
$t := \sum_{\alpha \in Q} X^{2\alpha}$, $f_\varepsilon := f - \varepsilon t$
while $f_\varepsilon \notin Q_D(S')$ **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 $\alpha \in Q$ **do** $\varepsilon_\alpha := \varepsilon$
  done
  $\texttt{c\_list}, \texttt{s\_list}, (\varepsilon_\alpha) := \texttt{absorb}(u, Q, (\varepsilon_\alpha), \texttt{c\_list}, \texttt{s\_list})$
  if $\min_{\alpha \in Q}\{\varepsilon_\alpha\} \geq 0$ **then** ok := true
  else $\delta := 2\delta$, $R := 2R$, $\delta_c := 2\delta_c$
  end
done
for $\alpha \in Q$ **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 $\texttt{c\_list}_0, \ldots, \texttt{c\_list}_m, \texttt{c\_alpha}, \texttt{s\_list}_0, \ldots, \texttt{s\_list}_m$

5.3. Bit complexity analysis

Theorem 27. *We use the notation and assumptions introduced above. For some constants $C_S > 0$ and K_S depending on S, there exist ε, δ, R, δ_c and $D = 2k$ of bit sizes less than $O(2^{C_S \tau d^{2n+2}})$ for which $\texttt{Putinarsos}(f, S, \varepsilon, \delta, R, \delta_c)$ terminates and outputs an exact Putinar's representation with rational coefficients of $f \in Q(S')$, with $S' := \{\boldsymbol{x} \in S : 1 - \boldsymbol{x}^{2\alpha} \geq 0, \forall \alpha \in \mathbb{N}_k^n\}$. The maximum bit size of these coefficients is bounded by $O(2^{C_S \tau d^{2n+2}})$ and the procedure runs in boolean time $O(2^{2^{K_S \tau d^{2n+2}}})$.*

Proof. The loops going from line 2 to line 3 and from line 6 to line 7 always terminate as respective consequences of Theorem 25 (i) and Theorem 25 (iii) with $\log_2 D \leq 2^{C_S \tau d^{2n+2}}$, $\varepsilon = \frac{1}{2^N}$, $N \leq 2^{C_S \tau d^{2n+2}}$, for some real $C_S > 0$ depending on S. What remains to prove is similar to Proposition 11 and Theorem 12.

Let $\nu, \mathring{\sigma}_0, \ldots, \mathring{\sigma}_m, (c_\alpha)_{|\alpha| \leq k}$ be as in the proof of Theorem 25. 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 $\alpha \in \mathbb{N}_k^n$. 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 $\tilde{G}_j \geq 2^\delta I$, $\sqrt{\operatorname{Tr}(\tilde{G}_j^2)} \leq R$, 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 $\alpha \in \mathbb{N}_k^n$. This happens when $2^{-\delta} \leq \varepsilon$ and $2^{-\delta} \leq \varepsilon - \nu$, thus for $\delta = O(2^{C_S \tau d^{2n+2}})$.

As in the proof of Proposition 10, we derive a similar upper bound of R by a symmetric argument while considering a Putinar representation of $\overline{f}_D - f \in Q_D(S')$, where

$$\overline{f}_D := \inf\{b : b - f \in Q_D(S)\}.$$

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 $Q = \mathbb{N}_k^n$. As in the proof of Proposition 11, 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 11. The difference now is that there are $m + \binom{n+k}{k} = m + r_0$ additional terms in each equality constraint of SDP (5), by comparison with SDP (1). Thus, we need to bound for all $j = 1 \ldots, m$, $\alpha \in \mathbb{N}_k^n$ 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 $2^{-\delta} \le \frac{\epsilon}{2r_0}$ and the smallest δ_c such that

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

Thus, one can choose δ and δ_c of bit size upper bounded by $O(2^{C_S \tau d^{2n+2}})$ in order to ensure that `Putinarsos` terminates. As in the proof of Proposition (11), one shows that the output is an exact Putinar's representation with rational coefficients of maximum bit size bounded by $O(2^{C_S \tau d^{2n+2}})$. As in the proof of Theorem 12, let n_{sdp} be the sum of the sizes of the matrices involved in SDP (5) and m_{sdp} be the number of entries. Note that

$$n_{\text{sdp}} \le (m+1)r_0 + r_0 \le (m+2)\binom{n+D}{n}, \quad m_{\text{sdp}} := \binom{n+D}{n}.$$

To bound the boolean run time, we consider the cost of solving SDP (5), which is performed in $O(n_{\text{sdp}}^4 \log_2(2^\tau n_{\text{sdp}} R\, 2^\delta))$ iterations of the ellipsoid method, where each iteration requires $O(n_{\text{sdp}}^2(m_{\text{sdp}} + n_{\text{sdp}}))$ arithmetic operations over $\log_2(2^\tau n_{\text{sdp}} R\, 2^\delta)$-bit numbers. Since m_{sdp} is bounded by $\binom{n+D}{n} \le 2D^n$ and $\log_2 D = O(2^{C_S \tau d^{2n+2}})$, one has

$$D^n = O(2^{n 2^{C_S \tau d^{2n+2}}}) \le O(2^{2^{(C_S+1)\tau d^{2n+2}}}) .$$

We obtain a similar bound for n_{sdp}, which ends the proof. □

As for `Reznicksos`, the complexity is polynomial in the degree D of the representation. On all the examples we tackled, it was close to the degrees of the involved polynomials, as shown in Section 6.

5.4. Comparison with the rounding-projection algorithm of Peyrl and Parrilo

We now state a constrained version of the rounding-projection algorithm from Peyrl and Parrilo (2008).

For $f \in \mathbb{Z}[X]$ positive over a basic compact semi-algebraic set S satisfying Assumption 21, Algorithm `RoundProjectPutinar` starts as in Algorithm `Putinarsos` (see Section 5.2): it outputs the smallest D such that $f \in Q_D(S)$, solves SDP (5) in Line 6, and performs Cholesky's factorization in Line 9 to obtain an approximate Putinar's representation of f. Note that the approximate Cholesky's factorization is performed to obtain weighted SOS decompositions associated to the constraints $g_1, \dots, g_m$ (i.e. $\tilde{\sigma}_1, \dots, \tilde{\sigma}_m$, respectively).
Next, the algorithm applies in Line 15 the same projection procedure of Algorithm `RoundProject` (see Section 3.4) on the polynomial $u := f - \sum_{j=1}^m \tilde{\sigma}_j g_j$. Note that when there are no constraints, one retrieves exactly the projection procedure from Algorithm `RoundProject`. Exact LDL^T is then performed on the matrix G corresponding to u.
If all input precision parameters are large enough, G is a Gram matrix associated to u and $\tilde{\sigma}_1, \dots, \tilde{\sigma}_m$ are weighted SOS polynomals, yielding the exact Putinar's representation $f = u + \sum_{j=1}^m \tilde{\sigma}_j g_j$. As for Theorem 13 and Theorem 27, Algorithm `RoundProjectPutinar` has a similar bit complexity than `Putinarsos`.

Algorithm 7 `RoundProjectPutinar`

Input: $f, g_1, \ldots, g_m \in \mathbb{Z}[X]$ of degrees less than $d \in \mathbb{N}$, $S := \{\boldsymbol{x} \in \mathbb{R}^n : g_1(\boldsymbol{x}) \geq 0, \ldots, g_m(\boldsymbol{x}) \geq 0\}$, rounding precision $\delta_i \in \mathbb{N}$, precision parameters $\delta, R \in \mathbb{N}$ for the SDP solver, precision $\delta_c \in \mathbb{N}$ for the Cholesky's decomposition

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

$k := \lceil d/2 \rceil$, $D := 2k$, $g_0 := 1$
while $f \notin Q_D(S)$ **do** $k := k + 1$, $D := D + 2$
done
ok := false
while not ok **do**
  $[\tilde{G}_0, \ldots, \tilde{G}_m, \tilde{\lambda}_0, \ldots, \tilde{\lambda}_m], := \texttt{sdp}(f, \delta, R, S)$
  $G' := \texttt{round}(\tilde{G}_0, \delta_i)$
  for $j \in \{1, \ldots, m\}$ **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
  $u := f - \sum_{j=1}^{m} \tilde{\sigma}_j$
  $Q := \mathbb{N}_k^n$
  for $\alpha, \beta \in Q$ **do** $\eta(\alpha + \beta) := \#\{(\alpha', \beta') \in Q^2 \mid \alpha' + \beta' = \alpha + \beta\}$
    $G(\alpha, \beta) := G'(\alpha, \beta) - \frac{1}{\eta(\alpha+\beta)} \Big(\sum_{\alpha'+\beta'=\alpha+\beta} G'(\alpha', \beta') - u_{\alpha+\beta} \Big)$
  done
  $(c_{10}, \ldots, c_{r_0 0}, s_{10}, \ldots, s_{r_0 0}) := \texttt{ldl}(G)$ ▷ $f = \sum_{i=1}^{r_0} c_{i0} s_{i0}^2 + \sum_{j=1}^{m} \tilde{\sigma}_j$
  if $c_{10}, \ldots, c_{mr_m} \in \mathbb{Q}^{\geq 0}$, $s_{01}, \ldots, s_{mr_m} \in \mathbb{Q}[X]$ **then** ok := true
  else $\delta_i := 2\delta_i$, $\delta := 2\delta$, $R := 2R$, $\delta_c := 2\delta_c$
  end
done
$\texttt{c\_list}_0 := [c_{10}, \ldots, c_{r_0 0}]$, $\texttt{s\_list}_0 := [s_{10}, \ldots, s_{r_0 0}]$
return $\texttt{c\_list}_0, \ldots, \texttt{c\_list}_m, \texttt{s\_list}_0, \ldots, \texttt{s\_list}_m$

Table 1: `multivsos` vs `univsos2` Magron et al. (2018) for benchmarks from Chevillard et al. (2011).

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

6. Practical experiments

We provide experimental results for Algorithms `intsos`, `Reznicksos` and `Putinarsos`. These are implemented in a procedure, called `multivsos`, and integrated in the `RealCertify` library by Magron and Safey El Din (2018b), written in Maple. More details about installation and benchmark execution are given on the dedicated webpage[1]. All results were obtained on an Intel Core i7-5600U CPU (2.60 GHz) with 16Gb of RAM. We use the Maple `Convex` package[2] to compute Newton polytopes. Our subroutine `sdp` relies on the arbitrary-precision solver SDPA-GMP by Nakata (2010) and the `cholesky` procedure is implemented with `LUDecomposition` available within Maple. Most of the time is spent in the `sdp` procedure for all benchmarks. To decide non-negativity of polynomials, we use either `RAGLib` or the `sdp` procedure as oracles. Recall that `RAGLib` relies on critical point methods whose runtime strongly depends on the number of (complex) solutions to polynomial systems encoding critical points. While these methods are more versatile, this number is generically exponential in n. Hence, we prefer to rely at first on a *heuristic* strategy based on using `sdp` first (recall that it does not provide an exact answer).

In Table 1, we compare the performance of `multivsos` for nine univariate polynomials being positive over compact intervals. More details about these benchmarks are given in (Chevillard et al., 2011, Section 6) and (Magron et al., 2018, Section 5). In this case, we use `Putinarsos`. The main difference is that we use SDP in `multivsos` instead of complex root isolation in `univsos2`. The results emphasize that `univsos2` is faster and provides more concise SOS certificates, especially for high degrees (see e.g. # 5). For # 3, we were not able to obtain a decomposition within a day of computation with `multivsos`, as meant by the symbol – in the corresponding column entries. Large values of d and 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 (Lasserre, 2001, Example 1), with $f_{12} := f^3$ and $f_{20} := f^5$. For these two benchmarks, we apply `intsos`. We use `Reznicksos` to handle M_{20} (resp. M_{100}), obtained as in Example 15 by adding 2^{-20} (resp. 2^{-100}) to the positive coefficients of the Motzkin polynomial

[1] `https://gricad-gitlab.univ-grenoble-alpes.fr/magronv/RealCertify`

[2] `http://www.home.math.uwo.ca/faculty/franz/convex`

Table 2: `multivsos` vs `RoundProject` Peyrl and Parrilo (2008) vs `RAGLib` vs `CAD` (Reznick).

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	316 479	3.99	3 274 148	3.87	0.15	0.07
f_{20}	2	20	754 168	113.	53 661 174	137.	0.16	0.03
M_{20}	3	8	4 397	0.14	3 996	0.16	0.13	0.05
M_{100}	3	8	56 261	0.26	12 200	0.20	0.15	0.03
r_2	2	4	1 680	0.11	1 031	0.12	0.09	0.01
r_4	4	4	13 351	0.14	47 133	0.15	0.32	–
r_6	6	4	52 446	0.24	475 359	0.37	623.	–
r_8	8	4	145 933	0.70	2 251 511	1.08	–	–
r_{10}	10	4	317 906	3.38	8 374 082	4.32	–	–
r_6^2	6	8	1 180 699	13.4	146 103 466	112.	10.9	–

and r_i, which is a randomly generated positive definite quartic with i variables. We implemented in Maple the projection and rounding algorithm from Peyrl and Parrilo (2008) (stated in Section 3.4) also 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.

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. This observation matches with the theoretical complexity estimates established in Proposition 11 and Theorem 13. Note that we could not solve the examples of Table 2 with less precision.

We compare with `RAGLib` Safey El Din (2007a) based on critical point methods (see e.g. Safey El Din and Schost (2003); Hong and Safey El Din (2012)) and the `SamplePoints` procedure Lemaire et al. (2005) (abbreviated as `CAD`) based on CAD Collins (1975), both available in Maple. Observe that `multivsos` can tackle examples which have large degree but a rather small number of variables ($n \leq 3$) and then return certificates of non-negativity. The runtimes are slower than what can be obtained with `RAGLib` and/or `CAD` (which in this setting have polynomial complexity when $n \leq 3$ is fixed). Note that the bit size of the certificates which are obtained here is quite large which explains this phenomenon.

When the number of variables increases, `CAD` cannot reach many of the problems we considered. Note that `multivsos` becomes not only faster but can solve problems which are not tractable by `RAGLib`.

Recall that `multivsos` relies first on solving numerically Linear Matrix Inequalities ; this is done at finite precision in time polynomial in the size of the input matrix, which, here is bounded by $\binom{n+d}{d}$. Hence, at fixed degree, that quantity evolves polynomially in n. On the other hand, the quantity which governs the behaviour of fast implementations based on the critical point method is the degree of the critical locus of some map. On the examples considered, this degree matches the worst case bound which is the Bézout number d^n. Besides, the doubly exponential theoretically proven complexity of CAD is also met on these examples.

These examples illustrate the potential of `multivsos` and more generally SDP-based methods: at fixed degree, one can hope to take advantage of fast numerical algorithms for SDP and tackle examples involving more variables than what could be achieved with more general tools.

Recall however that multivsos computes rational certificates of non-negativity in some "easy" situations: roughly speaking, these are the situations where the input polynomial lies in the interior of the SOS cone and has coefficients of moderate bit size. This fact is illustrated by Table 3.

Table 3: multivsos vs RAGLib vs CAD for non-negative polynomials which are presumably not in $\mathring{\Sigma}[X]$.

Id	n	d	multivsos		RAGLib	CAD
			τ_1 (bits)	t_1 (s)	t_2 (s)	t_3 (s)
S_1	4	24	–	–	1788.	–
S_2	4	24	–	–	1840.	–
V_1	6	8	–	–	5.00	–
V_2	5	18	–	–	1180.	–
M_1	8	8	–	–	351.	–
M_2	8	8	–	–	82.0	–
M_3	8	8	–	–	120.	–
M_4	8	8	–	–	84.0	–

This table reports on problems appearing enumerative geometry (polynomials S_1 and S_2 communicated by Sottile and appearing in the proof of the Shapiro conjecture Sottile (2000)), computational geometry (polynomials V_1 and V_2 appear in Everett et al. (2009)) and in the proof of the monotone permanent conjecture in Haglund et al. (1999) (M_1 to M_4).

We were not able to compute certificates of non-negativity for these problems which we presume do not lie in the interior of the SOS cone. This illustrates the current theoretical limitation of multivsos. These problems are too large for CAD but RAGLib can handle them. Note that some of these examples involve 8 variables ; we observed that the Bézout number is far above the degree of the critical loci computed by the critical point algorithms in RAGLib. This explains the efficiency of such tools on these problems. Recall however that RAGLib did not provide a certificate of non-negativity.

This whole set of examples illustrates first the efficiency and usability of multivsos as well as its complementarity with other more general and versatile methods. This shows also the need of further research to handle in a systematic way more general non-negative polynomials than what it does currently. For instance, we emphasize that certificates of non-negativity were computed for M_i ($1 \leq i \leq 4$) in Kaltofen et al. (2009) (see also Kaltofen et al. (2008)).

Finally, we compare the performance of multivsos (Putinarsos) on positive polynomials over basic compact semi-algebraic sets in Table 4. The first benchmark is from (Lasserre, 2001, Problem 4.6). Each benchmark f_i comes from an inequality of the Flyspeck project Hales (2013). The three last benchmarks are from Muñoz and Narkawicz (2013). 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 and multivsos can both solve large problems (involving e.g. 8 variables) but note that multivsos outputs certificates of emptiness which cannot be computed with implementations based on the critical point method. In terms of timings, multivsos is some-

Table 4: `multivsos` vs `RoundProjectPutinar` vs `RAGLib` vs `CAD` (Putinar).

Id	n	d	`multivsos`			`RoundProject`		`RAGLib`	`CAD`
			k	τ_1 (bits)	t_1 (s)	τ_2 (bits)	t_2 (s)	t_3 (s)	t_4 (s)
p_{46}	2	4	3	45 168	0.17	230 101	0.19	0.15	0.81
f_{260}	6	3	2	251 411	2.35	5 070 043	3.60	0.12	–
f_{491}	6	3	2	245 392	4.63	4 949 017	5.63	0.01	0.05
f_{752}	6	2	2	23 311	0.16	74 536	0.15	0.07	–
f_{859}	6	7	4	13 596 376	299.	2 115 870 194	5339.	5896.	–
f_{863}	4	2	1	12 753	0.13	30 470	0.13	0.01	0.01
f_{884}	4	4	3	423 325	13.7	10 122 450	16.1	0.21	–
f_{890}	4	4	2	80 587	0.48	775 547	0.56	0.08	–
butcher	6	3	2	538 184	1.36	8 963 044	3.35	47.2	–
heart	8	4	2	1 316 128	3.65	35 919 125	14.1	0.54	–
magnetism	7	2	1	19 606	0.29	16 022	0.28	434.	–

times way faster (e.g. magnetism, f_{859}) but that it is hard here to draw some general rules. Again, it is important to keep in mind the parameters which influence the runtimes of both techniques. As before, for `multivsos`, the size of the SDP to be solved is clearly the key quantity. Also, it is important to write the systems in an appropriate way also to limit the size of those matrices (e.g. write $1 - x^2 \leq 0$ to model $-1 \leq x \leq 1$). For `RAGLib`, it is way better to write $-1 \leq x$ and $x \leq 1$ to better control the Bézout bounds governing the difficulty of solving systems with purely algebraic methods. Note also that the number of inequalities increase the combinatorial complexity of those techniques.

Finally, note that `CAD` can only solve 3 benchmarks out of 10 and all in all `multivsos` and `RAGLib` solve a similar amount of problems; the latter one however does not provide certificates of emptiness. As for Table 2, `multivsos` and `RoundProjectPutinar` yield similar performance, while the former provides more concise output than the latter.

7. Conclusion and perspectives

We designed and analyzed new algorithms to compute rational SOS decompositions for several sub-classes of non-negative multivariate polynomials, including positive definite forms and polynomials positive over basic compact semi-algebraic sets. Our framework relies on SDP solvers implemented with interior-point methods. A drawback of such methods, in the context of unconstrained polynomial optimization, is that we are restricted to non-negative polynomials belonging to the interior of the SOS cone. We shall investigate the design of specific algorithms for the sub-class of polynomials lying in the border of the SOS cone. We also plan to adapt our framework, either for problems involving non-commutative polynomial data, or for alternative certification schemes, e.g. in the context of linear/geometric programming relaxations.

Appendix A. Appendix

Let $f \in \mathbb{Z}[X_1, \ldots, X_n]$ of degree d and τ be the maximum bit size of the coefficients of f in the standard monomial basis.

Let $V \subset \mathbb{C}^n$ be the algebraic set defined by

$$f = \frac{\partial f}{\partial X_2} = \cdots = \frac{\partial f}{\partial X_n} = 0 \tag{A.1}$$

By the algebraic version of Sard's theorem (see e.g. (Safey El Din and Schost, 2017, Appendix B)), when V is equidimensional and has at most finitely singular points, the projection of the set $V \cap \mathbb{R}^n$ on the X_1-axis is finite (and hence a real algebraic set of $\mathbb{R}$); we denote it by $Z_{\mathbb{R}}$. Hence, it is defined by the vanishing of some polynomial in $\mathbb{Z}[X_1]$.

The goal of this Appendix is to provide a proof of Proposition 6 which states that under the above notation and assumption, there exists a polynomial $w \in \mathbb{Z}[X_1]$ of degree $\leq d^n$ with coefficients of bit size $\leq \tau \cdot (4d+2)^{3n}$ such that its set of real roots contains $Z_{\mathbb{R}}$. To prove Proposition 6, our strategy is to rely on algorithms computing sample points in real algebraic sets: letting $C \subset V$ be a finite set of points which meet all connected components of $V \cap \mathbb{R}^n$, it is immediate that the projection of C on the X_1-axis contains $Z_{\mathbb{R}}$.
From the computation of an exact representation of such a set C, one will be able to analyze the bit size of a polynomimal whose set of roots contains $Z_{\mathbb{R}}$. We focus on algorithms based on the critical point method. Those yield the best complexity estimates which are known in theory and practical implementations reflecting these complexity gains have been obtained in Safey El Din (2007a) from e.g. Safey El Din and Schost (2003); Hong and Safey El Din (2012). Here, we focus on (Basu et al., 2006, Algorithm 13.3) since it is the more general one and it does not depend on probabilistic choices which make it easy to analyze from a bit complexity perspective. It starts by computing the polynomial

$$g = f^2 + \left(\frac{\partial f}{\partial X_2}\right)^2 + \cdots + \left(\frac{\partial f}{\partial X_n}\right)^2.$$

Observe that the set of real solutions of $g = 0$ coincides with $V \cap \mathbb{R}^n$. Next, one introduces two infinitesimals ϵ and η (see (Basu et al., 2006, Chap. 2) for an introduction on Puiseux series and infinitesimals). Consider the polynomial:

$$g_1 = g + \left(\eta(X_1^2 + \cdots + X_{n+1}^2) - 1\right)^2.$$

Its vanishing set over $\mathbb{R}\langle\eta\rangle^{n+1}$ corresponds to the intersection of the lifting of the vanishing set of g in $\mathbb{R}^n$ with the hyperball of $\mathbb{R}\langle\eta\rangle^{n+1}$ centered at the origin of radius $\frac{1}{\eta}$.
Let d_i be the degree of g_1 in X_i. Without loss of generality, up to reordering the variables, we assume that $d_1 \geq d_2 \geq \cdots \geq d_n$; we assume that after this process X_1 has been sent to X_k. Now, we let

$$h = g_1(1-\varepsilon) + \varepsilon(X_1^{2(d_1+1)} + \cdots + X_n^{2(d_n+1)} + X_{n+1}^6 - (n+1)\zeta^{d+1})$$

We finally focus on the polynomial system:

$$h = \frac{\partial h}{\partial X_2} = \cdots = \frac{\partial h}{\partial X_{n+1}} = 0$$

The rationale behind the last infinitesimal deformation is twofold (see (Basu et al., 2006, Chap. 12 and Chap. 13)):

- the algebraic set defined by the vanishing of h is smooth ;

- the above polynomial system is finite and forms a Gröbner basis G for any degree lexicographical ordering with $X_1 > \cdots > X_{n+1}$.

Besides, (Basu et al., 2006, Prop. 13.30) states that taking the limits (when infinitesimals tend to zero) of projections on the $(X_1, \ldots, X_n)$-space of a finite set of points meeting each connected component of the real algebraic set defined by $h = 0$ provides a finite set of points in the real algebraic set defined by $g = 0$.

In our situation, we do not need to to go into such details. We only need to compute a non-zero polynomial $w \in \mathbb{Z}[X_k]$ whose set of real roots contains $Z_{\mathbb{R}}$. Using Stickelberger's theorem (Basu et al., 2006, Theorem 4.98) and the process for computing limits in (Basu et al., 2006, Algoroithm 12.14) and Rouillier et al. (2000), it suffices to compute the characteristic polynomial of the multiplication operator by X_k in the ring of polynomials with coefficients in $\mathbb{Q}[\eta, \zeta]$ quotiented by the ideal $\langle G \rangle$. This is done using (Basu et al., 2006, Algorithm 12.9).

In order to analyze the bit size of the coefficients of the output characteristic polynomial, we need to bound the bit size of the entries in the matrix output by (Basu et al., 2006, Algorithm 12.9). Following the discussion in the complexity analysis of (Basu et al., 2006, Algorithm 13.1), we deduce that the coefficients of these entries have bit size dominated by $\tau\,(2(2d + 1))^{2n}$. Besides, this matrix has size bounded by $(2(2d + 1))^{2n}$. We deduce that the coefficients of its characteristic polynomial have bit size bounded by $(2(2d + 1))^{3n}$.

References

Artin, E., 1927. Über die Zerlegung definiter Funktionen in Quadrate. Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg 5 (1), 100–115.

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

Bai, Z., Demmel, J., McKenney, A., 1989. On Floating Point Errors in Cholesky. Tech. rep., (LAPACK Working Note 14).

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

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

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

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

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

Basu, S., Pollack, R., Roy, M.-F., 1998. 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.

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

Blekherman, G., 2006. There are significantly more nonegative polynomials than sums of squares. Israel Journal of Mathematics 153 (1), 355–380.

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

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

Choi, M. D., Lam, T. Y., Reznick, B., 1995. Sums of squares of real polynomials. Vol. 58 of Proc. Sympos. Pure Math. Amer. Math. Soc., pp. 103–126.

Collins, G. E., 1975. Quantifier elimination for real closed fields by cylindrical algebraic decompostion. In: ATFL 2nd GI Conf. Kaiserslautern. pp. 134–183.

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

Everett, H., Lazard, D., Lazard, S., Safey El Din, M., 2009. The voronoi diagram of three lines. Discrete & Computational Geometry 42 (1), 94–130.

Golub, G. H., Loan, C. F. V., 1996. Matrix Computations (3rd Ed.). Johns Hopkins University Press, Baltimore, MD, USA.

Greuet, A., Guo, F., Safey El Din, M., Zhi, L., 2012. Global optimization of polynomials restricted to a smooth variety using sums of squares. Journal of Symbolic Computation 47 (5), 503 – 518.
URL `http://www.sciencedirect.com/science/article/pii/S0747717111001957`

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

Grigoriev, D., Vorobjov, N., 1988. Solving systems of polynomials inequalities in subexponential time. Journal of Symbolic Computation 5, 37–64.

Grötschel, M., Lovász, L., Schrijver, A., 1993. Geometric Algorithms and Combinatorial Optimization, second corrected edition Edition. Vol. 2 of Algorithms and Combinatorics. Springer.

Guo, F., Kaltofen, E. L., Zhi, L., 2012. Certificates of Impossibility of Hilbert-Artin Representations of a Given Degree for Definite Polynomials and Functions. In: Proceedings of the 37th International Symposium on Symbolic and Algebraic Computation. ISSAC '12. ACM, New York, NY, USA, pp. 195–202.
URL `http://doi.acm.org/10.1145/2442829.2442859`

Guo, F., Safey El Din, M., Zhi, L., 2010. 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. ACM, New York, NY, USA, pp. 107–114.
URL `http://doi.acm.org/10.1145/1837934.1837960`

Guo, Q., Safey El Din, M., Zhi, L., 2013. Computing rational solutions of linear matrix inequalities. In: Proceedings of the 38th International Symposium on Symbolic and Algebraic Computation. ISSAC '13. ACM, New York, NY, USA, pp. 197–204.

Haglund, J., Ono, K., Wagner, D. G., 1999. Theorems and Conjectures Involving Rook Polynomials with Only Real Zeros. Springer US, Boston, MA, pp. 207–221.

Hales, T. C., 2013. The flyspeck project.
URL `http://code.google.com/p/flyspeck`

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

Hillar, C., 2009. Sums of squares over totally real fields are rational sums of squares. Proceedings of the American Mathematical Society 137 (3), 921–930.

Hong, H., Safey El Din, M., 2012. Variant quantifier elimination. Journal of Symbolic Computation 47 (7), 883–901.

J.B. Lasserre, M. Laurent, B. Mourrain, P. Rostalski and P. Trébuchet, 2013. Moment Matrices, Border Bases and Real Radical Computation. J. Symb. Computation. 51, 63–85.

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

Kaltofen, E., Yang, Z., Zhi, L., 2009. A Proof of the Monotone Column Permanent (MCP) Conjecture for Dimension 4 via Sums-of-squares of Rational Functions. In: Proceedings of the 2009 Conference on Symbolic Numeric Computation. SNC '09. ACM, New York, NY, USA, pp. 65–70.

Kaltofen, E. L., Li, B., Yang, Z., Zhi, L., 2008. Exact certification of global optimality of approximate factorizations via rationalizing sums-of-squares with floating point scalars. In: Proceedings of the 21st International Symposium on Symbolic and Algebraic computation. ISSAC '08. ACM, New York, NY, USA, pp. 155–164.

Kaltofen, E. L., Li, B., Yang, Z., Zhi, L., 2012. Exact certification in global polynomial optimization via sums-of-squares of rational functions with rational coefficients. Journal of Symbolic Computation 47 (1), 1 – 15.

Laplagne, S., 2018. Facial reduction for exact polynomial sum of squares decompositions. To appear in Mathematics of Computation, `https://arxiv.org/abs//1810.04215`.

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

Laurent, M., 2009. Sums of squares, moment matrices and optimization over polynomials. Springer.

Lemaire, F., Maza, M. M., Xie, Y., Sep. 2005. The RegularChains Library in MAPLE. SIGSAM Bull. 39 (3), 96–97.

Lombardi, H., Perrucci, D., Roy, M.-F., 2018. An elementary recursive bound for effective Positivstellensatz and Hilbert 17-th problem. To appear at *Memoirs of the AMS*.

Magron, V., Safey El Din, M., 2018a. On Exact Polya and Putinar's Representations. In: ISSAC'18: Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation. ACM, New York, NY, USA.

Magron, V., Safey El Din, M., 2018b. RealCertify: a Maple package for certifying non-negativity. In: ISSAC'18: Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation. ACM, New York, NY, USA.

Magron, V., Safey El Din, M., Schweighofer, M., 2018. Algorithms for weighted sum of squares decomposition of

non-negative univariate polynomials. Journal of Symbolic Computation.
Mignotte, M., 1992. Mathematics for Computer Algebra. Springer-Verlag New York, Inc., New York, NY, USA.
Muñoz, C., Narkawicz, A., 2013. Formalization of Bernstein Polynomials and Applications to Global Optimization. J. Aut. Reasoning 51 (2), 151–196.
Nakata, M., 2010. A numerical evaluation of highly accurate multiple-precision arithmetic version of semidefinite programming solver: SDPA-GMP, -QD and -DD. In: CACSD. pp. 29–34.
Nie, J., Ranestad, K., Sturmfels, B., 2010. The algebraic degree of semidefinite programming. Mathematical Programming 122 (2), 379–405.
Nie, J., Schweighofer, M., 2007. On the complexity of Putinar's Positivstellensatz. Journal of Complexity 23 (1), 135 – 150.
P. A. Parrilo, 2000. Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization. Ph.D. thesis, California Inst. Tech.
Peyrl, H., Parrilo, P., 2008. Computing sum of squares decompositions with rational coefficients. Theoretical Computer Science 409 (2), 269–281.
Putinar, M., 1993. Positive polynomials on compact semi-algebraic sets. Indiana University Mathematics Journal 42 (3), 969–984.
Quarez, R., Dec 2010. Tight bounds for rational sums of squares over totally real fields. Rendiconti del Circolo Matematico di Palermo 59 (3), 377–388.
Reznick, B., 1978. Extremal PSD forms with few terms. Duke Mathematical Journal 45 (2), 363–374.
Reznick, B., Dec 1995. Uniform denominators in Hilbert's seventeenth problem. Mathematische Zeitschrift 220 (1), 75–97.
URL https://doi.org/10.1007/BF02572604
Rouillier, F., Roy, M.-F., Safey El Din, M., 2000. Finding at least one point in each connected component of a real algebraic set defined by a single equation. Journal of Complexity 16 (4), 716–750.
Safey El Din, M., 2007a. RAGlib – A library for real solving polynomial systems of equations and inequalities. http://www-polsys.lip6.fr/~safey/RAGLib/distrib.html.
Safey El Din, M., 2007b. Testing sign conditions on a multivariate polynomial and applications. Mathematics in Computer Science 1 (1), 177–207.
Safey El Din, M., Schost, E., 2003. Polar varieties and computation of one point in each connected component of a smooth real algebraic set. In: ISSAC'03. ACM, pp. 224–231.
Safey El Din, M., Schost, É., 2017. A nearly optimal algorithm for deciding connectivity queries in smooth and bounded real algebraic sets. Journal of the ACM (JACM) 63 (6), 48.
Safey El Din, M., Zhi, L., Sep. 2010. Computing Rational Points in Convex Semialgebraic Sets and Sum of Squares Decompositions. SIAM J. on Optimization 20 (6), 2876–2889.
URL http://dx.doi.org/10.1137/090772459
Sottile, F., 2000. Real schubert calculus: Polynomial systems and a conjecture of shapiro and shapiro. Experimental Mathematics 9 (2), 161–182.
URL https://doi.org/10.1080/10586458.2000.10504644
Sturm, J. F., 1998. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones.
Wütrich, H. R., 1976. Ein entschiedungsverfahren für die theorie der reell-abgeschlossenen körper. In: Lecture Notes in Computer Science. Vol. 43. pp. 138–162.
Yamashita, M., Fujisawa, K., Nakata, K., Nakata, M., Fukuda, M., Kobayashi, K., Goto, K., 2010. A high-performance software package for semidefinite programs : SDPA7. Tech. rep., Dept. of Information Sciences, Tokyo Inst. Tech.
URL http://www.optimization-online.org/DB_FILE/2010/01/2531.pdf