SUM OF SQUARES DECOMPOSITIONS OF POLYNOMIALS OVER THEIR GRADIENT IDEALS WITH RATIONAL COEFFICIENTS

VICTOR MAGRON*, MOHAB SAFEY EL DIN†, AND TRUNG-HIEU VU‡

Abstract. Assessing non-negativity of multivariate polynomials over the reals, through the computation of *certificates of non-negativity*, is a topical issue in polynomial optimization. This is usually tackled through the computation of *sum of squares decompositions* which rely on efficient numerical solvers for semi-definite programming.

This method faces two difficulties. The first one is that the certificates obtained this way are *approximate* and then non-exact. The second one is due to the fact that not all non-negative polynomials are sums of squares.

In this paper, we build on previous works by Parrilo, Nie, Demmel and Sturmfels who introduced certificates of non-negativity modulo *gradient ideals*. We prove that, actually, such certificates can be obtained *exactly*, over the rationals if the polynomial under consideration has rational coefficients and we provide *exact* algorithms to compute them. We analyze the bit complexity of these algorithms and deduce bitsize bounds of such certificates.

Key words. Non-negative polynomial, sum of squares decomposition, gradient ideal, zero-dimensional and radical ideal, Gröbner basis, bit complexity

AMS subject classifications. 14P05, 11E25, 13P10

1. Introduction.

We denote by $\mathbb{Q}$ (resp. $\mathbb{R}$) the field of rational (resp. real) numbers and by $\boldsymbol{x}$ the n-tuple of variables $(x_1, \ldots, x_n)$. Let $\mathbb{K}$ be a field, we denote by $\mathbb{K}[\boldsymbol{x}]$ the polynomial ring with base field $\mathbb{K}$ and variables $\boldsymbol{x}$. For a polynomial f of degree d in $\mathbb{Q}[\boldsymbol{x}]$, we consider the problem of computing certificates of non-negativity of f over $\mathbb{R}^n$. This is a central issue in polynomial optimization as minimizing a polynomial f boils down to maximizing λ such that $f - \lambda$ is non-negative over $\mathbb{R}^n$. This hard non-negativity constraint can be replaced by a more tractable one that is $f - \lambda$ is a sum of squares (SOS) of polynomials.

Prior works. Computing certificates of non-negativity is usually done by decomposing f as an SOS of polynomials or rational fractions. It is well-known that all non-negative univariate polynomials with real coefficients can be decomposed as a sum of squares of polynomials. Also, any non-negative univariate polynomial f with rational coefficients can be decomposed as a *weighted* sum of squares with rational coefficients, i.e. $f = \sum_i c_i s_i^2$ where s_i has rational coefficients and c_i is a positive rational [21, 34]. Further, by SOS decompositions with rational coefficients, we mean *weighted* SOS decompositions with rational coefficients. Several algorithms already compute such SOS decomposition with rational coefficients of non-negative univariate polynomials with rational coefficients (see [42, 10]) and bit complexity and bitsize estimates are given in [27].

The multivariate case is more difficult. Following the seminal works by [22, 31], hierarchies of semi-definite programs yield *approximations* of weighted SOS decompositions of positive polynomials. Several heuristics have been proposed to lift such approximations to exact SOS decompositions of the input polynomial, starting with [33] and followed by [18, 19, 20]. Note that algorithms in [18, 20] allow us to compute SOS decompositions on some degenerate examples or compute SOS of rational frac-

*Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS), Toulouse, France (vmagron@laas.fr)

†Sorbonne Université, LIP6, Paris, France (mohab.safey@lip6.fr)

‡Sorbonne Université, LIP6, Paris, France (trung-hieu.vu@lip6.fr)

tions. Complexity issues are studied through the prism of perturbation-compensation techniques to compute SOS decompositions in the interior of the SOS cone [24, 25, 26].

Still, computing *exact* certificates of non-negativity is especially hard because of the two following reasons. The first one is that there exist non-negative polynomials which are not SOS, for example, Motzkin's polynomial and Robinson's polynomial. Moreover, Blekherman proved in [8] that there are many more non-negative polynomials in $\mathbb{R}[\boldsymbol{x}]$ than SOS polynomials. The second one is that, even if a given polynomial with rational coefficients is SOS, there is no guarantee that there exists an SOS decomposition involving rational coefficients, as established in [41]. Still, general algorithms for computing such exact certificates by means of sums of squares decompositions have been designed, either for computing sums of squares decompositions with *rational* coefficients [40] or with algebraic numbers by computing exact solutions to semi-definite programs [17] but suffer from a high complexity.

Alternative certificates of non-negativity, for instance, SAGE/SONC polynomials [28, 44] can also be used but they face similar issues to the ones met by SOS techniques when it comes with generality.

Deciding non-negativity over an arbitrary semi-algebraic set of a polynomial $f \in \mathbb{Q}[\boldsymbol{x}]$ can be done exactly using computer algebra algorithms. The best complexities for such a decision procedure are achieved by algorithms making effective the so-called critical point method [16, 6], further practical developments in [2, 3, 4, 38] and their applications in polynomial optimization in [14, 15, 5]. Note that, even if these algorithms are exact (i.e. their results are exact provided that no bug has been encountered), they do not provide a certificate assessing non-negativity which can be checked a posteriori since these are root-finding algorithms. Their complexities are exponential in the dimension of the ambient space as they reduce the input problem to computing finitely many critical points of some well-chosen maps, hence considering *gradient ideals*.

Hence, all in all, such gradient ideals can be used to reduce the dimension of the set over which certifying non-negativity can be done. Under some assumptions, this idea is translated in [32] to an algorithm assessing the non-negativity of a given $f \in \mathbb{R}[\boldsymbol{x}]$. Precisely, assuming the gradient ideal $\mathcal{I}_{grad}(f)$ (which is the set of all algebraic combinations of the partial derivatives of f) is zero-dimensional[1] and radical[2], and that f reaches its infimum over $\mathbb{R}^n$, this algorithm computes an SOS decomposition of f in the quotient ring $\mathbb{R}[\boldsymbol{x}]/\mathcal{I}_{grad}(f)$ (or, in other words, an SOS decomposition of f modulo $\mathcal{I}_{grad}(f)$), i.e., f is written as

$$c_1 s_1^2 + \cdots + c_k s_k^2 + \sum_{i=1}^{n} q_i \frac{\partial f}{\partial x_i}$$

where the s_i's and the q_i's lie in $\mathbb{R}[\boldsymbol{x}]$ and the c_i's are positive in $\mathbb{R}$. A similar result slightly relaxing the above assumptions is given in [30]. Note that when f has coefficients in $\mathbb{Q}$, there is no given guarantee that an SOS decomposition of it in $\mathbb{Q}[\boldsymbol{x}]/\mathcal{I}_{grad}(f)$ will have rational coefficients too (i.e., the s_i's and the q_i's have coefficients in $\mathbb{Q}$ and the c_i's lie in $\mathbb{Q}$).

Contributions. We build on the results of [32, 30], to investigate this issue when $f \in \mathbb{Q}[\boldsymbol{x}]$. *We assume in the whole paper that the gradient ideal associated to f is radical and zero-dimensional and that f reaches its infimum over* $\mathbb{R}^n$. We summarize our contributions as follows.

[1] This means that it has finitely many complex solutions.

[2] This means that if $h^k \in \mathcal{I}_{grad}(f)$ for some $k \in \mathbb{N} - \{0\}$, then $h \in \mathcal{I}_{grad}(f)$.

Existence of certificates of non-negativity with rational certificates. Under the above assumptions, we prove that f is non-negative over $\mathbb{R}^n$ if and only if f is an SOS of polynomials with rational coefficients over the quotient ring $\mathbb{Q}[\boldsymbol{x}]/\mathcal{I}_{grad}(f)$ (see Theorem 3.1). The new ingredients beyond those used in [32, 30] are a reduction to the univariate case thanks to the so-called shape position (see Lemma 2.2) as well as bit complexity analysis of algorithms providing zero-dimensional rational parametrization of the gradient variety (Corollary 2.3) and algorithms providing weighted rational SOS decompositions of univariate rational polynomials (Theorems 17 and 24 from [27]). Interestingly, Theorem 3.1 can be applied to Robinson's polynomial [36], which is not an SOS of polynomials (see Example 3.5), as well as Scheiderer's polynomial [41], which is an SOS of polynomials with real coefficients but not an SOS of polynomials with rational coefficients (see Example 3.6).

The next problem we tackle is to design algorithms computing such certificates of non-negativity, estimate their bit complexity.

To measure the *bitsize* of a polynomial with rational coefficients, we will use its *height*, defined as follows. The bitsize of an integer b is denoted by $ht(b) := \lfloor \log_2(|b|) \rfloor + 1$ with $ht(0) := 1$, where $\log_2$ is the logarithm in base 2. Given $a \in \mathbb{Z}$ and $b \in \mathbb{Z}$ with $b \neq 0$ and $\gcd(a, b) = 1$, we define $ht\left(\frac{a}{b}\right) = \max\{ht(a), ht(b)\}$. For a non-zero polynomial f with rational coefficients, the bitsize $ht(f)$ is defined as the maximum bitsize of the non-zero coefficients of f. For two mappings $p, q : \mathbb{N}^m \to \mathbb{R}$, the expression "$p(v) = O(q(v))$" means that there exists $b \in \mathbb{N}$ such that $p(v) \leq bq(v)$, for all $v \in \mathbb{N}^m$. We use the notation $p(v) = \widetilde{O}(q(v))$ in order to indicate that $p(v) = O(q(v) \log^k q(v))$ for some $k \in \mathbb{N}$.

Algorithms and bit complexity estimates. From the proof of Theorem 3.1, we derive an algorithm (Algorithm 3.1), named `sosgradientshape`, to compute an SOS decomposition of polynomials modulo the gradient ideal of f. This algorithm can certify non-negativity of polynomials which cannot be tackled with a direct SOS approach. We also investigate the bit complexity of `sosgradientshape`. We prove that, given as input an n-variate polynomial $f \in \mathbb{Q}[\boldsymbol{x}]$ of degree d with maximum bitsize of its coefficients τ, `sosgradientshape` uses

$$\widetilde{O}((\tau + n + d)^2 d^{6n} + (\tau + n + d) d^{6n+4})$$

boolean operations. This is better than the complexity estimates given in [26, Theorem 12], where the reported number of boolean operations is: $\widetilde{O}(\tau^2 (4d + 2)^{15n+6})$.

We design a variant of Algorithm `sosgradientshape`, named `sosgradient`, and which, on input $f \in \mathbb{Q}[\boldsymbol{x}]$ as above, decomposes it as a sum of *rational fractions* modulo the gradient ideal associated to f. We prove that this variant uses

$$\widetilde{O}\left((\tau + n + d) d^{4n+4}\right)$$

boolean operations, hence with better complexity than Algorithm `sosgradientshape`.

Both algorithms have been implemented using the Maple computer algebra system. We report on practical experiments showing that they can already assess the non-negativity of numerous polynomials which are out of reach of, e.g., hybrid methods computing sums of squares decompositions such as [24]. We emphasize that such complexity estimates are of interest to the polynomial optimization community as they give degree bounds for the SOS multipliers required when using the variant of the so-called "Moment-SOS hierarchy" (also called Lasserre's hierarchy [22]) to minimize polynomials over their gradient ideals [30]. Indeed, such degree bounds translate to convergence rates for the underlying optimization scheme and allow one to estimate the overall computational cost complexity. More importantly, our practical experiments show that the algorithm `sosgradient` can assess the non-negativity of multivariate polynomials of a large set of examples which are out of reach of the state of the art (when both the number of variables and degree increase).

Structure of the paper. In the next section, we recall basic notions and fundamental results used in the paper. In Section 3, we prove the existence of an SOS of polynomials modulo the gradient ideal of f, introduce Algorithm `sosgradientshape` and analyze its bit complexity. The results for decomposing f as an SOS of rational fractions modulo the gradient ideal are presented in Section 4. Practical experiments are given in the last section.

2. Preliminaries. This section recalls basic notions and results from algebraic geometry, computational commutative algebra, and complexity analysis. Further details can be found in [11].

Let $\mathbb{K}$ be a field. An additive subgroup $\mathcal{I}$ of $\mathbb{K}[\boldsymbol{x}]$ is said to be an *ideal* of $\mathbb{K}[\boldsymbol{x}]$ if $hg \in \mathcal{I}$ for any $h \in \mathcal{I}$ and $g \in \mathbb{K}[\boldsymbol{x}]$. Given $g_1, \dots, g_r$ in $\mathbb{K}[\boldsymbol{x}]$, we denote by $\langle g_1, \dots, g_r \rangle$ the ideal generated by $g_1, \dots, g_r$. If $\mathcal{I}$ is an ideal of $\mathbb{K}[\boldsymbol{x}]$ then, according to Hilbert's basis theorem (see, e.g., [11, Theorem 4]), there exist $g_1, \dots, g_r \in \mathbb{K}[\boldsymbol{x}]$ such that $\mathcal{I} = \langle g_1, \dots, g_r \rangle$.

Let $\mathcal{I}$ be an ideal of $\mathbb{R}[\boldsymbol{x}]$. The algebraic variety associated to $\mathcal{I}$ is defined as

$$V(\mathcal{I}) := \{x \in \mathbb{C}^n : \forall g \in \mathcal{I}, g(\boldsymbol{x}) = 0\}.$$

The real algebraic variety associated to $\mathcal{I}$ is $V^{\mathbb{R}}(\mathcal{I}) = V(\mathcal{I}) \cap \mathbb{R}^n$. Recall that the ideal $\mathcal{I}$ is *zero-dimensional* if the cardinality $\#V(\mathcal{I})$ is finite, and that $\mathcal{I}$ is *radical* if

$$g^k \in \mathcal{I} \text{ for some } k \in \mathbb{N} \Longrightarrow g \in \mathcal{I}.$$

We emphasize that $V(\mathcal{I})$ being finite (i.e. $\mathcal{I}$ being zero-dimensional) is a stronger assumption than $V^{\mathbb{R}}(\mathcal{I})$ being finite. It is worth noting here that if $\mathcal{I}$ is zero-dimensional then we can get a bound on the expected cardinality of $V(\mathcal{I})$ from Bezout's theorem.

Let f be a polynomial in $\mathbb{R}[\boldsymbol{x}]$. Recall that the *gradient ideal* $\mathcal{I}_{grad}(f)$ of f is the ideal generated by all partial derivatives of f in $\mathbb{R}[\boldsymbol{x}]$, i.e.,

$$\mathcal{I}_{grad}(f) := \left\langle \frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n} \right\rangle.$$

The (resp. *real*) *gradient variety* associated to f is respectively the (resp. real) algebraic variety associated to $\mathcal{I}_{grad}(f)$. We denote them respectively by $V_{grad}(f)$ and $V^{\mathbb{R}}_{grad}(f)$. Let $\mathbb{K}$ be a real field contained in $\mathbb{R}$. One says that f is a (weighted) *sum of squares* (SOS) of polynomials in $\mathbb{K}[\boldsymbol{x}]$ if there exist polynomials $q_1, \dots, q_s$ in $\mathbb{K}[\boldsymbol{x}]$ and positive numbers $c_1, \dots, c_s$ in $\mathbb{K}$ such that $f = \sum_{j=1}^s c_j q_j^2$. Furthermore, f is an SOS of polynomials over the quotient ring $\mathbb{K}[\boldsymbol{x}]/\mathcal{I}_{grad}(f)$ if there exists $g \in \mathcal{I}_{grad}(f)$ such that $f - g$ is SOS in $\mathbb{K}[\boldsymbol{x}]$, i.e., f can be decomposed as follows:

$$f = \sum_{j=1}^{s} c_j q_j^2 + \sum_{i=1}^{n} \phi_i \frac{\partial f}{\partial x_i},$$

for some polynomials $q_1, \dots, q_s, \phi_1, \dots, \phi_s$ in $\mathbb{K}[\boldsymbol{x}]$ and positive numbers $c_1, \dots, c_s$ in $\mathbb{K}$.

Clearly, if f is SOS over $\mathbb{R}[\boldsymbol{x}]/\mathcal{I}_{grad}(f)$ then f is non-negative over $V_{grad}^{\mathbb{R}}(f)$. We recall below [32, Theorem 1].

Let f be a polynomial in $\mathbb{R}[\boldsymbol{x}]$. Suppose that the gradient ideal $\mathcal{I}_{grad}(f)$ is zero-dimensional and radical. Then, f is non-negative over $V_{grad}^{\mathbb{R}}(f)$ if and only if f is SOS over the quotient ring $\mathbb{R}[\boldsymbol{x}]/\mathcal{I}_{grad}(f)$.

We now recall useful results in the univariate case. It is well-known that $f \in \mathbb{R}[t]$ is non-negative over $\mathbb{R}$ if and only if f is SOS. This property holds also for polynomials with coefficients in a subfield $\mathbb{K}$ of $\mathbb{R}$. More precisely, we have the following theorem:

THEOREM 2.1 ([21, 34]). *Let $\mathbb{K}$ be a subfield of $\mathbb{R}$ and $f \in \mathbb{K}[t]$. Then, f is non-negative over $\mathbb{R}$ if and only if f admits a weighted SOS decomposition of polynomials in $\mathbb{K}[t]$, i.e., there exists a positive integer s, non-negative numbers $c_1, \ldots, c_s \in \mathbb{K}$ and polynomials $g_1, \ldots, g_s \in \mathbb{K}[t]$, such that $f = \sum_{j=1}^{s} c_j g_j^2$.*

Let $\mathbb{K}$ be a field and $<$ be a monomial ordering on $\mathbb{K}[\boldsymbol{x}]$ and $\mathcal{I} \neq \{0\}$ be an ideal. We denote by $LT_<(\mathcal{I})$ the set of all leading terms $LT_<(g)$ of $g \in \mathcal{I}$, and by $\langle LT_<(\mathcal{I})\rangle$ the ideal generated by the elements of $LT_<(\mathcal{I})$.

A subset $G = \{g_1, \ldots, g_r\}$ of $\mathcal{I}$ is said to be a *Gröbner basis* of $\mathcal{I}$ w.r.t. some monomial order $<$ if

$$\langle LT_<(g_1), \ldots, LT_<(g_r)\rangle = \langle LT_<(\mathcal{I})\rangle\,.$$

Note that every ideal in $\mathbb{K}[\boldsymbol{x}]$ has a Gröbner basis. A Gröbner basis G is *reduced* if the two following conditions hold: the leading coefficient of g is 1, for all $g \in G$; there are no monomials of g lying in $\langle LT_<(G) \setminus \{g\}\rangle$. Every ideal $\mathcal{I}$ has a unique reduced Gröbner basis. We refer the reader to [11] for more details. Further, when the monomial order $<$ is clear from the context, we omit as a subscript in the above notation.

Assume that $\mathcal{I}$ is a zero-dimensional and radical ideal in $\mathbb{Q}[\boldsymbol{x}]$ and that G is the reduced Gröbner basis of $\mathcal{I}$ with respect to the lexicographical order $x_1 <_{lex} \cdots <_{lex} x_n$. One says that $\mathcal{I}$ is in *shape position* if G has the following form:

$$G = [w, x_2 - v_2, \ldots, x_n - v_n], \tag{2.1}$$

where $w, v_2, \ldots, v_n$ are polynomials in $\mathbb{K}[x_1]$ and $\deg w = \#V(\mathcal{I})$.

The following lemma, named Shape Lemma, gives us a criteria for the shape position of an ideal.

LEMMA 2.2 (Shape Lemma, [13]). *Let $\mathcal{I}$ be a zero-dimensional and radical ideal and $<_{lex}$ be a lexicographic monomial order in $\mathbb{Q}[\boldsymbol{x}]$. If $V(\mathcal{I})$ is the union of δ points in $\mathbb{C}^n$ with distinct x_1-coordinates, then $\mathcal{I}$ is in shape position as in (2.1), where $v_2, \ldots, v_n$ are polynomials in $\mathbb{Q}[x_1]$ of degrees at most $\delta - 1$.*

Let V be a zero-dimensional algebraic subset of $\mathbb{C}^n$, $\delta := \#V$. A *zero-dimensional rational parametrization* $\mathcal{Q} = ((w, \kappa_1, \ldots, \kappa_n), \lambda)$ of V consists in $n+1$ univariate polynomials $w, \kappa_1, \ldots, \kappa_n$ in $\mathbb{Q}[t]$, where w' is the derivative of w, such that w is monic and square-free, $\deg \kappa_i < \deg w$, for $i = 1, \ldots, n$, and a $\mathbb{Q}$-linear form λ in n variables satisfying $\lambda(\kappa_1, \ldots, \kappa_n) = tw' \mod w$, such that

$$V = \left\{ \left(\frac{\kappa_1(t)}{w'(t)}, \ldots, \frac{\kappa_n(t)}{w'(t)}\right) : w(t) = 0 \right\}.$$

The condition on the linear form λ states that the roots of w are precisely the values taken by λ on V, and that λ separates V, i.e., $\lambda(x) \neq \lambda(y)$ for any distinct pair x, y in V.

Let f be in $\mathbb{Q}[\boldsymbol{x}]$ of degree d and bitsize τ. Assume that $V_{grad}(f)$ is finite. By applying [39, Corollary 2] to the system of partial derivatives, we obtain the following corollary (Corollary 2.3) which states that there exists an algorithm computing a zero-dimensional rational parametrization of $V_{grad}(f)$ and provides bit complexity estimates for when applying the algorithm in [39] to gradient ideals. The proof of Corollary 2.3 is straightforward from [39, Corollary 2] and is then postponed to Appendix A.

COROLLARY 2.3. *Assume that $V_{grad}(f)$ is finite. There exists an algorithm that takes f as in input, and that produces one of the following outputs:*

a) *either a zero-dimensional rational parametrization of $V_{grad}(f)$;*
b) *or a zero-dimensional rational parametrization of degree less than that of $V_{grad}(f)$;*
c) *or fails.*

In any case, the algorithm uses

$$\widetilde{O}\Big(n^2(d+\tau)d^{2n+1}\binom{n+d}{d}\Big) \tag{2.2}$$

boolean operations. Moreover, the polynomials $w, \kappa_1, \ldots, \kappa_n$ involved in the zero-dimensional rational parametrization output have degree at most $(d-1)^n$ and bitsize $\widetilde{O}\left((d+\tau+n)(d-1)^n\right)$.

Assume that $\mathcal{Q} = ((w, \kappa_1, \ldots, \kappa_n), x_1)$ is a zero-dimensional rational parametrization of $V_{grad}(f)$ given by the algorithm from Corollary 2.3. The following lemma (Lemma 2.4) and its proof point out the explicit shape position of $\mathcal{I}_{grad}(f)$. Moreover, the degree and the bit complexity of the involved polynomials are estimated.

LEMMA 2.4. *There exist polynomials $w, v_2, \ldots, v_n$ in $\mathbb{Q}[x_1]$ satisfying $\deg v_i < \deg w$, for $i = 2, \ldots, n$, such that $\mathcal{I}_{grad}(f) = \langle w, x_2 - v_2, \ldots, x_n - v_n\rangle$. Furthermore, to compute $w, v_2, \ldots, v_n$, we use*

$$\widetilde{O}\left((\tau+n+d)^2 d^{6n}\right) \tag{2.3}$$

boolean operations. Their degrees are at most $(d-1)^n$ and their maximum bitsizes are bounded from above by $\widetilde{O}((\tau+n+d)d^{3n})$.

Proof. Here we give only the proof of the degree estimate. The proof of the bit complexity is routine but rather technical and postponed to Appendix B.

Because w is square-free and w' is the derivative of w, one sees that the gcd of w and w' is 1. From the extended Euclidean algorithm [43, Algorithm 3.14], there exist two Bézout coefficients of w and w', namely a, b in $\mathbb{Q}[x_1]$, such that $aw + bw' = 1$. For $i = 2, \ldots, n$, we see that $w'x_i(t) = \kappa_i(t)$ for any t satisfying $w(t) = 0$. As $\deg \kappa_i \le \deg w$ and the linear form $\lambda = x_1$ separates V, we have $w'x_i = \kappa_i$. This yields $bw'x_i = b\kappa_i$. Since $bw' = 1 - aw$, we observe that $x_i - awx_i = b\kappa_i$ and, hence, $x_i = b\kappa_i \mod w$. By denoting $v_i := b\kappa_i \mod w$, we obtain $w, v_2, \ldots, v_n$ which are the desired polynomials. ∎

The two following lemmas establish the bit complexity of Euclidean division algorithm and the extended Euclidean algorithm for univariate polynomials over $\mathbb{Z}$ which will be used later on (in Proposition 3.11) to investigate the bit complexity of our algorithms.

LEMMA 2.5. *Let a, b be polynomials in $\mathbb{Z}[t]$, with $\deg a = d \ge m = \deg b$, and τ an upper bound of $ht(a)$ and $ht(b)$. To compute the quotient q and the remainder r of the* *division of a by b, we use the Euclidean division algorithm [43, Algorithm 2.5]. Then, this algorithm uses* $O\left(m\tau(d-m)^2\right)$ *boolean operations. Furthermore, both bitsizes of* q *and* r *are bounded from above by* $O\left(\tau(d-m)\right)$.

Again, the proof of Lemma 2.5 is routine but rather technical. We postpone it to Appendix C.

Denote by $\mathbb{Q}(x_1)$ the field of rational fractions in variable x_1 with coefficients in $\mathbb{Q}$. With the lexicographic monomial order $x_2 < \cdots < x_n$, we consider the standard (multivariate) division [11, Ch. 2, Sec 3.] of $g \in \mathbb{Q}[x_1][x_2, \ldots, x_n]$ by the list $[x_2 - \frac{a_2}{a_0}, \ldots, x_n - \frac{a_n}{a_0}]$, with $a_0, a_2, \ldots, a_n \in \mathbb{Q}[x_1]$. To compute the quotients $\phi_2, \ldots, \phi_n \in \mathbb{Q}(x_1)[x_2, \ldots, x_n]$ and remainder $r \in \mathbb{Q}(x_1)$ such that $g = \sum_{i=2}^n \phi_i(x_i - \frac{a_i}{a_0}) + r$, we iterate classical univariate divisions by $x_i - \frac{a_i}{a_0}$ for $2 \le i \le n$ considering them as univariate in x_i so that we eliminate step by step the variables $x_2, \ldots, x_n$ in g. The details of this algorithm, which we name `Eliminate`, are given in Appendix D (Algorithm D.1). The inputs of `Eliminate` are $g, a_0, a_2, \ldots, a_n$ and the output is the list $[\phi_2, \ldots, \phi_n]$ and the remainder r.

The bit complexity of `Eliminate` is given in the following lemma whose proof (which is quite routine) is given in Appendix D.

LEMMA 2.6. *Assume that* $g \in \mathbb{Q}[x_1][x_2, \ldots, x_n]$ *has degree* d *in* $x_2, \ldots, x_n$ *and bitsize* τ_g, *and that the polynomials* $a_0, a_2, \ldots, a_n \in \mathbb{Q}[x_1]$ *have bitsizes at most* τ_a. *Then, Algorithm* `Eliminate` *runs in*

$$\widetilde{O}\left(n\tau_g + n^2 d\tau_a\right)$$

boolean operations and the bitsizes of the outputs $\phi_2, \ldots, \phi_n$ *are in* $\widetilde{O}\left(\tau_g + nd\tau_a\right)$.

3. SOS of polynomials modulo gradient ideals.

3.1. The existence of an SOS decomposition over the rationals. The main result of this section is stated below.

THEOREM 3.1. *Let* $f \in \mathbb{Q}[\boldsymbol{x}]$ *such that the following conditions hold:*

a) *The infimum* $f^\star = \inf\{f(x) : x \in \mathbb{R}^n\}$ *is attained.*
b) *The gradient ideal* $\mathcal{I}_{grad}(f)$ *is zero-dimensional and radical.*

Then, f *is non-negative over* $\mathbb{R}^n$ *if and only if* f *is an SOS of polynomials over the quotient ring* $\mathbb{Q}[\boldsymbol{x}]/\mathcal{I}_{grad}(f)$.

Proof. Suppose that f is non-negative over $\mathbb{R}^n$ and $\#V_{grad}(f) = \delta$. We prove that f is an SOS of polynomials over the quotient ring $\mathbb{Q}[\boldsymbol{x}]/\mathcal{I}_{grad}(f)$. We consider the two following cases:

CASE 1. Distinct points in $V_{grad}(f)$ have distinct x_1-coordinates. Consider the lexicographic monomial order $x_1 < x_2 < \cdots < x_n$ on $\mathbb{Q}[\boldsymbol{x}]$. Since the gradient ideal is zero-dimensional and radical, according to the Shape Lemma (Lemma 2.2), the reduced Gröbner basis of $\mathcal{I}_{grad}(f)$ has the following form:

$$[w, x_2 - v_2, \ldots, x_n - v_n], \tag{3.1}$$

where $v_2, \ldots, v_n$ are polynomials in $\mathbb{Q}[x_1]$ of degree at most $\delta - 1$. We denote

$$h(x_1) := f(x_1, v_2, \ldots, v_n), \tag{3.2}$$

where x_i is replaced by v_i in f for $i = 2, \ldots, n$. With the order $<$, we divide $f - h$ by the system in (3.1) by using the division algorithm in [11, Ch. 2, Sec 3.]. Then, there exist $\phi_1, \dots, \phi_n$ in $\mathbb{Q}[\boldsymbol{x}]$, and r in $\mathbb{Q}[x_1]$ such that

$$f - h = \phi_1 w + \sum_{i=2}^{n} \phi_i (x_i - v_i) + r, \tag{3.3}$$

with $\deg r < \delta$. Let x be in $V_{grad}(f)$. From (3.2) and (3.3), one sees that $f(x) = h(x)$. Hence, $f - h$ vanishes on $V_{grad}(f)$. Clearly, the value of $\phi_1 w + \sum_{i=2}^{n} \phi_i (x_i - v_i)$ is zero on $V_{grad}(f)$. This implies that r also vanishes on the image set $\pi(V_{grad}(f))$, where $\pi(x_1, \dots, x_n) = x_1$. Since distinct points in $V_{grad}(f)$ have distinct x_1-coordinates, it holds that $\#\pi(V_{grad}(f)) = \#V_{grad}(f) = \delta$. As $\deg r < \delta$, we conclude that $r \equiv 0$. Hence, from (3.3), we obtain the following representation:

$$f = h + \phi_1 w + \sum_{i=2}^{n} \phi_i (x_i - v_i). \tag{3.4}$$

The set $\{(x_1, x_2, \dots, x_n) \in \mathbb{R}^n : x_2 = v_2, \dots, x_n = v_n\}$ defines a curve which is parametrized by x_1. Recall that f is non-negative over $\mathbb{R}^n$. Hence f is non-negative over this curve. Since f takes the same values over this curve as h takes over x_1 when x_1 ranges in $\mathbb{R}$, one can conclude that the univariate polynomial h is also non-negative over $\mathbb{R}$. According to the results on SOS decompositions of univariate polynomials with rational coefficients in Theorem 2.1, h is a sum of s squares in $\mathbb{Q}[x_1]$, i.e., there exist $q_1, \dots, q_s \in \mathbb{Q}[x_1]$ and $c_1, \dots, c_s$ in $\mathbb{Q}_+$ such that $h = c_1 q_1^2 + \cdots + c_s q_s^2$. Therefore, from (3.4), we assert that f is an SOS of polynomials over $\mathbb{Q}[\boldsymbol{x}]/\mathcal{I}_{grad}(f)$.

Case 2. There are two distinct points in $V_{grad}(f)$ such that their x_1's-coordinates are equal. According to [37, Lemma 2.1], there is $j \in \{1, \dots, (n-1)\delta(\delta-1)/2\}$ such that the linear function $u := x_1 + jx_2 + \cdots + j^{n-1} x_n$ separates $V_{grad}(f)$, i.e., $u(x) \neq u(y)$ for any distinct points x, y in $V_{grad}(f)$. We consider the change of variables $\boldsymbol{y} = T\boldsymbol{x}$, where

$$T = \begin{bmatrix} 1 & j & j^2 & \cdots & j^{n-1} \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}. \tag{3.5}$$

We see that T is an invertible matrix. Then we obtain a polynomial $g(\boldsymbol{y}) = f(T^{-1}\boldsymbol{y})$ in variables $y_1, y_2, \dots, y_n$ having the following property: the infimum $g^\star = \inf\{g(y) : y \in \mathbb{R}^n\}$ is attained. Because of the chain rule $\nabla g = \nabla f \circ T^{-1}$, we have

$$V_{grad}(g) = \{y \in \mathbb{C}^n : y = Tx, x \in V_{grad}(f)\};$$

Thus, the gradient ideal $\mathcal{I}_{grad}(g)$ is zero-dimensional and radical. Moreover, since $y_1 = u(\boldsymbol{x})$ separates $V_{grad}(f)$, distinct points in $V_{grad}(g)$ have distinct y_1-coordinates. We observe that $g \in \mathbb{Q}[\boldsymbol{y}]$ is non-negative and satisfies the conditions of the theorem; Case 1 happens to $V_{grad}(g)$ as well. Hence, there exists an SOS decomposition of g modulo $\mathcal{I}_{grad}(g)$

$$g(\boldsymbol{y}) = \sum_{j=1}^{s} c_j \bar{q}_j^2(\boldsymbol{y}) + \sum_{i=1}^{n} \bar{\phi}_i(\boldsymbol{y}) \frac{\partial g}{\partial y_i}, \tag{3.6}$$

where $\bar{q}_1, \ldots, \bar{q}_s, \bar{\phi}_1, \ldots, \bar{\phi}_n \in \mathbb{Q}[\boldsymbol{y}]$ and $c_1, \ldots, c_s \in \mathbb{Q}_+$. In (3.6), we replace $\boldsymbol{y}$ by $T\boldsymbol{x}$ and $\frac{\partial g}{\partial y_i}$ by $\frac{\partial f}{\partial x_i} \circ T^{-1}$, we obtain a decomposition of f as follows:

$$f(\boldsymbol{x}) = g(T\boldsymbol{x}) = \sum_{j=1}^{s} c_j \bar{q}_j^2(T\boldsymbol{x}) + \sum_{i=1}^{n} \bar{\phi}_i(T\boldsymbol{x}) \frac{\partial f}{\partial x_i} \circ T^{-1}. \tag{3.7}$$

Because of $(\frac{\partial f}{\partial x_i} \circ T^{-1})(Tx) = \frac{\partial f}{\partial x_i}(x)$, (3.7) is an SOS decomposition of f modulo $\mathcal{I}_{grad}(f)$ ~~of f~~.

We now prove the reverse conclusion. Suppose that f is SOS over the quotient ring $\mathbb{Q}[\boldsymbol{x}]/\mathcal{I}_{grad}(f)$, i.e., f can be decomposed as follows:

$$f = \sum_{j=1}^{s} c_j q_j^2 + \sum_{i=1}^{n} \phi_i \frac{\partial f}{\partial x_i}, \tag{3.8}$$

for some polynomials $q_1, \ldots, q_s, \phi_1, \ldots, \phi_n \in \mathbb{Q}[\boldsymbol{x}]$, and $c_1, \ldots, c_s$ in $\mathbb{Q}_+$. Let $x^\star \in \mathbb{R}^n$ be such that $f(x^\star) = f^\star$. Then $x^\star$ is a critical point of f over $\mathbb{R}^n$, i.e., $x^\star$ belongs to the variety $V_{grad}(f)$; thus, we have

$$\sum_{i=1}^{n} \phi_i(x^\star) \frac{\partial f}{\partial x_i}(x^\star) = 0.$$

From (3.8), we see that $f(x^\star) = \sum_{j=1}^{s} c_j q_j^2(x^\star)$ and so this value is non-negative. By assumption, for all x in $\mathbb{R}^n$, $f(x) \geq f(x^\star)$. Hence, f is non-negative over $\mathbb{R}^n$. ∎

Remark 3.2. Assume that Q is a real field and R is the real closure of Q. All arguments in the proof of Theorem 3.1 can be applied for f in $\mathrm{Q}[\boldsymbol{x}]$. Hence, the conclusion of Theorem 3.1 holds for the case $\mathrm{Q}[\boldsymbol{x}]$, i.e., f is non-negative over R^n if and only if f is an SOS of polynomials over the quotient ring $\mathrm{Q}[\boldsymbol{x}]/\mathcal{I}_{grad}(f)$ provided that the two following conditions hold: the infimum $f^\star = \inf\{f(x) : x \in \mathrm{R}^n\}$ is attained; the gradient ideal $\mathcal{I}_{grad}(f)$ is zero-dimensional and radical.

Remark 3.3. In the proof of Theorem 3.1, one can see that $f - h$ vanishes not only on $V_{grad}(f)$ but also on the variety defined by $\langle x_2 - v_2, \ldots, x_n - v_n \rangle$. Hence, ϕ_1 in (3.4) is zero and (3.4) becomes $f = c_1 q_1^2 + \cdots + c_s q_s^2 + \sum_{i=2}^{n} \phi_i (x_i - v_i)$.

Remark 3.4. Note that if f does not attain its infimum, it could be SOS modulo the gradient ideal but fail to be nonnegative, as it may be negative at points where the gradient does not vanish. This is illustrated by the example $f = x^2 + (xy-1)^2 - \frac{1}{2}$ whose gradient ideal is generated by x, y. Hence, f is $1/2$ modulo its gradient ideal while it can have negative values (e.g. along the sequence of points $\left(\frac{1}{2^k}, 2^k\right)$ for $k \geq 1$). So the condition a) in Theorem 3.1 is used only to prove the reverse conclusion. Therefore, even without this condition, the following assertion still holds: if $\mathcal{I}_{grad}(f)$ is zero-dimensional and radical and f is non-negative over $\mathbb{R}^n$, then f is SOS modulo $\mathcal{I}_{grad}(f)$.

Theorem 3.1 provides certificates of non-negativity for polynomials in $\mathbb{Q}[\boldsymbol{x}]$ which satisfy its assumptions and which are not SOS of polynomials with real (or rational) coefficients. We illustrate this with two examples.

Example 3.5. We recall a polynomial of Robinson [36] that is non-negative but cannot be represented as an SOS of polynomials,

$$\bar{f}_R = x_1^6 + x_2^6 + x_3^6 - x_1^4 x_2^2 - x_1^4 x_3^2 - x_2^4 x_1^2 - x_2^4 x_3^2 - x_3^4 x_1^2 - x_3^4 x_2^2 + 3x_1^2 x_2^2 x_3^2.$$

By substituting the third variable x_3 by 1 in $\bar{f}_R$, we get the following non-negative polynomial:

$$f_R = x_1^6 + x_2^6 - x_1^4 x_2^2 + 3x_1^2 x_2^2 - x_1^2 x_2^4 - x_1^4 - x_2^4 - x_1^2 - x_2^2 + 1.$$

Because $\bar{f}_R$ is the homogenization of f_R, f_R cannot be represented as an SOS of polynomials [29, Proposition 1.2.4]. The gradient ideal $\mathcal{I}_{grad}(f_R)$ is zero-dimensional and radical. Therefore, Theorem 3.1 tells us that f_R is an SOS of polynomials over the quotient ring $\mathbb{Q}[\boldsymbol{x}]/\mathcal{I}_{grad}(f_R)$.

Example 3.6. In [41], Scheiderer introduced the following homogeneous polynomial:

$$\bar{f}_S = x_1^4 + x_1 x_2^3 + x_2^4 - 3x_1^2 x_2 x_3 - 4x_1 x_2^2 x_3 + 2x_1^2 x_3^2 + x_1 x_3^3 + x_2 x_3^3 + x_3^4,$$

that can be decomposed as an SOS of polynomials with algebraic coefficients but cannot be decomposed as an SOS of polynomials with rational coefficients. By replacing the third variable x_3 by -1, we obtain the non-negative polynomial

$$f_S = x_1^4 + x_1 x_2^3 + x_2^4 + 3x_1^2 x_2 + 4x_1 x_2^2 + 2x_1^2 - x_1 - x_2 + 1.$$

Note that the conclusion in [29, Proposition 1.2.4] holds for polynomials with rational coefficients, i.e., $g \in \mathbb{Q}[\boldsymbol{x}]$ is SOS in $\mathbb{Q}[\boldsymbol{x}]$ if and only if its homogenization is in $\mathbb{Q}[\boldsymbol{x}]$. Hence, the polynomial f_S is also SOS with algebraic coefficients but not SOS with rational ones. The gradient ideal $\mathcal{I}_{grad}(f_S)$ satisfies the zero-dimensional and radical condition. Hence, according to Theorem 3.1, f_S is an SOS of polynomials over the quotient ring $\mathbb{Q}[\boldsymbol{x}]/\mathcal{I}_{grad}(f_S)$.

An explicit SOS decomposition of f_S will be given in the next section.

3.2. Description of the algorithm. Based on the proof of Theorem 3.1, we design an algorithm to compute an SOS decomposition of polynomials modulo the gradient ideal of a non-negative polynomial with rational coefficients.

The input of `sosgradientshape` is a non-negative polynomial $f \in \mathbb{Q}[\boldsymbol{x}]$ whose gradient ideal $\mathcal{I}_{grad}(f)$ is zero-dimensional and radical and satisfies the Shape Lemma assumption, i.e., all points in $V_{grad}(f)$ have distinct x_1-coordinates. Our software implementation first checks that the gradient ideal is zero-dimensional and radical, and returns an error if the assumption is not satisfied. To do so, we rely on the procedures `IsZeroDimensional` and `IsRadical` from the Maple package `PolynomialIdeals`. These are all based on Gröbner bases computations (see e.g. [11]).

The output includes the cardinality $\delta = \#V_{grad}(f)$, the lists of polynomials and numbers

$$[w, v_2, \dots, v_n],\ [q_1, \dots, q_s],\ [\phi_2, \dots, \phi_n] \subset \mathbb{Q}[\boldsymbol{x}],\ \text{and } [c_1, \dots, c_s] \subset \mathbb{Q}_+$$

satisfying the relation

$$f = \sum_{j=1}^{s} c_j q_j^2 + \sum_{i=2}^{n} \phi_i (x_i - v_i)$$

In Step 1, we compute the reduced Gröbner basis G for $\mathcal{I}_{grad}(f)$ by relying on a zero-dimensional rational parametrization of $V_{grad}(f)$ mentioned in Lemma 2.4. In Step 2, we compute the quotients $\phi_2, \dots, \phi_n$ and the remainder r of the division of f by G. In Step 3, we compute a rational weighted SOS decomposition of the non-negative univariate polynomial h by using Algorithm `univsos1` or Algorithm `univsos2` described in [27, Fig. 1] or [27, Fig. 2], respectively.

ALGORITHM 3.1 Computing SOS of polynomials modulo the gradient ideal

`sosgradientshape` := proc(f)

Input: $f \in \mathbb{Q}[\boldsymbol{x}]$ non-negative over $\mathbb{R}^n$ such that $\mathcal{I}_{grad}(f)$ is zero-dimensional and radical and all points in $V_{grad}(f)$ have distinct x_1-coordinates

Output: δ in $\mathbb{N}$, $[q_1, \ldots, q_s]$, $[w, v_2, \ldots, v_n] \subset \mathbb{Q}[x_1]$, $[\phi_2, \ldots, \phi_n] \subset \mathbb{Q}[\boldsymbol{x}]$, and $[c_1, \ldots, c_s] \subset \mathbb{Q}_+$ satisfying

$$f = \sum_{j=1}^{s} c_j q_j^2 + \sum_{i=2}^{n} \phi_i (x_i - v_i). \tag{3.9}$$

1: Compute the reduced Gröbner basis $G = [w, x_2 - v_2, \ldots, x_n - v_n]$ of $\mathcal{I}_{grad}(f)$, with the lexicographical ordering $x_1 < x_2 < \cdots < x_n$, and $\delta = \deg w$
2: Compute the quotients $[\phi_2, \ldots, \phi_n]$ and remainder h of the division of f by G by performing `Eliminate`$(f, 1, v_2, \ldots, v_n)$
3: Compute a rational weighted SOS decomposition $h = c_1 q_1^2 + \cdots + c_s q_s^2$
4: Return δ, $[q_1, \ldots, q_s]$, $[\phi_2, \ldots, \phi_n]$, $[w, v_2, \ldots, v_n]$, and $[c_1, \ldots, c_s]$

REMARK 3.7. Suppose that the Shape Lemma assumption does not hold for $\mathcal{I}_{grad}(f)$, i.e., there are two distinct points in $V_{grad}(f)$ such that their x_1's-coordinates are equal. As mentioned in the proof of Theorem 3.1, we can find an invertible matrix T given by (3.5), change of variables $\boldsymbol{y} = T\boldsymbol{x}$, and assign $g(\boldsymbol{y}) := f(T^{-1}\boldsymbol{y})$. Here, we have $y_1 = x_1 + jx_2 + \cdots + j^{n-1}x_n$ for some $j > 0$ and $y_i = x_i$ for $i = 2, ..., n$. We get a new non-negative polynomial in n new variables with rational coefficients $g(\boldsymbol{y})$ whose gradient ideal satisfies the Shape Lemma assumption. Now we can apply Algorithm `sosgradientshape` for $g(\boldsymbol{y})$ and obtain the output: the number $\bar{\delta}$, two lists $[\bar{q}_1, \ldots, \bar{q}_s]$, $[\bar{w}, \bar{v}_2, \ldots, \bar{v}_n]$ of polynomials in $\mathbb{Q}[y_1]$, a list $[\bar{\phi}_1, \ldots, \bar{\phi}_n]$ of polynomials in $\mathbb{Q}[\boldsymbol{y}]$, and a list $[c_1, \ldots, c_s] \subset \mathbb{Q}_+$. Since $\#V_{grad}(f) = \#V_{grad}(g)$, one has $\bar{\delta} = \delta$. The new polynomial g can be decomposed as follows:

$$g(\boldsymbol{y}) = \sum_{j=1}^{s} c_j \bar{q}_j^2(y_1) + \bar{\phi}_1(\boldsymbol{y})\bar{w}(y_1) + \sum_{i=2}^{n} \bar{\phi}_i(\boldsymbol{y})(y_i - \bar{v}_i(y_1)).$$

Hence, f can be decomposed as:

$$f(\boldsymbol{x}) = \sum_{j=1}^{s} c_j \bar{q}_j^2(u(\boldsymbol{x})) + \bar{\phi}_1(T\boldsymbol{x})\bar{w}(u(\boldsymbol{x})) + \sum_{i=2}^{n} \bar{\phi}_i(T\boldsymbol{x})(x_i - \bar{v}_i(u(\boldsymbol{x}))), \tag{3.10}$$

where $u(\boldsymbol{x}) = x_1 + jx_2 + \cdots + j^{n-1}x_n$. Clearly, $[w(u), x_2 - \bar{v}_2(u), \ldots, x_n - \bar{v}_n(u)]$ is also a basis for $V_{grad}(f)$. Hence, (3.10) provides us an SOS decomposition of f modulo the gradient ideal of f.

THEOREM 3.8. *Let f be a non-negative polynomial in $\mathbb{Q}[\boldsymbol{x}]$. Suppose that f is non-negative over $\mathbb{R}^n$, $\mathcal{I}_{grad}(f)$ is zero-dimensional and radical, and all points in $V_{grad}(f)$ have distinct x_1-coordinates. On input f, Algorithm* `sosgradientshape` *terminates and computes an SOS decomposition of f modulo $\mathcal{I}_{grad}(f)$ with rational coefficients.*

Proof. Assume that $f \in \mathbb{Q}[\boldsymbol{x}]$ is non-negative over $\mathbb{R}^n$ and its gradient ideal is zero-dimensional and radical. Here, we use the lexicographic monomial order $x_1 < x_2 <$

$\cdots < x_n$. Because the Shape Lemma assumption holds, the reduced Gröbner basis of $\mathcal{I}_{grad}(f)$ in Step 1 has the form $G = [w, x_2 - v_2, \ldots, x_n - v_n]$, and can be computed by using a zero-dimensional rational parametrization of $V_{grad}(f)$ as in Lemma 2.4. In Step 2, we compute the quotients $[\phi_2, \ldots, \phi_n]$ and the remainder r of the division of f by G by performing $\texttt{Eliminate}(f, 1, v_2, \ldots, v_n)$ (as in Algorithm D.1). Here, we see that r coincides with h, where $h = f(x_1, v_2, \ldots, x_n)$ as in the proof of Theorem 3.1, because of

$$r = f - \sum_{i=2}^{n} \phi_i (x_i - v_i) = h.$$

In Step 3, the univariate polynomial h is non-negative with rational coefficients, so by using `univsos1` or `univsos2` [27], we can compute an SOS decomposition of $h = c_1 q_1^2 + \cdots + c_s q_s^2$. Hence, according to the proof of Theorem 3.1, we get (3.9) which is an SOS decomposition modulo the gradient ideal of f. ◻

To illustrate how the algorithm works, we consider the following simple example.

Example 3.9. Consider the polynomial $f(x_1, x_2) = 2x_1^4 + 2x_1x_2 + x_2^2 + 10$. This polynomial is non-negative over $\mathbb{R}^n$. Firstly, the gradient ideal $\mathcal{I}_{grad}(f)$ is given by $\mathcal{I}_{grad}(f) = \langle 8x_1^3 + 2x_2, 2x_1 + 2x_2 \rangle$ which is zero-dimensional and radical. We compute the reduced Gröbner basis of $\mathcal{I}_{grad}(f)$, namely $\langle x_1^3 - \frac{1}{4}x_1, x_2 + x_1 \rangle$, here $v_2(x_1) = -x_1$, with $\delta = \deg(x_1^3 - \frac{1}{4}x_1) = 3 = \#V_{grad}(f)$. Secondly, with the order $x_1 < x_2$, the quotients of the division of f by the Gröbner basis are $\phi_1 = 0$ and $\phi_2 = x_1 + x_2$, and the remainder is given by $h(x_1) = f(x_1, v_2) = 2x_1^4 - x_1^2 + 10$. Thirdly, by using Algorithm `univsos2` in [27], one gets an SOS decomposition of $h = \frac{1}{2}x_1^4 + \frac{3}{2}(x_1^2 - \frac{5}{2})^2 + \frac{13}{2}x_1^2 + \frac{5}{8}$. Finally, we obtain the following SOS decomposition of f modulo its gradient ideal:

$$f = \frac{1}{2}x_1^4 + \frac{3}{2}\Big(x_1^2 - \frac{5}{2}\Big)^2 + \frac{13}{2}x_1^2 + \frac{5}{8} + (x_1 + x_2) \times (x_2 + x_1).$$

3.3. Bit complexity analysis. This subsection investigates the bit complexity of `sosgradientshape`. Assume that d and τ are respectively the degree and an upper bound of the bitsize of the coefficients of $f \in \mathbb{Q}[\boldsymbol{x}]$. We provide estimates for the bitsizes of polynomials in the output of $\texttt{sosgradientshape}(f)$ as well as for the number of boolean operations required to execute it.

We use Algorithm `univsos1` in [27, Fig. 1] or Algorithm `univsos2` in [27, Fig. 2] to compute an SOS decomposition of the non-negative univariate polynomial h. The corresponding bit complexities are given as follows:

Proposition 3.10. *Let $v_2, \ldots, v_n$ be as in Lemma 2.4 and $h(x_1) = f(x_1, v_2, \ldots, v_n)$. To compute an SOS decomposition of h, Algorithm `univsos1` and Algorithm `univsos2` run in*

$$\widetilde{O}\Big((d^{n+1}/2)^{3d^{n+1}/2}(\tau + n + d)d^{3n+1}\Big) \tag{3.11}$$

and

$$\widetilde{O}\left((\tau + n + d)d^{6n+4}\right) \tag{3.12}$$

boolean operations, respectively.

Proof. Let $\tau_v = \max_i\{ht(v_i)\}$. Lemma 2.4 tells us that the bitsize of τ_v is bounded from above by $\widetilde{O}\left((\tau + n + d)d^{3n}\right)$, and that the polynomials $w, v_2, \ldots, v_n$ have degree at most $(d-1)^n$. Since $\deg f = d$ and $h(x_1) = f(x_1, v_2, \ldots, v_n)$, the degree of h is at most $d(d-1)^n$.

Let β be the minimum common denominator of all non-zero coefficients of h. Computing an SOS decomposition of h boils down to computing an SOS decomposition of βh. In particular, the execution time of `univsos1` (resp., `univsos2`) on h is the same as for βh. Now we estimate the bitsize of the polynomial $\beta h \in \mathbb{Z}[x_1]$. By the definition of h, we observe that $ht(h) \le \tau + d\tau_v$. It follows that $ht(\beta h) \le ht(\beta) + \tau + d\tau_v$. By definition we have $ht(\beta) \le \tau + d\tau_v$. This yields

$$ht(\beta h) \le 2\left(\tau + d\tau_v\right). \tag{3.13}$$

From (3.13) and above results, we obtain the following bitsize estimate for βh:

$$\widetilde{O}\left(2(\tau + d(\tau + n + d)d^{3n})\right) = \widetilde{O}\left((\tau + n + d)d^{3n+1}\right).$$

To compute an SOS decomposition of βh, we rely on `univsos1` or `univsos2`. From [27, Theorem 17], the boolean running time of `univsos1` corresponds to the quantity given by (3.11). If we use `univsos2` then the number of boolean operations, by applying [27, Theorem 24], will be bounded from above by

$$\widetilde{O}\left(d^4(d-1)^{4n} + d^4(\tau + n + d)(d-1)^{6n}\right),$$

which can be further reduced to (3.12). ∎

Proposition 3.11. *Let $v_2, \ldots, v_n$ be as in Proposition 3.10. To compute the list $\phi_2, \ldots, \phi_n$ in the output of Algorithm* `sosgradientshape`*, Algorithm* `Eliminate` *runs in $\widetilde{O}(n^2(\tau + n + d)d^{3n+1})$ boolean operations and the bitsizes of $\phi_2, \ldots, \phi_n$ are $\widetilde{O}\left(n(\tau + n + d)d^{3n+1}\right)$.*

Proof. From Lemma 2.4, the polynomial v_i has bitsize at most $\widetilde{O}((\tau+n+d)d^{3n})$. We divide f by $[x_2 - v_2, \ldots, x_n - v_n]$ while performing `Eliminate`$(f, 1, v_2, \ldots, v_n)$ as in Algorithm D.1 to obtain the list of quotients $[\phi_2, \ldots, \phi_n]$ and the remainder $h = h(x_1, v_2, \ldots, v_n)$. Applying Lemma 2.6 for this division, we conclude that Algorithm `Eliminate` runs in $\widetilde{O}(n^2(\tau + n + d)d^{3n+1})$ boolean operations, the estimate for the bitsize of ϕ_i is $\widetilde{O}\left(n(\tau + n + d)d^{3n+1}\right)$ as claimed. ∎

We are now ready to analyze the bit complexity of Algorithm 3.1.

Theorem 3.12. *Let $f \in \mathbb{Q}[\boldsymbol{x}]$ of degree d and let τ be the maximum bitsize of its coefficients. Assume that the two conditions in Theorem 3.1 hold. Then, on input f, Algorithm* `sosgradientshape` *runs in*

$$\widetilde{O}\Big((\tau + n + d)^2 d^{6n} + (\tau + n + d)d^{3n+1}(d^{n+1}/2)^{3d^{n+1}/2}\Big) \tag{3.14}$$

or

$$\widetilde{O}\Big((\tau + n + d)^2 d^{6n} + (\tau + n + d)d^{6n+4}\Big) \tag{3.15}$$

boolean operations if in Step 3 we use Algorithm `univsos1` *or Algorithm* `univsos2`*, respectively.*

Proof. Assume that in Step 3 we use `univsos1` to compute an SOS decomposition of h. Then, the number of boolean operations that `sosgradientshape` uses to compute the SOS decomposition of f is the sum of the four following ones:

1. the number of boolean operations required to compute the zero-dimensional rational parametrization $\mathcal{Q}$ of $V_{grad}(f)$ as in (2.2);

2. the number of boolean operations required to compute $w, v_2, \dots, v_n \in \mathbb{Q}[x_1]$, defined in Lemma 2.4 as in (2.3);
3. the number of boolean operations required to compute an SOS decomposition of h by using Algorithm `univsos1` as in (3.11);
4. the number of boolean operations required to compute $\phi_2, \dots, \phi_n$ in the output of `sosgradientshape` by using Algorithm `Eliminate` (mentioned in Proposition 3.11).

This sum equals

$$\widetilde{O}\Big(n^2(d+\tau)d^{2n+1}\binom{n+d}{d} + (\tau+n+d)^2 d^{6n} + (\tau+n+d)d^{3n+1}\Big(\frac{d^{n+1}}{2}\Big)^{3d^{n+1}/2} + (\tau+n+d)n^2 d^{3n+2}\Big).$$

In this sum, the third term is larger than the first and last term for large enough d and n, yielding the estimate (3.14).

If in Step 3 we use `univsos2`, the number of boolean operations of the algorithm is

$$\widetilde{O}\Big(n^2(d+\tau)d^{2n+1}\binom{n+d}{d} + (\tau+n+d)^2 d^{6n} + (\tau+n+d)d^{6n+4} + n^2(\tau+n+d)d^{3n+2}\Big).$$

Noting that $\binom{n+d}{d} \le (d+1)^n \le d^{2n}$ for large enough d and n, we obtain (3.15). ∎

THEOREM 3.13. *Assume that $f \in \mathbb{Q}[\boldsymbol{x}]$ satisfies the conditions of Theorem 3.12. Let $w, v_2, \dots, v_n$, h be as in Proposition 3.10. Then, the maximum bitsize of the coefficients involved in the SOS decomposition of h obtained by using Algorithm* `univsos1` *and Algorithm* `univsos2` *are bounded from above, respectively, by*

$$\widetilde{O}\Big((\tau+n+d)(d^{n+1}/2)^{3d^{n+1}/2} d^{3n+1}\Big), \tag{3.16}$$

and

$$\widetilde{O}\left((\tau+n+d)d^{5n+3}\right). \tag{3.17}$$

Proof. From the proof of Proposition 3.10, the estimates for degree and bitsize of βh are $d(d-1)^n$ and $\widetilde{O}\left((\tau+n+d)d^{3n+1}\right)$, respectively. According to [27, Theorem 16] and [27, Theorem 23], the maximum bitsize of the coefficients involved in the SOS decomposition of βh obtained by using `univsos1` and `univsos2` are bounded from above by (3.16) and (3.17), respectively. ∎

4. SOS of rational fractions modulo gradient ideals. Artin's Theorem [1] states that if $f \in \mathbb{R}[\boldsymbol{x}]$ is non-negative then there exists a nonzero $g \in \mathbb{R}[\boldsymbol{x}]$ such that $g^2 f$ is SOS, yielding a decomposition of f as an SOS of rational fractions. In this section, we explain how to decompose $f \in \mathbb{Q}[\boldsymbol{x}]$ as an SOS of rational fractions modulo its gradient ideal. One says that $f \in \mathbb{Q}[\boldsymbol{x}]$ is an *SOS of rational fractions* in $\mathbb{Q}(\boldsymbol{x})$, where $\mathbb{Q}(\boldsymbol{x})$ is the field of rational fractions in the variable $\boldsymbol{x}$ over $\mathbb{Q}$, if there exist rational fractions $f_1, \dots, f_s$ in $\mathbb{Q}(\boldsymbol{x})$ and $[c_1, \dots, c_s] \subset \mathbb{Q}_+$ such that $f = \sum_{j=1}^s c_j f_j^2$. Furthermore, f is an SOS of rational fractions over the quotient ring $\mathbb{Q}(\boldsymbol{x})/\mathcal{I}_{grad}(f)$ if there exists $g \in \mathcal{I}_{grad}(f)$ such that $f - g$ is an SOS of rational fractions in $\mathbb{Q}(\boldsymbol{x})$, i.e., f can be decomposed as follows:

$$f = \sum_{j=1}^s c_j f_j^2 + \sum_{i=1}^n \phi_i \frac{\partial f}{\partial x_i},$$

for some rational fractions $f_1, \dots, f_s, \phi_1, \dots, \phi_s$ in $\mathbb{Q}(\boldsymbol{x})$ and $[c_1, \dots, c_s] \subset \mathbb{Q}_+$.

4.1. The existence of an SOS decomposition over the rationals. Denote by $\mathbb{Q}(x_1)[x_2, \ldots, x_n]$ the vector space of polynomials in $n-1$ variables $(x_2, \ldots, x_n)$ with coefficients in $\mathbb{Q}(x_1)$.

In the following theorem, we prove the existence of an SOS decomposition of rational fractions modulo the gradient ideal for f.

THEOREM 4.1. *Assume that $f \in \mathbb{Q}[\boldsymbol{x}]$ is a non-negative polynomial of degree d and that $\mathcal{I}_{grad}(f)$ is zero-dimensional and radical. Let $\mathcal{Q} = ((w, \kappa_1, \ldots, \kappa_n), x_1)$ be a zero-dimensional rational parametrization of $V_{grad}(f)$. Then, f can be decomposed as an SOS of rational fractions modulo the gradient ideal, in particular*

$$f = \frac{1}{(w')^d} \sum_{j=1}^{s} c_j q_j^2 + \sum_{i=1}^{n} \frac{\phi_i}{(w')^d} (w' x_i - \kappa_i), \tag{4.1}$$

for some $q_1, \ldots, q_s \in \mathbb{Q}[x_1]$, $\phi_1, \ldots, \phi_n \in \mathbb{Q}[\boldsymbol{x}]$, and $[c_1, \ldots, c_s] \subset \mathbb{Q}_+$.

Proof. By substituting $x_i = \frac{\kappa_i}{w'}$ in f, for $i = 2, \ldots, n$, one has

$$f\Big(x_1, \frac{\kappa_2}{w'}, \ldots, \frac{\kappa_n}{w'}\Big) = \frac{1}{(w')^d} \bar{h}, \tag{4.2}$$

where $\bar{h}(x_1)$ is a univariate polynomial. Since f is non-negative with even degree d, $\bar{h}$ is also non-negative. In addition, the coefficients of $w', \kappa_1, \ldots, \kappa_n$ and f are rational numbers, so the coefficients of $\bar{h}$ are also rational numbers. Applying Theorem 2.1 for $\bar{h}$, we conclude that there are $q_1, \ldots, q_s \in \mathbb{Q}[x_1]$ and $[c_1, \ldots, c_s] \subset \mathbb{Q}_+$ such that

$$\bar{h} = \sum_{j=1}^{s} c_j q_j^2. \tag{4.3}$$

Next, one considers the division of $(w')^d f - \bar{h}$ by $[w' x_1 - \kappa_1, \ldots, w' x_n - \kappa_n]$ with the lexicographic order $x_1 < \cdots < x_n$. Based on Buchberger's Criterion [9], we can show that this system is a Gröbner basis of the ideal generated by this system w.r.t. the order $<$ in $\mathbb{Q}[\boldsymbol{x}]$. Hence, there exist a (unique) list of quotients $[\phi_1, \ldots, \phi_n]$ in $\mathbb{Q}[\boldsymbol{x}]$, and r in $\mathbb{Q}[x_1]$ such that

$$(w')^d f - \bar{h} = \sum_{i=1}^{n} \phi_i (w' x_i - \kappa_i) + r, \tag{4.4}$$

with r of smaller degree than the cardinality δ of $V_{grad}(f)$. The gradient variety of f can be represented as follows:

$$V_{grad}(f) = \{\boldsymbol{x} \in \mathbb{C}^n : w = 0, w' x_1 - \kappa_1 = \cdots = w' x_n - \kappa_n = 0\}.$$

From (4.2), one sees that $(w')^d f - \bar{h}$ vanishes on $V_{grad}(f)$. With the same arguments as in the proof of Theorem 3.1, we conclude that $r \equiv 0$. Hence, from (4.2), (4.3), and (4.4), we obtain a representation of f as in (4.1). ∎

In Theorem 4.5, we assume that $\mathcal{Q} = ((w, \kappa_1, \ldots, \kappa_n), x_1)$ is a zero-dimensional rational parametrization of $V_{grad}(f)$ which is a generic assumption. In this assumption, the linear form λ is given by $\lambda(\boldsymbol{x}) = x_1$. If the assumption does not hold, we can change the coordinate system such that the obtained polynomial (with new variables) satisfies the assumption as in Case 2 of the proof of Theorem 3.1.

REMARK 4.2. From (4.2), we see that $\deg \bar{h}$ does not exceed $\deg_{x_1} f + d \deg(w')$, where $\deg_{x_1} f$ is the degree of f in the variable x_1 and $\deg w' = \deg w - 1$. Thus, the degree of the univariate polynomial $\bar{h}$ is at most $d(d-1)^n$.

4.2. Algorithm to compute an SOS of rational fractions. From the proof of Theorem 4.1, we design an algorithm named `sosgradient` to compute the SOS decomposition of rational fractions for f. Algorithm `sosgradient` is obtained by a modification of Step 1 in `sosgradientshape` to get a zero-dimensional rational parametrization of the gradient variety of f.

ALGORITHM 4.1 Computing SOS of rational fractions modulo the gradient ideal

---

`sosgradient` := proc(f)
Input: $f \in \mathbb{Q}[\boldsymbol{x}]$ of degree d such that f is non-negative over $\mathbb{R}^n$ and $\mathcal{I}_{grad}(f)$ is zero-dimensional and radical
Output: $[w, \kappa_1, \ldots, \kappa_n]$, $[q_1, \ldots, q_s] \subset \mathbb{Q}[x_1]$, $[\phi_2, \ldots, \phi_n] \subset \mathbb{Q}(x_1)[x_2, \ldots, x_n]$, and $[c_1, \ldots, c_s] \subset \mathbb{Q}_+$ satisfying

$$f = \frac{1}{(w')^d} \sum_{j=1}^{s} c_j q_j^2 + \sum_{i=1}^{n} \frac{\phi_i}{(w')^d} \Big(x_i - \frac{\kappa_i}{w'} \Big). \tag{4.5}$$

1: Compute a zero-dimensional rational parametrization $[w, \kappa_1, \ldots, \kappa_n]$ of $V_{grad}(f)$
2: Compute the quotients $[\phi_2, \ldots, \phi_n]$ and the remainder $\bar{h}$ of the division of $(w')^d f$ by $[x_2 - \frac{\kappa_2}{w'}, \ldots, x_n - \frac{\kappa_n}{w'}]$ by performing `Eliminate`($(w')^d f, w', \kappa_2, \ldots, \kappa_n$)
3: Compute a rational weighted SOS decomposition of $\bar{h} = c_1 q_1^2 + \cdots + c_s q_s^2$
4: Return $[w, \kappa_1, \ldots, \kappa_n]$, $[q_1, \ldots, q_s]$, $[\phi_2, \ldots, \phi_n]$, and $[c_1, \ldots, c_s]$

The input of `sosgradient` is a non-negative polynomial f in $\mathbb{Q}[\boldsymbol{x}]$ whose gradient ideal $\mathcal{I}_{grad}(f)$ is zero-dimensional and radical. The outputs are a zero-dimensional rational parametrization of $V_{grad}(f)$, a list of polynomials $[q_1, \ldots, q_s] \subset \mathbb{Q}[x_1]$, and a list of $[\phi_2, \ldots, \phi_n] \subset \mathbb{Q}(x_1)[x_2, \ldots, x_n]$ satisfying (4.5). Note that the ϕ_i's in (4.1) and (4.5) are different a multiplier $\frac{1}{w'}$. Computing ϕ_i's in (4.5) is more convenient by using `Eliminate`.

In Step 1, we compute a zero-dimensional rational parametrization $[w, \kappa_1, \ldots, \kappa_n]$ of $V_{grad}(f)$. In Step 2, we compute the quotients $[\phi_2, \ldots, \phi_n]$ of the division of $(w')^d f$ by $[x_2 - \frac{\kappa_2}{w'}, \ldots, x_n - \frac{\kappa_n}{w'}]$ while using Algorithm `Eliminate`. Note that the remainder of this division coincides with $\bar{h}$ given in (4.2). In Step 3, we compute a rational weighted SOS decomposition of the univariate polynomial $\bar{h}$ by relying on Algorithms `univsos1` or `univsos2`.

The correctness of `sosgradient` is proved in a similar way as for Algorithm `sosgradientshape` in Theorem 3.8.

THEOREM 4.3. *Suppose that $f \in \mathbb{Q}[\boldsymbol{x}]$ is non-negative over $\mathbb{R}^n$ and $\mathcal{I}_{grad}(f)$ is zero-dimensional and radical. On input f, Algorithm* `sosgradient` *terminates and the outputs provide us an SOS decomposition of f as in* (4.1).

4.3. Bit complexity analysis. We now estimate the bitsizes of polynomials in the output as well as the number of boolean operations required to perform Algorithm `sosgradient`.

PROPOSITION 4.4. *Assume that τ is the maximum bitsize of the coefficients of f in the input of* `sosgradient`. *To compute the list $[\phi_2, \ldots, \phi_n]$ in the output, Algorithm* `Eliminate` *runs in $\widetilde{O}\left(n^2(\tau + n + d)d^{n+1}\right)$ boolean operations. Furthermore, the bitsize of ϕ_i is $\widetilde{O}\left(n(\tau + n + d)d^{n+1}\right)$, $i = 2, \ldots, n$.*

Proof. We compute the division of $(w')^d f$ by $[x_2 - \frac{\kappa_2}{w'}, \ldots, x_n - \frac{\kappa_n}{w'}]$ by performing `Eliminate`$((w')^d f, w', \kappa_2, \ldots, \kappa_n)$. We obtain the list of quotients $[\phi_2, \ldots, \phi_n]$ and the remainder $\bar{h}$. The degree of $(w')^d f$ in $x_2, \ldots, x_n$ is d, and $ht((w')^d f) = \widetilde{O}\left((\tau + n + d)d^{n+1}\right)$. The conclusions are obtained by applying Lemma 2.6 with $ht(\kappa_i) = \widetilde{O}\left((\tau + n + d)(d-1)^n\right)$. ∎

THEOREM 4.5. *Let $f \in \mathbb{Q}[\boldsymbol{x}]$ of degree d and let τ be the maximum bitsize of its coefficients. Assume that f is non-negative over $\mathbb{R}^n$ and $\mathcal{I}_{grad}(f)$ is zero-dimensional and radical. Then, on input f, Algorithm* `sosgradient` *uses*

$$\widetilde{O}\Big((d^{n+1}/2)^{3d^{n+1}/2}(\tau + n + d)d^{n+1}\Big), \tag{4.6}$$

or

$$\widetilde{O}((\tau + n + d)d^{4n+4}) \tag{4.7}$$

boolean operations if in Step 3 we use Algorithm `univsos1` *or Algorithm* `univsos2`*, respectively.*

Proof. From Corollary 2.3, the polynomials $w, \kappa_1, \ldots, \kappa_n$ in the zero-dimensional parametrization of the gradient variety $V_{grad}(f)$ have degree at most $(d-1)^n$ and bitsize $\widetilde{O}\left((\tau + n + d)(d-1)^n\right)$. We can see that the degree of the remainder $\bar{h}$ (as defined in (4.2)) in Step 2 of `sosgradient` is at most $d(d-1)^n + d$ and its bitsize is $\widetilde{O}\left((\tau + n + d)d^{n+1}\right)$. To compute an SOS decomposition of $\bar{h}$, by applying [27, Theorem 17] and [27, Theorem 24], Algorithm `univsos1` and Algorithm `univsos2` use

$$\widetilde{O}\Big((d^{n+1}/2)^{3d^{n+1}/2}(\tau + n + d)d^{n+1}\Big) \tag{4.8}$$

and

$$\widetilde{O}\left((\tau + n + d)d^{4n+4}\right) \tag{4.9}$$

boolean operations, respectively.

The estimates (4.6) and (4.7) are obtained from Corollary 2.3, Proposition 4.4, and the estimates (4.8) and (4.9) with the same line of reasoning as in the proof of Theorem 3.12. ∎

THEOREM 4.6. *Assume that $f \in \mathbb{Q}[\boldsymbol{x}]$ satisfies the conditions of Theorem 4.5. Then, the maximum bitsizes of the coefficients involved in the SOS decomposition of $\bar{h}$, obtained by using Algorithm* `univsos1` *and Algorithm* `univsos1`*, are bounded from above respectively by*

$$\widetilde{O}\Big((d^{n+1}/2)^{3d^{n+1}/2}(\tau + n + d)d^{n+1}\Big) \tag{4.10}$$

and

$$\widetilde{O}\left((\tau + n + d)d^{3n+3}\right). \tag{4.11}$$

Proof. From the proof of Theorem 4.5, the degree of $\bar{h}$ is at most $d(d-1)^n$ and the bitsize of $\bar{h}$ is $\widetilde{O}\left((\tau + n + d)d^{n+1}\right)$. The conclusions follow from [27, Theorem 16] and [27, Theorem 23] and the second assertion in Proposition 4.4. ∎

REMARK 4.7. In general, `sosgradient` is faster than `sosgradientshape` to certify non-negativity of polynomials with rational coefficients. When relying on `univsos2`, by comparing the estimates in (3.15) and (4.7), we conclude that the number of boolean operations to run `sosgradientshape` is about d^{2n} times larger than the one of `sosgradient`. The underlying reason is that the maximum bitsizes of $w, v_2, \ldots, v_n$ are $(d-1)^{2n}$ times bigger than the ones of $\kappa_1, \ldots, \kappa_n$ that are obtained by a zero-dimensional rational parametrization of the gradient variety.

To finish the section, we present an explicit SOS decomposition for the polynomial f_S obtained from Scheiderer's polynomial given in Example 3.6. Here, we rely on `sosgradient` to get the SOS decomposition.

EXAMPLE 4.8. We first compute a zero-dimensional rational parametrization $\mathcal{Q}$ of the gradient variety $V_{grad}(f_S)$:

$$\begin{aligned} w &= 4x_1^9 + x_1^6 - 16x_1^5 - 4x_1^3 - 4x_1^2 - 1, \\ \kappa_1 &= 15x_1^7 - 32x_1^6 - 9x_1^4 - 36x_1^3 - 6x_1 - 4, \\ \kappa_2 &= -3x_1^6 + 64x_1^5 + 24x_1^3 + 28x_1^2 + 9. \end{aligned}$$

In f_S, by substituting $x_2 = \kappa_2/w'$ as in (4.2), we get the non-negative univariate polynomial $\bar{h} = 1679616x_1^{36} + 3359232x_1^{34} - 559872x_1^{33} - 13670208x_1^{32} + 11197440x_1^{31} - 32799168x_1^{30} + 7301664x_1^{29} + 40124160x_1^{28} - 56581740x_1^{27} + 118393488x_1^{26} - 29030400x_1^{25} - 11429649x_1^{24} + 91968984x_1^{23} - 162286560x_1^{22} + 52664472x_1^{21} - 95470992x_1^{20} - 51948224x_1^{19} + 37314854x_1^{18} - 36173624x_1^{17} + 103156448x_1^{16} + 27660704x_1^{15} + 94133752x_1^{14} + 56849248x_1^{13} + 51186288x_1^{12} + 42348048x_1^{11} + 20765728x_1^{10} + 17391200x_1^9 + 7273168x_1^8 + 4607744x_1^7 + 1946186x_1^6 + 880960x_1^5 + 413632x_1^4 + 86580x_1^3 + 75816x_1^2 + 6561.$

Based on Algorithm `Eliminate`, we obtain the quotients of the division in Step 3 of `sosgradient`: $\phi_1 = 0$ and ϕ_2 given at polsys.lip6.fr/∼hieu/phisos.mm.

By using `univsos2` to compute an SOS decomposition of $\bar{h}$, we obtain the list `sos` given at above link such that $\bar{h} = \sum_{i=1}^{m} \texttt{sos}[2i-1]\texttt{sos}[2i]^2$, where $\texttt{sos}[i]$ stands for the i-th entry of `sos`, m is the half length of `sos`.

Combining the above results, we obtain an SOS of rational fractions modulo the gradient of f_S as in (4.5).

5. Practical experiments. This section is dedicated to showing experimental results obtained by using the algorithms `sosgradientshape` (Algorithm 3.1 from Section 3) and `sosgradient` (Algorithm 4.1 from Section 4). Both algorithms are implemented in MAPLE, and the results are obtained on an Intel Xeon E7-4820 CPU (2GHz) with 1.5 TB of RAM.

In practice, Algorithm `univsos2` runs faster than Algorithm `univsos1`, which is consistent with the theoretical results stated in [27, Theorem 17] and [27, Theorem 24]. In addition, as mentioned in Remark 4.7, it is practically faster to compute SOS decompositions involving rational fractions than polynomials. We compare timings of the slowest algorithm, namely `sosgradientshape` using `univsos1`, with the fastest algorithm, namely `sosgradient` using `univsos2`. For each algorithm, the first step consists of obtaining h by computing either the shape position (using the procedure `Basis` in MAPLE) in `sosgradientshape` or the zero-dimensional rational parametrization (using the procedure `RationalUnivariateRepresentation` in MAPLE) in `sosgradient`. The runtime of this step is denoted by t_h. The degree and the bitsize of h are denoted by d_h and τ_h, respectively. The second step outputs an SOS decomposition of the non-negative univariate polynomial h by using either Algorithm `univsos1` in `sosgradientshape` or Algorithm `univsos2` in `sosgradient`.

Here, $t_{\mathtt{sos}}$ is the runtime of the second step and $\tau_{\mathtt{sos}}$ is the maximum bitsize of the output polynomials.

			`sosgradientshape`					`sosgradient`				
				bitsize 10^6-bits		time (s)			bitsize 10^4-bits		time (s)	
n	τ	δ	d_h	τ_h	$\tau_{\mathtt{sos}}$	t_h	$t_{\mathtt{sos}}$	d_h	τ_h	$\tau_{\mathtt{sos}}$	t_h	$t_{\mathtt{sos}}$
2	74	9	32	0.3	8.1	0.1	2.6	36	0.5	1.6	0.1	1.8
3	149	27	104	2.4	153	1.1	781	108	6.6	13.4	0.2	13.3
4	312	81	320	117	–	399	–	324	88	169	3.9	505
5	590	243	968	–		–		972	940	1306	169	4965

Table 1. Comparison results of output size and performance between Algorithm `sosgradientshape` and Algorithm `sosgradient`

In Table 1, we consider random polynomials of fixed degree $d = 4$ with number of variables n being between 2 and 5, generated as follows: $a^4+b_1^2+\cdots+b_n^2+c+10^6$, where a (resp., b_i, c) is a dense linear (resp., quadratic, cubic) polynomial in n variables. The coefficients of a (resp., b_i, c) are chosen randomly in $\{-1, 1\}$ (resp., $\{-3, \dots, 3\}$, $\{-1, 0, 1\}$) with respect to the uniform distribution. For $n \geq 4$, `sosgradientshape` failed to provide an SOS decomposition as the execution of `univsos1` did not finish after 12 hours of computation, as indicated by the symbol "–" in the corresponding lines. The underlying reason is that τ_h and d_h are both very large and that the complexity of `univsos1` is exponential in the degree of h [27, Theorem 17]. Note that the intermediate polynomials correspond to worst cases, i.e., the maximum possible degree of w is attained, namely $\delta = \deg w = (d-1)^n$, so the degree of h is also maximum, i.e., $\deg h = d(d-1)^n - d$ (resp. $d(d-1)^n$) in `sosgradientshape` (resp. in `sosgradient`). For such cases, `sosgradient` cannot compute decompositions for $n \geq 4$ (corresponding to $\deg h \geq 324$) within 12 hours.

Next, we compare the performance of `sosgradient` (using `univsos2`) and Algorithm `multivsos` [25]. Recall that `multivsos` is designed to compute SOS decompositions of polynomials lying in the interior of the SOS cone. We report our experimental results in Table 2, obtained with seven classes of 50 randomly generated polynomials. The random polynomials corresponding to the four first rows, with $d = 4$ and $n = 2, \dots, 5$, are obtained a similar way: $a^4 + b_1^2 + b_2^2 + c + 10^6$, where a (resp., b_i, c) is a dense linear (resp., quadratic, cubic) polynomial in n variables. The coefficients of a (resp., b_i, c) are chosen randomly in $\{\pm 1, \pm 2\}$ (resp., $\{-3, \dots, 3\}$, $\{-1, \dots, 1\}$) with respect to the uniform distribution. The polynomials from the three last rows, with $d = 6$ and $n = 2, 3, 4$, are constructed in a similar way: $a^6 + b^2 + c + 10^6$, where a (resp., b, c) is a dense linear (resp., cubic, cubic) polynomial in n variables. Coefficients of a (resp., b_i, c) are chosen randomly in $\{\pm 1, \pm 2\}$ (resp., $\{-3, \dots, 3\}$, $\{-1, \dots, 1\}$) with respect to the uniform distribution. Note that here the univariate polynomials generated when running the algorithm do not correspond to the worst case scenario in terms of degree and bitsize. For both algorithms, we denote by τ (10^4-bits) the average bitsize of the output and by t the average runtime in seconds.

	`multivsos`			`sosgradient`	
d, n	success	τ	t	τ	t
4,2	100%	1.3	0.16	2	2
4,3	94%	3.7	0.26	18	22
4,4	38%	8.9	0.18	78	153
4,5	8%	12.5	0.32	234	630
6,2	82%	3.5	0.24	45	142
6,3	0%			160	500
6,4	0%			744	4662

Table 2. Comparison of performance between Algorithm `sosgradientshape` and Algorithm `multivsos`

From this table, we deduce that when the number of variables n increases, then the rate of success of `multivsos` decreases. This fact illustrates Blekherman's theorem [8] which says that if the degree $d \geq 4$ is fixed then, as the number of variables n grows, the cone of non-negative polynomials is significantly bigger than the cone of SOS polynomials. It also illustrates that `sosgradient` can tackle a large range of polynomial optimization problems which are out of reach of state-of-the-art algorithms such as `multivsos`. When `multivsos` succeeds in computing SOS decompositions, then it provides more concise certificates than `sosgradient` while being more efficient. However, when $d = 4$ and $n = 5$, `multivsos` can only decompose four polynomials out of 50 while `sosgradient` succeeds for all of them. This demonstrates the need of alternative procedures such as `sosgradient` for polynomials which presumably do not lie in the interior of the SOS cone.

Conclusions and perspectives. We designed and analyzed two algorithms to decompose a non-negative polynomial as an SOS of polynomials/rational fractions modulo the gradient ideal with rational coefficients. The correctness of our framework relies on a generic condition, namely that the gradient ideal of the input polynomial is zero-dimensional and radical. We shall improve the scalability of our algorithms by exploiting the specific structure of the input polynomial, such as correlative [23] or term sparsity [45], symmetries [35] or by using recent improvements on the computation of critical sets when the related system is invariant under group actions [12]. Furthermore, we also plan to extend our algorithms to the constrained case by relying on polar varieties as in [14] and to extend the result for positive polynomials without imposing the zero-dimensional and radical condition on the gradient ideal.

Acknowledgements. We thank the anonymous reviewers for their comments and suggestions which helped a lot to improve this article.
This work has been supported by European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska–Curie Actions, grant agreement 813211 (POEMA).

Appendix.

Appendix A. Proof of Corollary 2.3. Assume that the system of partial derivatives $\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n}$ is given by a straight-line program Γ of size L, i.e., the program uses L elementary operations $+, -, \times$ to evaluate the system from variables $x_1, \ldots, x_n$ and integers with bitsizes at most $\max_{i=1}^n \{ht(\frac{\partial f}{\partial x_i})\}$.

We claim that L is $O(d\binom{n+d}{d})$. Indeed, f has at most $\binom{n+d}{d}$ terms and each term in f is defined by at most $d+1$ multiplications. Hence, the size of a straight-line program Γ_f which defines f does not exceed $(d+1)\binom{n+d}{d}$. By applying Baur-Strassen Theorem [7, Theorem 1], the size L is $O(d\binom{n+d}{d})$.

Recall that $ht(\frac{\partial f}{\partial x_i}) \leq \log d + ht(f) = \log d + \tau$, for $i = 1, \ldots, n$. By applying [39, Corollary 2] for the system and a single group of variables, there exists an algorithm that takes the system as in input, and that produces one of the outputs given as in items a)– c) of Corollary 2.3. The number of boolean operations of the algorithm is $\widetilde{O}\Big(n^2 d^{2n} (\log d + \tau + (d-1))(d\binom{n+d}{d} + n(d-1) + n^2)\Big)$. Reduce this formula, we get (2.2). Furthermore, the polynomials in the output have degree at most $(d-1)^n$ and bitsize $\widetilde{O}\left((d-1)^n (\log d + \tau + n + (d-1))\right) = \widetilde{O}\left((\tau + n + d)(d-1)^n\right)$ as claimed.

Appendix B. Proof of the bit complexity in Lemma 2.4. From Corollary 2.3, the degree of w is at most $(d-1)^n$, and then $\deg w'$ is at most $(d-1)^n-1$. Assume that β is the positive minimum common denominator of all non-zero coefficients of w. Then, βw and $\beta w'$ belong to $\mathbb{Z}[t]$. Clearly, $\deg(\beta w') = \deg(\beta w) - 1$, $\deg(\beta w) \leq (d-1)^n$, and the bitsize of βw and $\beta w'$ are bounded by $\widetilde{O}\left((\tau + n + d)(d-1)^n\right)$. We can apply [43, Theorem 6.52] to βw and $\beta w'$. The extended Euclidean algorithm computes the Bézout coefficient, denoted by b, of $\beta w'$ using

$$\widetilde{O}(\tau + n + d)^2(d-1)^{6n}) \tag{B.1}$$

boolean operations. The bitsize of b is bounded by

$$O\left((\tau + n + d)(d-1)^{2n}\right). \tag{B.2}$$

Furthermore, one sees that the degree of b satisfies

$$\deg b \leq \deg w - \deg \gcd(w, w') = \deg w \leq (d-1)^n. \tag{B.3}$$

For every $i = 2, \ldots, n$, we will estimate the bitsize of the polynomial $b\kappa_i$. Recall from Corollary 2.3 that $\deg \kappa_i \leq (d-1)^n$, hence from (B.3) one has $\deg b\kappa_i \leq 2(d-1)^n$. From (B.2), we obtain

$$ht(b\kappa_i) \leq ht(b) + ht(\kappa_i) = \widetilde{O}((\tau + n + d)(d-1)^{2n}) + \widetilde{O}((\tau + n + d)(d-1)^n).$$

After simplifying the last estimate, the bitsize of $b\kappa_i$ is bounded from above by $\widetilde{O}((\tau + n + d)(d-1)^{2n})$. Hence, the bitsize of $\eta b\kappa_i$, where η is the minimum common denominator of all non-zero coefficients of $b\kappa_i$, can be estimated as follows

$$ht(\eta b\kappa_i) \leq 2ht(b\kappa_i) \leq \widetilde{O}((\tau + n + d)(d-1)^{2n}).$$

In the proof of Lemma 2.4, we considered the division of $b\kappa_i$ by w and defined $v_i = b\kappa_i \mod w$. Thus, the degree of v_i is at most $\deg w \leq (d-1)^n$. From Lemma 2.5, the Euclidean division algorithm computes v_i using at most

$$\widetilde{O}((\tau + n + d)(d-1)^{5n}) \tag{B.4}$$

boolean operations. Thus, the bitsize of v_i is $\widetilde{O}((\tau+n+d)(d-1)^{3n})$, for $i = 2, \ldots, n$. Therefore, computing $[w, v_2, \ldots, v_n]$ from the zero-dimensional rational parametrization $\mathcal{Q}$ of $V_{grad}(f)$, requires

$$\widetilde{O}((\tau + n + d)^2(d-1)^{6n} + (n-1)(\tau + n + d)(d-1)^{5n})$$

boolean operations, as a consequence of (B.1) and (B.4). By applying further simplification, we obtain the desired result (2.3).

The bit complexity results of the two division algorithms used in Lemma 2.5 and Lemma 2.6 are basic but we could not find their proofs in the literature. Here we state these two algorithms and prove estimates for their bit complexities.

Appendix C. Proof of Lemma 2.5. Assume that a, b are polynomials in $\mathbb{Z}[t]$ with $\deg a = d \geq \deg b = m$ and that $ht(a)$, $ht(b)$ are bounded from above by τ. We recall the Euclidean division algorithm in Algorithm C.1 [43, Algorithm 2.5] to compute the quotient q and the remainder r of the division of a by b, i.e., $a = qb + r$ with $\deg r < \deg b$.

We denote by r_i (resp. q_i, h_i) the value of r (resp. q, h) after the i-th iteration of the while loop from Step 2. The initial values are $q_0 = 0$ and $r_0 = a$. After each

Algorithm C.1 Euclidean division algorithm

Input: polynomials $a, b \in \mathbb{Z}[t]$
Output: polynomials $q, r \in \mathbb{Q}[t]$ such that $a = qb + r$ and $\deg r < \deg b$
1: Let $q := 0$ and $r := a$
2: While $\deg r \geq \deg b$ do
  3: Let $h := lc(r)/lc(b)t^{\deg r - \deg b}$
  4: Let $q := q + h$
  5: Let $r := r - hb$
6: Return q and r

iteration of the while loop, the degree of r is strictly decreasing. Hence, the while loop will terminate after k iterations, where $k \leq d - m$.

We now compute the numbers of boolean operations to perform the operations in Steps 3–5. From $h_i = lc(r_{i-1})/lc(b)t^{\deg r_{i-1} - \deg b}$ in Step 3, we observe that

$$ht(h_i) = \max\{ht(b), ht(r_{i-1})\} \leq \max\{\tau, ht(r_{i-1})\} \leq \tau + ht(r_{i-1}), \tag{C.1}$$

and the number of boolean operations to perform Step 3 is bounded by $\tau + ht(r_{i-1})$. Note that, the number of boolean operations to perform the operation in Step 4 is bounded by $O(1)$. We consider the operation in Step 5, i.e., $r_i = r_{i-1} - h_i b$. The estimate in (C.1) implies $ht(h_i b) \leq 2\tau + ht(r_{i-1})$; then, the bitsize of r_i is bounded by $2\tau + ht(r_{i-1})$. We get the recurrence formula $ht(r_{i+1}) \leq ht(r_i) + 2\tau$, for each $i = 0, \dots, k$, with $ht(r_0) = \tau$. It follows that $ht(r_i) \leq 2i\tau + \tau$, for each $i = 0, \dots, k$. This yields

$$ht(r) = ht(r_k) \leq 2(d - m)\tau + \tau = O((d - m)\tau).$$

In Step 5, the number of boolean operations to compute $h_i b$ is $O(m(\tau + ht(r_{i-1})))$, so r_i is also computed in $O(m(\tau + ht(r_{i-1})))$ boolean operations.

From above, the boolean operations to compute every iteration in Step 2 is $O(m\tau(d - m))$. Since the algorithm has at most $d - m$ iterations, the number of boolean operations to perform the algorithm is $O(m\tau(d - m)^2)$.

To complete the proof, we estimate for the bitsize of q. Since $q_i = q_{i-1} + h_i$, from (C.1), one has

$$ht(q_i) \leq \max\{ht(q_{i-1}), ht(h_i)\} \leq ht(q_{i-1}) + \tau + ht(r_{i-1}).$$

This yields $ht(q) \leq (d - m)\tau + ht(r) = O((d - m)\tau)$. This is the desired estimate.

Appendix D. Algorithm `Eliminate` and the proof of Lemma 2.6.

Algorithm `Eliminate`. Let us consider $g \in \mathbb{Q}[x_1][x_2, \dots, x_n]$, with $\deg g = d$ (in variables $x_2, \dots, x_n$) and $ht(g) = \tau_g$, and the list of rational fractions:

$$G = [x_2 - \frac{a_2}{a_0}, \dots, x_n - \frac{a_n}{a_0}],$$

where $a_0, a_2, \dots, a_n$ are polynomials in $\mathbb{Q}[x_1]$, $a_0 \neq 0$, and $ht(a_i) \leq \tau_a$ for $i = 0, 2, \dots, n$. Recall that $\mathbb{Q}(x_1)$ is the field of rational fractions in variable x_1 with coefficients in $\mathbb{Q}$. Let $x_2 < \cdots < x_n$ be a lexicographic monomial order on $\mathbb{Q}(x_1)[x_2, \dots, x_n]$. Algorithm `Eliminate` outputs the quotients $\phi_2, \dots, \phi_n \in \mathbb{Q}(x_1)[x_2, \dots, x_n]$ and the remainder $r \in \mathbb{Q}(x_1)$ of the multivariate division of g by the list G satisfying

$$g = \sum_{i=2}^{n} \phi_i (x_i - \frac{a_i}{a_0}) + r. \tag{D.1}$$

ALGORITHM D.1 Elimination algorithm

`Eliminate` := proc($g, a_0, a_2, \dots, a_n$)
Input: $n+1$ polynomials $g \in \mathbb{Q}[x_1][x_2, \dots, x_n]$, $a_0, a_2, \dots, a_n \in \mathbb{Q}[x_1]$
Output: $\phi_2, \dots, \phi_n$ in $\mathbb{Q}(x_1)[x_2, \dots, x_n]$ and $r \in \mathbb{Q}(x_1)$ satisfying (D.1)
1: Set $r_{n+1} := g$
2: For $i = n$ to 2 do
  3: Compute $\phi_i := quo(r_{i+1}, x_i - \frac{a_i}{a_0}, x_i)$
  4: Substitute x_i by $\frac{a_i}{a_0}$ in r_{i+1} to define $r_i := r_{i+1}(x_1, \dots, x_{i-1}, \frac{a_i}{a_0})$
5: Set $r := r_2$
6: Return $\phi_2, \dots, \phi_n$, and r

In Step 3, ϕ_i is the quotient of the univariate division (in the variable x_i) of r_{i+1} by $x_i - \frac{a_i}{a_0}$. Since the degree of x_i in $x_i - \frac{a_i}{a_0}$ is 1, ϕ_i belongs to $\mathbb{Q}(x_1)[x_2, \dots, x_i]$. The remainder r_i of the division in Step 3 is given in Step 4 after replacing x_i by $\frac{a_i}{a_0}$ in r_{i+1}; hence one has $r_i \in \mathbb{Q}(x_1)[x_2, \dots, x_{i-1}]$. After Steps 3-4, we obtain

$$r_{i+1} = \phi_i\Big(x_i - \frac{a_i}{a_0}\Big) + r_i. \tag{D.2}$$

Therefore, after Step 5, we get $g = \sum_{i=2}^{n} \phi_i(x_i - \frac{a_i}{a_0}) + r$, with $r \in \mathbb{Q}(x_1)$. Based on Buchberger's Criterion [9], we can show that the system of $n-1$ polynomials $[x_2 - \frac{a_2}{a_0}, \dots, x_n - \frac{a_n}{a_0}]$ is a Gröbner basis of the ideal generated by this system w.r.t. the order $<$ in $\mathbb{Q}(x_1)[x_2, \dots, x_n]$. Hence, $\phi_2, \dots, \phi_n$ are defined uniquely. The correctness of the algorithm is proved.

THE PROOF OF LEMMA 2.6. Now we estimate the bitsizes of ϕ_i, for $i = 2, \dots, n$. From the definition of r_i in Step 4, one sees that $ht(r_i) \le ht(r_{i+1}) + 2d\tau_a$. Since $ht(r_{n+1}) = \tau_g$, the bitsize of r_i is bounded from above by $\tau_g + 2(n-1)d\tau_a$. The relation (D.2) leads to $ht(\phi_i) \le ht(r_{i+1} - r_i) + ht(x_i - \frac{a_i}{a_0})$. Because of $ht(r_{i+1} - r_i) \le \max\{ht(r_{i+1}), ht(r_i)\}$, and $ht(\frac{a_i}{a_0}) \le 2\tau_a$, we get $ht(\phi_i) \le \tau_g + 2(nd - d + 1)\tau_a$. It follows that $ht(\phi_i) = \widetilde{O}(\tau_g + nd\tau_a)$.

We see that the number of boolean operations to perform Steps 3 and 4 are $\widetilde{O}(\tau_g + nd\tau_a)$ and $O(1)$, respectively. The for loop in Step 2 has $n-1$ steps. Therefore, the number of boolean operations to perform the loop is $\widetilde{O}(n\tau_g + n^2 d\tau_a)$. This is also the number of boolean operations that Algorithm `Eliminate` uses.

REFERENCES

[1] E. ARTIN, *Über die zerlegung definiter funktionen in quadrate*, in Abhandlungen aus dem mathematischen Seminar der Universität Hamburg, vol. 5, Springer, 1927, pp. 100–115.
[2] B. BANK, M. GIUSTI, J. HEINTZ, M. S. EL DIN, AND E. SCHOST, *On the geometry of polar varieties*, Applicable Algebra in Engineering, Communication and Computing, 21 (2010), pp. 33–83.
[3] B. BANK, M. GIUSTI, J. HEINTZ, AND G. M. MBAKOP, *Polar varieties and efficient real elimination*, Mathematische Zeitschrift, 238 (2001), pp. 115–144.
[4] B. BANK, M. GIUSTI, J. HEINTZ, AND L. M. PARDO, *Generalized polar varieties: Geometry and algorithms*, Journal of complexity, 21 (2005), pp. 377–412.
[5] B. BANK, M. GIUSTI, J. HEINTZ, AND M. SAFEY EL DIN, *Intrinsic complexity estimates in polynomial optimization*, Journal of Complexity, 30 (2014), pp. 430–443.
[6] S. BASU, R. POLLACK, AND M.-F. ROY, *A new algorithm to find a point in every cell defined by a family of polynomials*, in Quantifier elimination and cylindrical algebraic decomposition, Springer, 1998, pp. 341–350.

[7] W. Baur and V. Strassen, *The complexity of partial derivatives*, Theoretical computer science, 22 (1983), pp. 317–330.
[8] G. Blekherman, *There are significantly more nonegative polynomials than sums of squares*, Israel Journal of Mathematics, 153 (2006), pp. 355–380.
[9] B. Buchberger, *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*, PhD thesis, Universitat Insbruck, (1965).
[10] S. Chevillard, J. Harrison, M. Joldeş, and C. Lauter, *Efficient and accurate computation of upper bounds of approximation errors*, Theoretical Computer Science, 412 (2011), pp. 1523–1543.
[11] D. Cox, J. Little, and D. OShea, *Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra*, Springer Science & Business Media, 2013.
[12] J.-C. Faugère, G. Labahn, M. Safey El Din, É. Schost, and T. X. Vu, *Computing critical points for invariant algebraic systems*, arXiv preprint arXiv:2009.00847, (2020).
[13] P. Gianni and T. Mora, *Algebraic solution of systems of polynomial equations using Gröbner bases*, in Applied Algebra, Algebraic Algorithms and Error Correcting Codes, Proceedings of AAECC-5, volume 356 of LNCS, Springer, 1989, pp. 247–257.
[14] A. Greuet, F. Guo, M. Safey El Din, and L. Zhi, *Global optimization of polynomials restricted to a smooth variety using sums of squares*, Journal of Symbolic Computation, 47 (2012), pp. 503–518.
[15] A. Greuet and M. Safey El Din, *Probabilistic algorithm for polynomial optimization over a real algebraic set*, SIAM Journal on Optimization, 24 (2014), pp. 1313–1343.
[16] D. Y. Grigor'ev and N. N. Vorobjov Jr, *Solving systems of polynomial inequalities in subexponential time*, Journal of symbolic computation, 5 (1988), pp. 37–64.
[17] D. Henrion, S. Naldi, and M. Safey El Din, *Exact algorithms for linear matrix inequalities*, SIAM Journal on Optimization, 26 (2016), pp. 2512–2539.
[18] E. Kaltofen, B. Li, Z. Yang, and L. Zhi, *Exact certification of global optimality of approximate factorizations via rationalizing sums-of-squares with floating point scalars*, in Proceedings of the twenty-first international symposium on Symbolic and algebraic computation, 2008, pp. 155–164.
[19] E. Kaltofen, Z. Yang, and L. Zhi, *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, 2009, pp. 65–70.
[20] E. L. Kaltofen, B. Li, Z. Yang, and L. Zhi, *Exact certification in global polynomial optimization via sums-of-squares of rational functions with rational coefficients*, Journal of Symbolic Computation, 47 (2012), pp. 1–15.
[21] E. Landau, *Über die darstellung definiter funktionen durch quadrate*, Mathematische Annalen, 62 (1906), pp. 272–285.
[22] J. B. Lasserre, *Global optimization with polynomials and the problem of moments*, SIAM Journal on optimization, 11 (2001), pp. 796–817.
[23] J. B. Lasserre, *Convergent SDP-relaxations in polynomial optimization with sparsity*, SIAM Journal on Optimization, 17 (2006), pp. 822–843.
[24] V. Magron and M. Safey El Din, *On Exact Polya and Putinar's Representations*, in Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation, 2018, pp. 279–286.
[25] V. Magron and M. Safey El Din, *RealCertify: a Maple package for certifying non-negativity*, ACM Communications in Computer Algebra, 52 (2018), pp. 34–37.
[26] V. Magron and M. Safey El Din, *On Exact Reznick, Hilbert-Artin and Putinar's Representations*, Journal of Symbolic Computation, 107 (2021), pp. 221–250.
[27] V. Magron, M. Safey El Din, and M. Schweighofer, *Algorithms for weighted sum of squares decomposition of non-negative univariate polynomials*, Journal of Symbolic Computation, 93 (2019), pp. 200–220.
[28] V. Magron, H. Seidler, and T. de Wolff, *Exact optimization via sums of nonnegative circuits and arithmetic-geometric-mean-exponentials*, in Proceedings of the 2019 on International Symposium on Symbolic and Algebraic Computation, 2019, pp. 291–298.
[29] M. Marshall, *Positive polynomials and sums of squares*, no. 146, American Mathematical Soc., 2008.
[30] J. Nie, J. Demmel, and B. Sturmfels, *Minimizing polynomials via sum of squares over the gradient ideal*, Mathematical Programming, 106 (2006), pp. 587–606.
[31] P. A. Parrilo, *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*, PhD thesis, California Institute of Technology, 2000.
[32] P. A. Parrilo, *An explicit construction of distinguished representations of polynomials non-*

negative over finite sets, IfA AUT02-02, ETH Zürich, (2002).

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

[34] Y. Pourchet, *Sur la représentation en somme de carrés des polynômes à une indéterminée sur un corps de nombres algébriques*, Acta Arithmetica, 19 (1971), pp. 89–104.

[35] C. Riener and M. Safey el Din, *Real root finding for equivariant semi-algebraic systems*, in Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation, 2018, pp. 335–342.

[36] R. M. Robinson, *Some definite polynomials which are not sums of squares of real polynomials*, Selected questions of algebra and logic (collection dedicated to the memory of A. I. Malcev) (Russian), (1973), pp. 264–282.

[37] F. Rouillier, *Solving zero-dimensional systems through the rational univariate representation*, Applicable Algebra in Engineering, Communication and Computing, 9 (1999), pp. 433–461.

[38] M. Safey El Din and E. Schost, *Polar varieties and computation of one point in each connected component of a smooth real algebraic set*, in Proceedings of the 2003 international symposium on Symbolic and algebraic computation, 2003, pp. 224–231.

[39] M. Safey El Din and É. Schost, *Bit complexity for multi-homogeneous polynomial system solving–Application to polynomial minimization*, Journal of Symbolic Computation, 87 (2018), pp. 176–206.

[40] M. Safey El Din and L. Zhi, *Computing rational points in convex semi-algebraic sets and sums-of-squares decompositions*, SIAM Journal on Optimization, 20 (2010), pp. 2876–2889.

[41] C. Scheiderer, *Sums of squares of polynomials with rational coefficients*, Journal of the European Mathematical Society, 18 (2016), pp. 1495–1513.

[42] M. Schweighofer, *Algorithmische beweise für nichtnegativ-und positivstellensätze*, Master's thesis, Universität Passau, 136 (1999).

[43] J. Von Zur Gathen and J. Gerhard, *Modern Computer Algebra*, Cambridge University Press, 2013.

[44] J. Wang and V. Magron, *A second order cone characterization for sums of nonnegative circuits*, in Proceedings of the 45th International Symposium on Symbolic and Algebraic Computation, 2020, pp. 450–457.

[45] J. Wang, V. Magron, and J.-B. Lasserre, *Chordal-TSSOS: a moment-SOS hierarchy that exploits term sparsity with chordal extension*, SIAM Journal on Optimization, 31 (2021), pp. 114–141.