Triangular Decomposition of Semi-algebraic Systems

Changbo Chen [a] James H. Davenport [b] John P. May [c]
Marc Moreno Maza [a] Bican Xia [d] Rong Xiao [a]

[a] *University of Western Ontario London, Ontario, Canada N6A 5B7*
[b] *University of Bath, Bath BA2 7AY, United Kingdom*
[c] *Maplesoft, Waterloo, Ontario N2V 1K8, Canada*
[d] *Peking University, Beijing 100871, China*

Abstract

Regular chains and triangular decompositions are fundamental and well-developed tools for describing the complex solutions of polynomial systems. This paper proposes adaptations of these tools focusing on solutions of the real analogue: semi-algebraic systems.

We show that any such system can be decomposed into finitely many *regular semi-algebraic systems*. We propose two specifications (full and lazy) of such a decomposition and present corresponding algorithms. Under some simplifying assumptions, the lazy decomposition can be computed in singly exponential time w.r.t. the number of variables. We have implemented our algorithms and present experimental results illustrating their effectiveness.

Key words: Regular semi-algebraic system, regular chain, lazy decomposition, triangular decomposition, border polynomial, fingerprint polynomial set.

1. Introduction

Regular chains, the output of triangular decompositions of systems of polynomial equations, enjoy remarkable properties. Size estimates play in their favor [20] and permit the design of modular [21] and fast [30] methods for computing triangular decompositions. These features stimulate the development of algorithms and software for solving polynomial systems via triangular decompositions.

⋆ This research was partly supported by NSERC, Maplesoft and MITACS of Canada

Email addresses: cchen252@csd.uwo.ca (Changbo Chen), J.H.Davenport@bath.ac.uk (James H. Davenport), jmay@maplesoft.com (John P. May), moreno@csd.uwo.ca (Marc Moreno Maza), xbc@math.pku.edu.cn (Bican Xia), rong@csd.uwo.ca (Rong Xiao).

For the fundamental case of semi-algebraic systems with rational number coefficients, to which this paper is devoted, several algorithms for studying the real solutions of such systems take advantage of the structure of a regular chain. Some are specialized to isolating the real solutions of systems with finitely many complex solutions [39, 16, 3]. Other algorithms deal with parametric polynomial systems via real root classification (RRC) [41] or with arbitrary systems via cylindrical algebraic decompositions (CAD) [15].

In this paper, we introduce the notion of a *regular semi-algebraic system*, which in broad terms is the "real" counterpart of the notion of a regular chain. Then we define two notions of a *decomposition of a semi-algebraic system*: one that we call *lazy triangular decomposition*, where the analysis of components of strictly smaller (complex) dimension is deferred, and one that we call *full triangular decomposition* where all cases are worked out. These decompositions are obtained by combining triangular decompositions of algebraic sets over the complex field with a special Quantifier Elimination (QE) method based on RRC techniques.

Definition 1. Let $T \subset \mathbb{Q}[\mathbf{x}]$ be a squarefree regular chain for an ordering of the variables $\mathbf{x} = x_1, \ldots, x_n$. Let $\mathbf{u} = u_1, \ldots, u_d$ and $\mathbf{y} = y_1, \ldots, y_{n-d}$ designate respectively the variables of $\mathbf{x}$ that are free and algebraic w.r.t. T. Let $P \subset \mathbb{Q}[\mathbf{x}]$ be finite and such that each polynomial in P is regular w.r.t. the saturated ideal of T. Define $P_> := \{p > 0 \mid p \in P\}$. Let $\mathcal{Q}$ be a quantifier-free formula over $\mathbb{Q}[\mathbf{x}]$ involving only the $\mathbf{u}$ variables. Let S be the semi-algebraic subset of $\mathbb{R}^d$ defined by $\mathcal{Q}$. When $d = 0$, the 0-ary Cartesian product $\mathbb{R}^d$ is treated as a singleton set. We say that $R := [\mathcal{Q}, T, P_>]$ (also written as $[R^Q, R^T, R^P]$) is a *regular semi-algebraic system* if:

(i) S is a non-empty open subset in $\mathbb{R}^d$,

(ii) the regular system $[T, P]$ specializes well at every point u of S (see Section 2 for this notion),

(iii) at each point u of S, the specialized system $[T(u), P(u)_>]$ admits real solutions.

The *zero set* of R, denoted by $Z_{\mathbb{R}}(R)$, is the set of points $(u, y) \in \mathbb{R}^d \times \mathbb{R}^{n-d}$ such that $\mathcal{Q}(u)$ holds and $t(u, y) = 0$, $p(u, y) > 0$, for all $t \in T$ and all $p \in P$.

Using the notations of Definition 1, Let $R = [\mathcal{Q}, T, P_>]$ be a regular semi-algebraic system. Since $\mathcal{Q}$ is open, each connected component C of $\mathcal{Q}$ is locally homeomorphic to the hypercube $(0, 1)^d$. From Property (ii), the zero set $Z_{\mathbb{R}}(R)$ consists of disjoint graphs of continuous semi-algebraic functions defined on each such C. Moreover, from Property (iii), there is at least one such graph. For these reasons, which are formally stated in Theorem 1, the regular semi-algebraic system R can be understood as a parameterization of the set $Z_{\mathbb{R}}(R)$. Clearly, the dimension of $Z_{\mathbb{R}}(R)$ is d.

Example 1. For the variables $z > y > x$, we consider two classical surfaces (from the Algebraic Surface Gallery[1]) called *Sofa* and *Cylinder* with equations:

$$x^2 + y^3 + z^5 = 0 \;\text{ and }\; x^4 + y^2 = 1.$$

The common points of these surfaces with real coordinates can be described as the union of the zero sets of the following 5 regular semi-algebraic systems R_1 to R_5 (unspecified

[1] www1-c703.uibk.ac.at/mathematik/project/bildergalerie/gallery.html

R_i^P are empty and unspecified R_i^Q are "true"):

$$R_1^T = \begin{cases} z^5 + (1-x^4)y + x^2 \\ y^2 + x^4 - 1 \end{cases} \qquad R_2^T = \begin{cases} z+1 \\ y \\ x-1 \end{cases} \qquad R_3^T = \begin{cases} z+1 \\ y \\ x+1 \end{cases}$$

$$R_1^Q = \begin{cases} -1 < x < 1 \\ x^{12} - 3x^8 + 4x^4 - 1 \neq 0, \end{cases}$$

$$R_4^T = \begin{cases} z^5 + (1-x^4)y + x^2 \\ (x^4-1)y + x^2 \\ x^{12} - 3x^8 + 4x^4 - 1 \end{cases} \qquad R_5^T = \begin{cases} z \\ (x^4-1)y - x^2 \\ x^{12} - 3x^8 + 4x^4 - 1 \end{cases}$$

This decomposition is obtained by the algorithms of Section 7. The fact that R_2 to R_5 are regular semi-algebraic systems is clear, since each of them consists only of a zero-dimensional squarefree regular chain. For R_1, we observe that

$$(-1 < x < 1) \quad \wedge \quad (x^{12} - 3x^8 + 4x^4 - 1 \neq 0)$$

is a quantifier-free formula[2] defining an open set S; moreover $p_y := y^2 + x^4 - 1$, regarded as a univariate polynomial in y, admits two distinct real roots for each $x \in S$ while $p_z := z^5 + (1-x^4)y + x^2$, as a univariate polynomial in z, is squarefree and admits (exactly) one real root for any $x \in S$ and any y defined by $y^2 + x^4 - 1 = 0$. Indeed, the discriminant of p_z in z is $3125\ \left(-y + yx^4 - x^2\right)^4$ and the resultant w.r.t. y of this latter polynomial and p_y is $9765625\ \left(x^{12} - 3\,x^8 + 4\,x^4 - 1\right)^4$.

In Section 3 we show that the zero set of any semi-algebraic system $\mathfrak{S}$ can be decomposed as a finite union of zero sets of regular semi-algebraic systems. We call such a decomposition a *full triangular decomposition* (or simply *triangular decomposition* when clear from context) of $\mathfrak{S}$, and denote by RealTriangularize an algorithm to compute it.

The existence of such a triangular decomposition can be understood in terms of CAD. Indeed, consider a CAD of the polynomials defining $\mathfrak{S}$ and a cell C where all constraints of $\mathfrak{S}$ are satisfied. The cell C is a connected semi-algebraic set homeomorphic to hypercube $(0,1)^d$, for some d, and from the CAD data (see for instance [15]) one can extract a regular semi-algebraic system R whose zero set is C. However, we should stress the fact that a triangular decomposition of $\mathfrak{S}$ has much less information and structure than a CAD of the polynomials defining $\mathfrak{S}$. For instance, the zero sets of the regular semi-algebraic systems in a triangular decomposition of $\mathfrak{S}$ need not be cylindrically arranged.

Our motivations in introducing this concept of triangular decomposition are threefold. First, we aim at proposing an encoding of the solutions of an arbitrary semi-algebraic system which, as much as possible, is both explicit (thus using "triangular representation" of the components) and compact (thus trying to keep the size of output under control). Secondly, we aim at developing algorithms that are capable of producing either a full description of the solution set, or partial answers (such as dimension information or sample points) at a lower cost than a full description. Thirdly, we aim at proposing an

[2] We said 'involving only strict inequalities', but we are using the shorthand $f \neq 0$ for $f > 0 \vee f < 0$.

encoding of semi-algebraic sets that can support efficient algorithms for the set theoretical operations on such sets. We believe that the work reported here address the former two motivations while the latter one is the subject of [12].

Triangular decomposition of algebraic sets come in two flavors, recalled in Section 2. The first one, proposed by Kalkbrener in [26], focuses on representing the generic points of the irreducible components of the input algebraic set. In [36], Szántó establishes that this representation is computable in singly exponential time w.r.t. the number of variables.

The second one, introduced by Wu [38] and studied by many authors (see [14] and the references therein) represents all the points of the input algebraic set. Our proposed algorithm, RealTriangularize, leads to triangular decompositions of this second type for which it is not known whether or not they can be computed in singly exponential time w.r.t. the number of variables. Meanwhile, we are hoping to obtain an algorithm for decomposing semi-algebraic systems (certainly under some genericity assumptions) that would fit in that complexity class. Moreover, we observe that, in practice, full triangular decompositions are not always necessary and providing information about the components of maximum dimension is often sufficient. These theoretical and practical considerations yield a weaker notion of a decomposition of a semi-algebraic system.

Definition 2. Let $\mathfrak{S} = [F, N_{\geq}, P_{>}, H_{\neq}]$ (see Section 3 for this notation) be a semi-algebraic system of $\mathbb{Q}[\mathbf{x}]$ and $Z_{\mathbb{R}}(\mathfrak{S}) \subseteq \mathbb{R}^n$ be its zero set. Denote by d the dimension of the constructible set $\{x \in \mathbb{C}^n \mid f(x) = 0, g(x) \neq 0, \text{ for all } f \in F, g \in P \cup H\}$. A finite set of regular semi-algebraic systems $\{R_i \mid i = 1 \cdots t\}$ is called a *lazy triangular decomposition* of $\mathfrak{S}$ if

- $\cup_{i=1}^{t} Z_{\mathbb{R}}(R_i) \subseteq Z_{\mathbb{R}}(\mathfrak{S})$ holds, and
- there exists $G \subset \mathbb{Q}[\mathbf{x}]$ such that the real-zero set $Z_{\mathbb{R}}(G) \subset \mathbb{R}^n$ contains $Z_{\mathbb{R}}(\mathfrak{S}) \setminus (\cup_{i=1}^{t} Z_{\mathbb{R}}(R_i))$ and the complex-zero set $V(G) \subset \mathbb{C}^n$ either is empty or has dimension less than d.

We denote by LazyRealTriangularize an algorithm computing such a decomposition. In our software implementation presented hereafter, LazyRealTriangularize outputs additional information in order to continue the computations and obtain a full triangular decomposition, if needed. This additional information appears in the form of *unevaluated recursive calls*, explaining the usage of the adjective *lazy* in this type of decompositions.

Complexity results for lazy triangular decomposition. In Section 4, we provide a running time estimate for computing a lazy triangular decomposition of the semi-algebraic system $\mathfrak{S}$ when $\mathfrak{S}$ has no inequations nor inequalities, (that is, when $N_{\geq} = P_{>} = H_{\neq} = \emptyset$ holds) and when F generates a strongly equidimensional ideal of dimension d. We show that one can compute such a decomposition in time singly exponential w.r.t. n. Our estimates are not sharp and are just meant to reach a singly exponential bound. We rely on the work of J. Renagar [33] for QE. In Sections 5, 6 and 7 we turn our attention to algorithms that are more suitable for implementation even though they rely on sub-algorithms with a doubly exponential running time w.r.t. d.

A special case of quantifier elimination. By means of triangular decomposition of algebraic sets over $\mathbb{C}$, triangular decomposition of semi-algebraic systems (both full and lazy) reduces to a special case of QE. In Section 5, we perform this latter step via the concept of a *fingerprint polynomial set*, which is inspired by that of a *discrimination polynomial set* used for RRC in [41, 40].

Complexity results for fingerprint polynomial set. In Section 6, we show that the fingerprint polynomial set of a pre-regular semi-algebraic system R (See Section 3 for this notion) can be computed in singly exponential time w.r.t. the number of variables as long as the regular chain part of R is in generic position. The advantage of this result, compared to that of Section 4, is that its proof leads to a practical algorithm, actually used in our software implementation. Despite its stronger assumptions, this latter result is practically important since regular chains are often in generic position.

Implementation and experimental results. In Section 7 we describe the algorithms that we have implemented for computing triangular decompositions of semi-algebraic systems. Our MAPLE code is part of the `RegularChains` library. We provide experimental data for two groups of well-known problems. In the first group, each input semi-algebraic system consists of equations only while the second group is a collection of semi-algebraic systems from QE problems. To illustrate the difficulty of our test problems, and only for this purpose, we provide timings obtained with other well-known polynomial system solvers which are based on algorithms whose running time estimates are comparable to ours. For this first group we use MAPLE's `Groebner:-Basis` command for computing lexicographical Gröbner bases. For the second group we use a general purpose QE software, QEPCAD B (in non-interactive mode) [6], on the respective QE problems. Our results show that LazyRealTriangularize code solves most of our test problems and more problems than the tools it is compared to, though these solving tools have different specifications.

With respect to our ISSAC 2010 article [11], the present paper offers the following enhancements and extensions. First of all, Section 6 and its results are new developments. Secondly, Section 7 contains a new algorithm called SamplePoints which appears to be useful on problems for which RealTriangularize is too expensive. Section 9 is also a completely new section which illustrates the solving tools presented in this paper, namely RealTriangularize, LazyRealTriangularize and SamplePoints, on concrete applications. Section 3 contains new results such as Theorem 1, which is essential for Section 7. Sections 4 and 5 provide more details. Finally, Section 10 is another new section, offering a discussion and concluding remarks,

We conclude this introduction by computing a triangular decomposition of a particular semi-algebraic system taken from [8]. Consider the following question: when does $p(z) = z^3 + az + b$ have a non-real root $x + iy$ satisfying $xy < 1$? This problem can be expressed as $(\exists x)(\exists y)[f = g = 0 \wedge y \neq 0 \wedge xy - 1 < 0]$, where $f = \mathrm{Re}(p(x + iy)) = x^3 - 3xy^2 + ax + b$ and $g = \mathrm{Im}(p(x + iy))/y = 3x^2 - y^2 + a$. We call our LazyRealTriangularize command on the semi-algebraic system $f = 0, g = 0, y \neq 0, xy - 1 < 0$ with the variable order $y > x > b > a$. Its first step is to call the Triangularize command of the `RegularChains` library on the algebraic system $f = g = 0$. We obtain one squarefree regular chain $T = [t_1, t_2]$, where $t_1 = g$ and $t_2 = 8x^3 + 2ax - b$, satisfying $V(f, g) = V(T)$. The second step of LazyRealTriangularize is to check whether the polynomials defining inequalities and inequations are regular w.r.t. the saturated ideal of T, which is the case here. The third step is to compute the so called *border polynomial set* (see Section 2) which is $B = [h_1, h_2]$ with $h_1 = 4a^3 + 27b^2$ and $h_2 = -4a^3b^2 - 27b^4 + 16a^4 + 512a^2 + 4096$. One can check that the regular system $[T, \{y, xy - 1\}]$ specializes well outside of the hypersurface $h_1h_2 = 0$. The fourth step is to compute the fingerprint polynomial set which yields the quantifier-free formula $\mathcal{Q} = h_1 > 0 \wedge h_2 \neq 0$ telling us that $[\mathcal{Q}, T, 1 - xy > 0]$ is a regular semi-algebraic system. After performing these four steps, (based on Algorithm 5, Section 7) the function call $\mathsf{LazyRealTriangularize}([f, g, y \neq 0, xy - 1 < 0], [y, x, b, a])$ in our implementation returns the following:

$$\begin{cases} [[t_1 = 0, t_2 = 0, 1 - xy > 0]] & h_1 > 0 \wedge h_2 \neq 0 \\ \%\texttt{LazyRealTriangularize}([t_1 = 0, t_2 = 0, f = 0, & \\ h_1 = 0, 1 - xy > 0, y \neq 0], [y, x, b, a]) & h_1 = 0 \\ \%\texttt{LazyRealTriangularize}([t_1 = 0, t_2 = 0, f = 0, & \\ h_2 = 0, 1 - xy > 0, y \neq 0], [y, x, b, a]) & h_2 = 0 \\ [\,] & \text{otherwise} \end{cases}$$

The above output shows that $\{[\mathcal{Q}, T, 1-xy > 0]\}$ forms a lazy triangular decomposition of the input semi-algebraic system. Moreover, together with the output of the recursive calls, one obtains a full triangular decomposition. Note that the cases of the two recursive calls correspond to $h_1 = 0$ and $h_2 = 0$. Since LazyRealTriangularize uses the Maple piecewise structure for output format, one simply needs to evaluate the recursive calls with the `value` command, yielding the same result as directly calling RealTriangularize

$$\begin{cases} [[t_1 = 0, t_2 = 0, 1 - xy > 0]] & h_1 > 0 \wedge h_2 \neq 0 \\ [\,] & h_1 = 0 \\ [[t_3 = 0, t_4 = 0, h_2 = 0]] & h_2 = 0 \\ [\,] & \text{otherwise} \end{cases}$$

where $t_3 = xy + 1$ and $t_4 = 2a^3x - a^2b + 32ax - 48b + 18xb^2$.

From this output, after some simplification, one could obtain the equivalent quantifier-free formula, $4a^3 + 27b^2 > 0$, of the original QE problem.

2. Triangular decomposition of algebraic sets

This section reviews the basic notions related to regular chains and triangular decompositions of algebraic sets. Throughout this paper, $\mathbf{k}$ is a field of characteristic 0 and $\mathbf{K}$ is its algebraic closure. Let $\mathbf{k}[\mathbf{x}]$ be the polynomial ring over $\mathbf{k}$ and with ordered variables $\mathbf{x} = x_1 < \cdots < x_n$. Let $p, q \in \mathbf{k}[\mathbf{x}]$. Assume that $p \notin \mathbf{k}$. Then denote by $\mathrm{mvar}(p)$, $\mathrm{init}(p)$, and $\mathrm{mdeg}(p)$ respectively the greatest variable appearing in p (called the *main variable* of p), the leading coefficient of p w.r.t. $\mathrm{mvar}(p)$ (called the *initial* of p), and the degree of p w.r.t. $\mathrm{mvar}(p)$ (called the *main degree* of p); denote by $\mathrm{der}(p)$ the derivative of p w.r.t. $\mathrm{mvar}(p)$ and by $\mathrm{discrim}(p)$ the discriminant of p w.r.t. $\mathrm{mvar}(p)$. Let $F \subset \mathbf{k}[\mathbf{x}]$. If $F = \emptyset$, define $\prod_{f \in F} f$ as 1; otherwise define $\prod_{f \in F} f$ as the product of polynomials in F.

Triangular set. Let $T \subset \mathbf{k}[\mathbf{x}]$ be a *triangular set*, that is, a set of non-constant polynomials with pairwise distinct main variables. Denote by $\mathrm{mvar}(T)$ the set of main variables of the polynomials in T. A variable v in $\mathbf{x}$ is called *algebraic* w.r.t. T if $v \in \mathrm{mvar}(T)$, otherwise it is said *free* w.r.t. T. If no confusion is possible, we shall always denote by $\mathbf{u} = u_1, \ldots, u_d$ and $\mathbf{y} = y_1, \ldots, y_m$ respectively the free and the main variables of T. We let $d = 0$ whenever T has no free variables. Let h_T be the product of the initials of the polynomials in T. We denote by $\mathrm{sat}(T)$ the *saturated ideal* of T: if T is the empty triangular set, then $\mathrm{sat}(T)$ is defined as the trivial ideal $\langle 0\rangle$, otherwise it is the ideal $\langle T\rangle : h_T^{\infty}$. The *quasi-component* $W(T)$ of T is defined as $V(T) \setminus V(h_T)$. Denote $\overline{W(T)} = V(\mathrm{sat}(T))$ as the Zariski closure of $W(T)$.

Iterated resultant. Let $p, q \in \mathbf{k}[\mathbf{x}]$. Assume $q \notin \mathbf{k}$ holds. Let $v = \mathrm{mvar}(q)$. We define $\mathrm{res}(p, q, v)$ as follows: if v does not appear in p, then $\mathrm{res}(p, q, v) = p$; otherwise $\mathrm{res}(p, q, v)$ is the resultant of p and q w.r.t. v. Let T be a triangular set of $\mathbf{k}[\mathbf{x}]$. We define $\mathrm{res}(p, T)$ by induction: if T is empty, then $\mathrm{res}(p, T) = p$; otherwise let v be the greatest variable appearing in T, then $\mathrm{res}(p, T) = \mathrm{res}(\mathrm{res}(p, T_v, v), T_{<v})$, where T_v and $T_{<v}$ denote respectively the polynomials of T with main variables equal to and less than v.

Regular chain. A triangular set $T \subset \mathbf{k}[\mathbf{x}]$ is a *regular chain* if: either T is empty; or (letting t be the polynomial in T with maximum main variable), $T \setminus \{t\}$ is a regular chain, and the initial of t is regular (that is neither zero nor zero divisor) modulo $\mathrm{sat}(T \setminus \{t\})$. The empty regular chain is denoted by $\varnothing$. Let $H \subset \mathbf{k}[\mathbf{x}]$. The pair $[T, H]$ is a *regular system* if each polynomial in H is regular modulo $\mathrm{sat}(T)$. If h is the only element in H, we also write as $[T, h]$ for short. A regular chain T or a regular system $[T, H]$, is *squarefree* if for all $t \in T$, the $\mathrm{der}(t)$ is regular w.r.t. $\mathrm{sat}(T)$.

Triangular decomposition. Let $F \subset \mathbf{k}[\mathbf{x}]$. Regular chains $T_1, \ldots, T_e$ of $\mathbf{k}[\mathbf{x}]$ form a *triangular decomposition* of $V(F)$ if either: $V(F) = \cup_{i=1}^{e} \overline{W(T_i)}$ (Kalkbrener's sense [26]) or $V(F) = \cup_{i=1}^{e} W(T_i)$ (Lazard's sense). In this paper, we denote by $\mathsf{Triangularize}$ an algorithm, such as the one of [31, 14], computing a Kalkbrener triangular decomposition.

Regularization. Let $p \in \mathbf{k}[\mathbf{x}]$ and $T \subset \mathbf{k}[\mathbf{x}]$ be a regular chain. The function call $\mathsf{Regularize}(p, T)$ computes a set of regular chains $\{T_1, \ldots, T_e\}$ such that (1) for each $i = 1 \cdots e$, either $p \in \mathrm{sat}(T_i)$ or p is regular w.r.t. $\mathrm{sat}(T_i)$; (2) we have $\overline{W(T)} = \overline{W(T_1)} \cup \cdots \cup \overline{W(T_e)}$, and $\mathrm{mvar}(T) = \mathrm{mvar}(T_i)$ for each $i = 1 \cdots e$.

Good specialization [13]. Consider a squarefree regular system $[T, H]$ of $\mathbf{k}[\mathbf{u}, \mathbf{y}]$. Recall that $\mathbf{y}$ and $\mathbf{u} = u_1, \ldots, u_d$ stand respectively for $\mathrm{mvar}(T)$ and $\mathbf{x} \setminus \mathbf{y}$. Let $z = (z_1, \ldots, z_d)$ be a point of $\mathbf{K}^d$. We say that $[T, H]$ *specializes well* at z if: (i) none of the initials of the polynomials in T vanishes modulo the ideal $\langle z_1 - u_1, \ldots, z_d - u_d\rangle$; (ii) the image of $[T, H]$ modulo $\langle z_1 - u_1, \ldots, z_d - u_d\rangle$ is a squarefree regular system.

Border polynomial [41]. Let $[T, H]$ be a squarefree regular system of $\mathbf{k}[\mathbf{u}, \mathbf{y}]$. Let bp be the primitive and square free part of the product of all $\mathrm{res}(\mathrm{der}(t), T)$ and all $\mathrm{res}(h, T)$ for $h \in H$ and $t \in T$. We call bp the *border polynomial* of $[T, H]$ and denote by $\mathsf{BorderPolynomial}(T, H)$ an algorithm to compute it. We call the set of irreducible factors of bp the *border polynomial set* of $[T, H]$. Denote by $\mathsf{BorderPolynomialSet}(T, H)$ an algorithm to compute it. Proposition 1 follows from the specialization property of subresultants and states a fundamental property of border polynomials.

Proposition 1. The system $[T, H]$ specializes well at $u \in \mathbf{K}^d$ if and only if the border polynomial $bp(u) \neq 0$.

Corollary 1. Let $[T, H]$ be a squarefree regular system of $\mathbf{k}[\mathbf{u}, \mathbf{y}]$ and B be its border polynomial set. Let $D \subset \mathbf{k}[\mathbf{u}]$ such that $B \subseteq D$. Then we have

$$V(\mathrm{sat}(T)) \setminus V(\prod_{h \in H} h) \setminus V(\prod_{f \in D} f) = W(T) \setminus V(\prod_{f \in D} f)$$

and $V(\mathrm{sat}(T)) \cap V(\prod_{h \in H} h) \setminus V(\prod_{f \in D} f) = \emptyset$ hold.

3. Triangular decomposition of semi-algebraic systems

In this section, we prove that any semi-algebraic system decomposes into finitely many regular semi-algebraic systems. This latter notion was defined in the introduction.

Semi-algebraic system. Let us consider four finite polynomial subsets $F = \{f_1, \ldots, f_s\}$, $N = \{n_1, \ldots, n_t\}$, $P = \{p_1, \ldots, p_r\}$ and $H = \{h_1, \ldots, h_\ell\}$ of $\mathbb{Q}[x_1, \ldots, x_n]$. Let $N_\geq$ denote the set of the inequalities $\{n_1 \geq 0, \ldots, n_t \geq 0\}$. Let $P_>$ denote the set of the inequalities $\{p_1 > 0, \ldots, p_r > 0\}$. Let $H_{\neq}$ denote the set of inequations $\{h_1 \neq 0, \ldots, h_\ell \neq 0\}$. We will denote by $[F, P_>]$ the *basic semi-algebraic system* $\{f_1 = 0, \ldots, f_s = 0, p_1 > 0, \ldots, p_r > 0\}$. We denote by $\mathfrak{S} = [F, N_\geq, P_>, H_{\neq}]$ the semi-algebraic system (SAS) which is the conjunction of the following conditions: $f_1 = 0, \ldots, f_s = 0$, $n_1 \geq 0, \ldots, n_t \geq 0$, $p_1 > 0, \ldots, p_r > 0$ and $h_1 \neq 0, \ldots, h_\ell \neq 0$.

Notations for zero sets. In this paper, we use "Z" to denote the zero set in $\mathbb{C}^n$ of a polynomial system, involving equations and inequations, and "$Z_\mathbb{R}$" to denote the zero set in $\mathbb{R}^n$ of a semi-algebraic system.

Pre-regular semi-algebraic system. Let $[T, P]$ be a squarefree regular system of $\mathbb{Q}[\mathbf{u}, \mathbf{y}]$. Let bp be the border polynomial of $[T, P]$. Let $B \subset \mathbb{Q}[\mathbf{u}]$ be a polynomial set such that bp divides the product of polynomials in B. We call the triple $[B_{\neq}, T, P_>]$ a *pre-regular semi-algebraic system* of $\mathbb{Q}[\mathbf{x}]$. Its zero set, written as $Z_\mathbb{R}(B_{\neq}, T, P_>)$, is the set $(u, y) \in \mathbb{R}^n$ such that $b(u) \neq 0$, $t(u, y) = 0$, $p(u, y) > 0$, for all $b \in B$, $t \in T$, $p \in P$. Lemma 1 and Theorem 1 are fundamental properties of pre-regular semi-algebraic systems.

Lemma 1. Let $\mathfrak{S}$ be a semi-algebraic system of $\mathbb{Q}[\mathbf{x}]$. Then there exists finitely many pre-regular semi-algebraic systems $[B_{i\neq}, T_i, P_{i>}]$, $i = 1 \cdots e$, s.t. $Z_\mathbb{R}(\mathfrak{S}) = \cup_{i=1}^{e} Z_\mathbb{R}(B_{i\neq}, T_i, P_{i>})$.

Proof. The semi-algebraic system $\mathfrak{S}$ decomposes into basic semi-algebraic systems, by rewriting inequality of type $n \geq 0$ as: $n > 0 \vee n = 0$. Let $[F, P_>]$ be one of those basic semi-algebraic systems. If F is empty, then the triple $[\emptyset, \varnothing, P_>]$, is a pre-regular semi-algebraic system. If F is not empty, by Proposition 1 and the specifications of Triangularize and Regularize, one can compute finitely many squarefree regular systems $[T_i, H]$ such that $V(F) \cap Z(P_{\neq}) = \cup_{i=1}^{e} \big(V(T_i) \cap Z(B_{i\neq})\big)$ holds and where B_i is the border polynomial set of the regular system $[T_i, H]$. Hence, we have $Z_\mathbb{R}(F, P_>) = \cup_{i=1}^{e} Z_\mathbb{R}(B_{i\neq}, T_i, P_>)$, where each $[B_{i\neq}, T_i, P_>]$ is a pre-regular semi-algebraic system. □

Next, we exhibit properties of pre-regular semi-algebraic systems. To this end, we recall the notion of delineability [17]. Assume $n > 1$. Let C be a connected cell in $\mathbb{R}^{n-1}$. A polynomial $p \in \mathbb{R}[x_1, \ldots, x_n]$ is *delineable* on C if the real zeros of p define continuous real-valued functions $\theta_1, \ldots, \theta_s$ such that, for all $\alpha \in C$ we have $\theta_1(\alpha) < \cdots < \theta_s(\alpha)$.

Lemma 2 (Theorem 1 in [17])**.** Let p be a polynomial of $\mathbb{R}[y_1 < \cdots < y_n]$ and C be a connected semi-algebraic subset of $\mathbb{R}^{n-1}$. If $init(p) \neq 0$ on C and the number of distinct complex roots of p is invariant on C, then p is delineable on C.

Theorem 1. Let $[B_{\neq}, T, P_>]$ be a pre-regular semi-algebraic system of $\mathbb{Q}[\mathbf{u}, \mathbf{y}]$, with T non-empty. Let h be the product of the polynomials in B. Let C be a connected subset of the complement of $h = 0$ in $\mathbb{R}^d$. Then there exist finitely many, say k, continuous semi-algebraic functions $\psi_1(\mathbf{u}), \ldots, \psi_k(\mathbf{u})$ defined on C, such that $Z_\mathbb{R}([T, P_>]) = \uplus_{i=1}^{k} \{(\alpha, \psi_i(\alpha)) \mid \alpha \in C\}$ holds, where $\uplus$ denotes a disjoint union. In particular, for each $\alpha \in C$, we have $Z_\mathbb{R}([T(\alpha), P_>(\alpha)]) = \{\psi_1(\alpha), \ldots, \psi_k(\alpha)\}$, which is a set of k points.

Proof. We prove by induction on m, the number of variables in $\mathbf{y} = y_1 < \cdots < y_m$. For $1 \le i \le m$, let $P_i = \{p \in P \mid \mathrm{mvar}(p) \le y_i\}$. Write $T = \{t_1, \ldots, t_m\}$, where polynomials are sorted by main variables.

Case $m = 1$. For any $\alpha \in C$, the regular system $[\{t_1\}, P_1]$ specializes well at α by Proposition 1, which implies that $\mathrm{init}(t_1)(\alpha) \neq 0$ and $t_1(\alpha, y_1)$ is a squarefree polynomial in $\mathbb{R}[y_1]$. Therefore, the polynomial t_1 is delineable on C by Lemma 2, which implies that the real zero set of t_1 over C consists of finitely many (possibly none) disjoint graphs of continuous functions. Let $\psi_1(\mathbf{u}), \ldots, \psi_{k'}(\mathbf{u})$ be these functions. For $i = 1, \ldots, k'$, the graph of ψ_i over C, denoted by G_i, is a connected semi-algebraic set. Moreover, since $[\{t_1\}, P_1]$ specializes well above C, we deduce that the sign of each $p \in P_1$ does not change above G_i. We pick those ψ_i such that $G_i \cap Z_{\mathbb{R}}(P_{1>}) \neq \emptyset$ holds and renumber them as $\psi_1(\mathbf{u}), \ldots, \psi_k(\mathbf{u})$. Clearly we have $Z_{\mathbb{R}}([t_1, P_{1>}]) = \uplus_{i=1}^{k}\{(\alpha, \psi_i(\alpha)) \mid \alpha \in C)\}$ holds.

Case $m > 1$. Assume that the conclusion holds for the pre-regular semi-algebraic system $[B_{\neq}, \{t_1, \ldots, t_{m-1}\}, P_{m-1>}]$, that is, there exist k continuous semi-algebraic functions $\psi_1(\mathbf{u}), \ldots, \psi_k(\mathbf{u})$ defined on C such that

$$Z_{\mathbb{R}}([\{t_1, \ldots, t_{m-1}\}, P_{m-1>}]) = \uplus_{i=1}^{k}\{(\alpha, \psi_i(\alpha)) \mid \alpha \in C\}$$

holds. For $i = 1, \ldots, k$, let $G_i := \{(\alpha, \psi_i(\alpha)) \mid \alpha \in C\}$. Then each G_i is a connected semi-algebraic set. Moreover, by Proposition 1, $[T, P]$ specializes well above $Z_{\mathbb{R}}(B_{\neq})$, which implies that $[\{t_m\}, P_m]$ specializes well above G_i. By similar arguments as in the proof of the case $m = 1$, we deduce that for each $i = 1, \ldots, k$, there exists $n_i \ge 0$ continuous semi-algebraic functions $\psi_{i,1}(\mathbf{u}, y_1, \ldots, y_{m-1}), \ldots, \psi_{i,n_i}(\mathbf{u}, y_1, \ldots, y_{m-1})$ defined on G_i such that $\{(\gamma, \beta) \in \mathbb{R}^{d+m-1} \times \mathbb{R} \mid \gamma \in G_i, t_m(\gamma, \beta) = 0, p(\gamma, \beta) > 0 \text{ for all } p \in P_m\}$ equals to $\uplus_{j=1}^{n_i}\{(\gamma, \psi_{i,j}(\gamma)) \mid \gamma \in G_i\}$, which implies that

$$Z_{\mathbb{R}}([T, P_>]) = \uplus_{i=1}^{k}\uplus_{j=1}^{n_i}\{(\alpha, \psi_i(\alpha), \psi_{i,j}(\alpha, \psi_i(\alpha))) \mid \alpha \in C\}$$

holds. Clearly $(\psi_i(\mathbf{u}), \psi_{i,j}(\mathbf{u}, \psi_i(\mathbf{u})))$, where $i = 1, \ldots, k$, $j = 1, \ldots, n_i$, are continuous semi-algebraic functions defined on C, so the conclusion holds. □

Lemma 3. Let $[B_{\neq}, T, P_>]$ be a pre-regular semi-algebraic system of $\mathbb{Q}[\mathbf{u}, \mathbf{y}]$. One can decide whether its zero set is empty or not. If it is not empty, then one can compute a regular semi-algebraic system $[\mathcal{Q}, T, P_>]$ whose zero set is the same as that of $[B_{\neq}, T, P_>]$.

Proof. If $T = \varnothing$, we can test whether the zero set of $[B_{\neq}, P_>]$ is empty or not, for instance using CAD. If it is empty, we are done. Otherwise, defining $\mathcal{Q} = B_{\neq} \wedge P_>$, $[\mathcal{Q}, T, P_>]$ is a regular semi-algebraic system whose zero set equals that of $[B_{\neq}, T, P_>]$. If T is not empty, we solve the quantifier elimination problem $\exists \mathbf{y}(B(\mathbf{u}) \neq 0, T(\mathbf{u}, \mathbf{y}) = 0, P(\mathbf{u}, \mathbf{y}) > 0)$ and let $\mathcal{Q}$ be the resulting formula. By Theorem 1, above each connected component of $B(\mathbf{u}) \neq 0$, the number of real zeros of the system $[B_{\neq}, T, P_>]$ is constant. Hence, we claim that the zero set defined by $\mathcal{Q}$ is the union of the connected components of $B(\mathbf{u}) \neq 0$ above which $[B_{\neq}, T, P_>]$ possesses at least one solution. If $\mathcal{Q}$ is false, we are done. Otherwise, $\mathcal{Q}$ defines a nonempty open set of $\mathbb{R}^d$ and $[\mathcal{Q}, T, P_>]$ is a regular semi-algebraic system whose zero set equals that of $[B_{\neq}, T, P_>]$. □

Theorem 2. Let $\mathfrak{S}$ be a semi-algebraic system of $\mathbb{Q}[\mathbf{x}]$. Then one can compute a (full) triangular decomposition of $\mathfrak{S}$, that is, as defined in the introduction, finitely many regular semi-algebraic systems such that the union of their zero sets is the zero set of $\mathfrak{S}$.

Proof. This follows from Lemma 1 and 3. □

4. Complexity results for computing a lazy triangular decomposition: a theoretical perspective

We prove that, under some genericity assumptions, a lazy triangular decomposition of a polynomial system is computed in singly exponential time w.r.t. the number of variables. First, we state complexity estimates for basic multivariate polynomial operations.

Complexity of basic polynomial operations. Let $p, q \in \mathbb{Q}[\mathbf{x}]$ be polynomials with respective total degrees δ_p, δ_q, and let $x \in \mathbf{x}$. Let $\hbar_p, \hbar_q, \hbar_{pq}$ and $\hbar_r$ be the *height* (that is, the bit size of the maximum absolute value of the numerator or denominator of a coefficient) of p, q, the product pq and the resultant $\mathrm{res}(p, q, x)$, respectively; let $\delta := \max(\delta_p, \delta_q)$ and $\hbar := \max(\hbar_p, \hbar_q)$. In [22], it is proved that $\gcd(p, q)$ can be computed within $O(n^{2\delta+1}\hbar^3)$ bit operations. It is easy to establish that $\hbar_{pq}$ and $\hbar_r$ are respectively upper bounded by $\hbar_p + \hbar_q + n\log(\min(\delta_p, \delta_q) + 1)$ and $\delta_q\hbar_p + \delta_p\hbar_q + n\delta_q\log(\delta_p + 1) + n\delta_p\log(\delta_q + 1) + \log((\delta_p + \delta_q)!)$. Finally, according to [24], the bit operations of p pseudo-dividing q w.r.t. x is $O((\delta + 1)^{3n}\hbar^2)$; let M be a $k \times k$ matrix over $\mathbb{Q}[\mathbf{x}]$, δ (resp. $\hbar$) be the maximum total degree (resp. height) of an element of M, then $\det(M)$ can be computed within $O(k^{2n+5}(\delta + 1)^{2n}\hbar^2)$ bit operations.

We turn now to the main subject of this section, that is, complexity estimates for a lazy triangular decomposition of a polynomial system under some genericity assumptions. Let $F \subset \mathbb{Q}[\mathbf{x}]$. A lazy triangular decomposition (defined in the Introduction) of the semi-algebraic system $\mathfrak{S} = [F, \emptyset, \emptyset, \emptyset]$, involving only equations, is obtained by Algorithm 1.

Algorithm 1: LazyRealTriangularize($\mathfrak{S}$)

```
Input: a semi-algebraic system $\mathfrak{S} = [F, \emptyset, \emptyset, \emptyset]$
Output: a lazy triangular decomposition of $\mathfrak{S}$
$\mathfrak{T}$ := Triangularize($F$)
for $T_i \in \mathfrak{T}$ do
    $bp_i$ := BorderPolynomial($T_i, \emptyset$)
    solve $\exists\mathbf{y}(bp_i(\mathbf{u}) \neq 0, T_i(\mathbf{u}, \mathbf{y}) = 0)$; let $\mathcal{Q}_i$ be the resulting quantifier-free formula
    if $\mathcal{Q}_i \neq false$ then output $[\mathcal{Q}_i, T_i, \emptyset]$
```

Proof of Algorithm 1. The termination of the algorithm is obvious. Let us prove its correctness. Let $R_i = [\mathcal{Q}_i, T_i, \emptyset]$, for $i = 1 \cdots t$ be the output of Algorithm 1 and let T_j for $j = t + 1 \cdots s$ be the regular chains such that $\mathcal{Q}_j = false$. By Lemma 3, each R_i is a regular semi-algebraic system. For $i = 1 \cdots s$, define $F_i = \mathrm{sat}(T_i)$. Then we have $V(F) = \cup_{i=1}^{s} V(F_i)$, where each F_i is equidimensional. For each $i = 1 \cdots s$, by Proposition 1, we have $V(F_i) \setminus V(bp_i) = V(T_i) \setminus V(bp_i)$. Moreover, we have $V(F_i) = (V(F_i) \setminus V(bp_i)) \cup V(F_i \cup \{bp_i\})$. Hence, $Z_{\mathbb{R}}(R_i) = Z_{\mathbb{R}}(T_i) \setminus Z_{\mathbb{R}}(bp_i) \subseteq Z_{\mathbb{R}}(F_i) \subseteq Z_{\mathbb{R}}(F)$ holds. In addition, since bp_i is regular modulo F_i, we have

$$\begin{aligned} Z_{\mathbb{R}}(F) \setminus \cup_{i=1}^{t} Z_{\mathbb{R}}(R_i) &= \cup_{i=1}^{s} Z_{\mathbb{R}}(F_i) \setminus \cup_{i=1}^{t} Z_{\mathbb{R}}(R_i) \\ &\subseteq \cup_{i=1}^{s} Z_{\mathbb{R}}(F_i) \setminus (Z_{\mathbb{R}}(T_i) \setminus Z_{\mathbb{R}}(bp_i)) \\ &\subseteq \cup_{i=1}^{s} Z_{\mathbb{R}}(F_i \cup \{bp_i\}), \end{aligned}$$

and $\dim(\cup_{i=1}^{s} V(F_i \cup \{bp_i\})) < \dim(V(F))$. So the R_i, for $i = 1 \cdots t$, form a lazy triangular decomposition of $\mathfrak{S}$. □

In this section, under some genericity assumptions for F, we establish running time estimates for Algorithm 1, see Theorem 4. This is achieved through Proposition 2 (which gives running time and output size estimates for a Kalkbrener triangular decomposition of an algebraic set) and Theorem 3 (which states running time and output size estimates for a border polynomial computation). Our assumptions for these results are the following:

($\mathbf{H_0}$) $V(F)$ is equidimensional of dimension d,

($\mathbf{H_1}$) $x_1, \ldots, x_d$ are algebraically independent modulo each associated prime ideal of the ideal generated by F in $\mathbb{Q}[\mathbf{x}]$,

($\mathbf{H_2}$) F consists of $m := n - d$ polynomials, $f_1, \ldots, f_m$.

Hypotheses ($\mathbf{H_0}$) and ($\mathbf{H_1}$) are equivalent to the existence of regular chains $T_1, \ldots, T_e$ of $\mathbb{Q}[x_1, \ldots, x_n]$ such that $x_1, \ldots, x_d$ are free w.r.t. each of $T_1, \ldots, T_e$ and such that we have $V(F) = \overline{W(T_1)} \cup \cdots \cup \overline{W(T_e)}$.

Denote by δ, $\hbar$ respectively the maximum total degree and height of $f_1, \ldots, f_m$. In her PhD Thesis [36], Á. Szántó describes an algorithm which computes a Kalkbrener triangular decomposition, $T_1, \ldots, T_e$, of $V(F)$. Under hypotheses ($\mathbf{H_0}$) to ($\mathbf{H_2}$), this algorithm runs in time $m^{O(1)}(\delta^{O(n^2)})^{d+1}$ counting operations in $\mathbb{Q}$, while the total degrees of the polynomials in the output are bounded by $n\delta^{O(m^2)}$. In addition, $T_1, \ldots, T_e$ are square free, *strongly normalized* [31] and *reduced* [1].

From $T_1, \ldots, T_e$, we obtain regular chains $E_1, \ldots, E_e$ forming another Kalkbrener triangular decomposition of $V(F)$, as follows. Let $i = 1 \cdots e$ and $j = (d+1) \cdots n$. Let $t_{i,j}$ be the polynomial of T_i with x_j as main variable. Let $e_{i,j}$ be the primitive part of $t_{i,j}$ regarded as a polynomial in $\mathbb{Q}[x_1, \ldots, x_d][x_{d+1}, \ldots, x_n]$. Define $E_i = \{e_{i,d+1}, \ldots, e_{i,n}\}$. According to the complexity results for polynomial operations stated at the beginning of this section, this transformation can be done within $\delta^{O(m^4)O(n)}$ operations in $\mathbb{Q}$.

Dividing $e_{i,j}$ by its initial we obtain a monic polynomial $d_{i,j}$ of the polynomial ring $\mathbb{Q}(x_1, \ldots, x_d)[x_{d+1}, \ldots, x_n]$. Denote by D_i the regular chain $\{d_{i,d+1}, \ldots, d_{i,n}\}$. Observe that D_i is the reduced lexicographic Gröbner basis of the radical ideal it generates in $\mathbb{Q}(x_1, \ldots, x_d)[x_{d+1}, \ldots, x_n]$. So Theorem 1 in [20] applies to each regular chain D_i. For each polynomial $d_{i,j}$, this theorem provides height and total degree estimates expressed as functions of the *degree* [9] and the *height* [32, 27] of the algebraic set $\overline{W(D_i)}$. Note that the degree and height of $\overline{W(D_i)}$ are upper bounded by those of $V(F)$. Write $d_{i,j} = \Sigma_\mu \frac{\alpha_\mu}{\beta_\mu} \mu$ where each $\mu \in \mathbb{Q}[x_{d+1}, \ldots, x_n]$ is a monomial and α_μ, β_μ are in $\mathbb{Q}[x_1, \ldots, x_d]$ such that $\gcd(\alpha_\mu, \beta_\mu) = 1$ holds. Let γ be the lcm of the β_μ's. Then for γ and each α_μ:

- the total degree is bounded by $2\delta^{2m}$ and,
- the height by $O(\delta^{2m}(m\hbar + dm\log(\delta) + n\log(n)))$.

Multiplying $d_{i,j}$ by γ brings $e_{i,j}$ back. We deduce the height and total degree estimates for each $e_{i,j}$ below.

Proposition 2. Under the hypotheses ($\mathbf{H_0}$), ($\mathbf{H_1}$), ($\mathbf{H_2}$), the Kalkbrener triangular decomposition $E_1, \ldots, E_e$ of $V(F)$ can be computed in $\delta^{O(m^4)O(n)}$ operations in $\mathbb{Q}$. In addition, every polynomial $e_{i,j}$ has total degree upper bounded by $4\delta^{2m} + \delta^m$, and has height upper bounded by $O(\delta^{2m}(m\hbar + dm\log(\delta) + n\log(n)))$.

Next we estimate running time and output size for a border polynomial computation.

Theorem 3. Let $R = [T, P]$ be a squarefree regular system of $\mathbb{Q}[\mathbf{u}, \mathbf{y}]$, with $m = \#T$ and $\ell = \#P$. Let bp be the border polynomial of R. Denote by δ_R, $\hbar_R$ respectively the maximum total degree and height of a polynomial in R. Then the total degree of bp is upper bounded by $(\ell+m)2^{m-1}{\delta_R}^m$, and bp can be computed within $(n\ell+nm)^{O(n)}(2\delta_R)^{O(n)O(m)}{\hbar_R}^3$ bit operations.

Proof. Define $G := P \cup \{\mathrm{der}(t) \mid t \in T\}$. We need to compute the $\ell+m$ iterated resultants $\mathrm{res}(g,T)$, for all $g \in G$. Let $g \in G$. Observe that the total degree and height of g are bounded by δ_R and $\hbar_R + \log(\delta_R)$ respectively. Define $r_{m+1} := g, \ldots, r_i := \mathrm{res}(t_i, r_{i+1}, y_i), \ldots, r_1 := \mathrm{res}(t_1, r_2, y_1)$. Let $i \in \{1, \ldots, m\}$. Denote by δ_i and $\hbar_i$ the total degree and height of r_i, respectively. Using the complexity estimates stated at the beginning of this section, we have $\delta_i \le 2^{m-i+1}{\delta_R}^{m-i+2}$ and $\hbar_i \le 2\delta_{i+1}(\hbar_{i+1}+n\log(\delta_{i+1}+1))$. Therefore, we have $\hbar_i \le (2\delta_R)^{O(m^2)}n^{O(m)}\hbar_R$. From these size estimates, one can deduce that each resultant r_i (thus the iterated resultants) can be computed within $(2\delta_R)^{O(mn)+O(m^2)}n^{O(m)}{\hbar_R}^2$ bit operations, by the complexity of computing a determinant stated at the beginning of this section. Hence, the product of all iterated resultants has total degree and height bounded by $(\ell+m)2^{m-1}{\delta_R}^m$ and $(\ell+m)(2\delta_R)^{O(m^2)}n^{O(m)}\hbar_R$, respectively. Thus, the primitive and squarefree part of this product can be computed within $(n\ell+nm)^{O(n)}(2\delta_R)^{O(n)O(m)}{\hbar_R}^3$ bit operations, based on the complexity of a polynomial gcd computation stated at the beginning of this section. □

Theorem 4. From the Kalkbrener triangular decomposition $E_1, \ldots, E_e$ of Proposition 2, a lazy triangular decomposition of $f_1 = \cdots = f_m = 0$ can be computed in $\left(\delta^{n^2}n4^n\right)^{O(n^2)}\hbar^{O(1)}$ bit operations. Thus, under the hypotheses $(\mathbf{H_0})$, $(\mathbf{H_1})$ and $(\mathbf{H_2})$, a lazy triangular decomposition of this system is computed from the input polynomials in singly exponential time w.r.t. n, counting operations in $\mathbb{Q}$.

Proof. For each $i \in \{1 \cdots e\}$, let bp_i be the border polynomial of $[E_i, \emptyset]$ and let $\hbar_{R_i}$ (resp. δ_{R_i}) be the height (resp. the total degree) bound of the polynomials in the pre-regular semi-algebraic system $R_i = [\{bp_i\}_{\neq}, E_i, \emptyset]$. According to Algorithm 1, the remaining task is to solve the QE problem $\exists \mathbf{y}(bp_i(\mathbf{u}) \neq 0, E_i(\mathbf{u},\mathbf{y}) = 0)$ for each $i \in \{1 \cdots e\}$, which can be solved within $((m+1)\delta_{R_i})^{O(dm)}\,\hbar_{R_i}^{O(1)}$ bit operations, based on the results of [33]. The conclusion follows from the size estimates in Proposition 2 and Theorem 3. □

5. Quantifier elimination via real root classification

In Section 4, we saw that in order to compute a triangular decomposition of a semi-algebraic system, a key step was to solve the following quantifier elimination problem:

$$\exists \mathbf{y}(B(\mathbf{u}) \neq 0, T(\mathbf{u},\mathbf{y}) = 0, P(\mathbf{u},\mathbf{y}) > 0), \tag{1}$$

where $[B_{\neq}, T, P_{>}]$ is a pre-regular semi-algebraic system of $\mathbb{Q}[\mathbf{u},\mathbf{y}]$. This problem is an instance of the so-called *real root classification* (RRC) [43]. In this section, we show how to solve this problem when B is what we call a *fingerprint polynomial set.*

Definition 3. Let $R := [B_{\neq}, T, P_{>}]$ be a pre-regular semi-algebraic system of $\mathbb{Q}[\mathbf{u},\mathbf{y}]$. Let $D \subset \mathbb{Q}[\mathbf{u}]$. Let dp be the product of all polynomials in D. We call D a *fingerprint polynomial set* (FPS) of R if:

(i) for all $\alpha \in \mathbb{R}^d$, for all $b \in B$ we have: $dp(\alpha) \neq 0 \Longrightarrow b(\alpha) \neq 0$,

(ii) for all $\alpha, \beta \in \mathbb{R}^d$ with $\alpha \neq \beta$, $dp(\alpha) \neq 0$ and $dp(\beta) \neq 0$: if $p(\alpha)$ and $p(\beta)$ have the same sign for all $p \in D$, then $R(\alpha)$ has real solutions if and only if $R(\beta)$ does.

Now, we present a method for constructing an FPS based on CAD projection operators.

Open projection operator [35, 5]. Hereafter in this section, we let $\mathbf{u} = u_1 < \cdots < u_d$ be ordered variables. Let $p \in \mathbb{Q}[\mathbf{u}]$ be non-constant. We denote by $\mathsf{factor}(p)$ the set of the non-constant irreducible factors of p. For $A \subset \mathbb{Q}[\mathbf{u}]$, we define $\mathsf{factor}(A) = \cup_{p \in A} \mathsf{factor}(p)$. Let C_d (resp. C_0) be the set of the polynomials in $\mathsf{factor}(p)$ with main variable equal to (resp. less than) u_d. The *open projection operator* (oproj) w.r.t. variable u_d maps p to a set of polynomials of $\mathbb{Q}[u_1, \ldots, u_{d-1}]$ defined below:

$$\begin{aligned} \mathrm{oproj}(p, u_d) := C_0 \cup \bigcup\nolimits_{f,g \in C_d,\, f \neq g} & \mathsf{factor}(\mathrm{res}(f, g, u_d)) \\ \cup \bigcup\nolimits_{f \in C_d} & \mathsf{factor}(\mathrm{init}(f, u_d) \cdot \mathrm{discrim}(f, u_d)). \end{aligned}$$

Then, we define: $\mathrm{oproj}(A, u_d) := \mathrm{oproj}(\Pi_{p \in A}\, p, u_d)$.

Augmentation. Let $A \subset \mathbb{Q}[\mathbf{u}]$ and $x \in \{u_1, \ldots, u_d\}$. Denote by $\mathrm{der}(A, x)$ the *derivative closure* of A w.r.t. x, that is, $\mathrm{der}(A, x) := \cup_{p \in A} \{\mathrm{der}^{(i)}(p, x) \mid 0 \leq i < \deg(p, x)\}$. The *open augmented projected factors* of A is denoted by $\mathrm{oaf}(A)$ and defined as follows. Let k be the smallest positive integer such that $A \subset \mathbb{Q}[u_1, \ldots, u_k]$ holds. Denote by C the set $\mathsf{factor}(\mathrm{der}(A, u_k))$; we have

- if $k = 1$, then $\mathrm{oaf}(A) := C$;
- if $k > 1$, then $\mathrm{oaf}(A) := C \cup \mathrm{oaf}(\mathrm{oproj}(C, u_k))$.

Proposition 3. Let $A \subset \mathbb{Q}[\mathbf{u}]$ be finite and let σ be an arbitrary map from $\mathrm{oaf}(A)$ to the set of signs $\{-1, +1\}$. We define:

$$S_d := \bigcap\nolimits_{p \in \mathrm{oaf}(A)} \{u \in \mathbb{R}^d \mid p(u)\, \sigma(p) > 0\}.$$

Then the set S_d is either empty or a connected open set in $\mathbb{R}^d$.

Proof. By induction on d. When $d = 1$, the conclusion follows from Thom's Lemma [2]. Assume $d > 1$. If d is not the smallest positive integer k such that $A \subset \mathbb{Q}[u_1, \ldots, u_k]$ holds, then S_d writes $S_{d-1} \times \mathbb{R}$ and the conclusion follows by induction. Otherwise, write $\mathrm{oaf}(A)$ as $C \cup E$, where $C = \mathsf{factor}(\mathrm{der}(A, u_d))$ and $E = \mathrm{oaf}(\mathrm{oproj}(C, u_d))$. We have: $E \subset \mathbb{Q}[u_1, \ldots, u_{d-1}]$. Let $\mathrm{M} = \cap_{p \in E} \{u \in \mathbb{R}^{d-1} \mid p(u)\sigma(p) > 0\}$. If M is empty then so is S_d and the conclusion is clear. From now on assume M not empty. Then, by induction hypothesis, M is a connected open set in $\mathbb{R}^{d-1}$. By the definition of the operator oproj and Lemma 2, the product of the polynomials in C is delineable over M w.r.t. u_d. Moreover, C is derivative closed (may be empty) w.r.t. u_d. Therefore $\cap_{p \in \mathrm{oaf}(A)} \{u \in \mathbb{R}^d \mid p(u)\, \sigma(p) > 0\} \subset M \times \mathbb{R}$ is either empty or a connected open set by Thom's Lemma. □

Theorem 5. Let $R := [B_{\neq}, T, P_{>}]$ be a pre-regular semi-algebraic system of $\mathbb{Q}[\mathbf{u}, \mathbf{y}]$. The polynomial set $\mathrm{oaf}(B)$ is a fingerprint polynomial set of R.

Proof. Recall that the border polynomial bp of $[T, P]$ divides the product of the polynomials in B. We have $\mathsf{factor}(B) \subseteq \mathrm{oaf}(B)$. So $\mathrm{oaf}(B)$ clearly satisfies (i) in Definition 3. Let us prove (ii). Let dp be the product of the polynomials in $\mathrm{oaf}(B)$. Let $\alpha, \beta \in \mathbb{R}^d$ such that both $dp(\alpha) \neq 0$, $dp(\beta) \neq 0$ hold and the signs of $p(\alpha)$ and $p(\beta)$ are equal for all $p \in \mathrm{oaf}(B)$. Then, by Proposition 3, α and β belong to the same connected component of $dp(\mathbf{u}) \neq 0$, and thus to the same connected component of $B(\mathbf{u}) \neq 0$. Therefore the number of real solutions of $R(\alpha)$ and that of $R(\beta)$ are the same by Theorem 1. □

From now on, let us assume that the set B in the pre-regular semi-algebraic system $R = [B_{\neq}, T, P_{>}]$ is an FPS of R. We solve the quantifier elimination problem (1) in three steps: (s_1) compute at least one sample point in each connected component of the semi-algebraic set defined by $B(\mathbf{u}) \neq 0$; (s_2) for each sample point α such that the specialized system $R(\alpha)$ possesses real solutions, compute the sign of $b(\alpha)$ for each $b \in B$; (s_3) generate the corresponding quantifier-free formulas.

In practice, when the set B is not an FPS, one adds some polynomials from $\mathrm{oaf}(B)$, using a heuristic procedure (for instance one by one) until Property *(ii)* of the definition of an FPS is satisfied. This strategy is implemented in Algorithm 3 of Section 7.

6. Complexity results for computing a fingerprint polynomial set: a practical perspective

Let $R := [B_{\neq}, T, P_{>}]$ be a pre-regular semi-algebraic system of $\mathbb{Q}[\mathbf{u}, \mathbf{y}]$, where $\mathbf{u}$ stands for the free variables of T and $\mathbf{y} = y_1 < \cdots < y_m$ are the main variables of T. We write $P = \{p_1, \ldots, p_\ell\}$ and $T = \{t_1, \ldots, t_m\}$. In this section, we always assume that T is in *generic position*, that is, the main degree of t_i is 1 for $1 < i \leq m$. Under such an assumption, we show that a fingerprint polynomial set of R can be computed in singly exponential time w.r.t. the number of variables. Note that the construction in Section 5 is doubly exponential [7]. Since a regular chain is often in generic position and detecting this shape is easy, this new construction leads to a practical and more effective way for computing fingerprint polynomial set, which has been integrated in our tools.

To achieve this, we present an alternative way (w.r.t. the one presented in last section) to construct a fingerprint polynomial set of R. This new method relies on a tool called *generalized discriminant sequence* (GDS) for counting the number of real solutions of a univariate polynomial with parametric coefficients, which we review as follows.

Definition 4 ([41, 42])**.** Let $p, q \in \mathbb{R}[x]$. We denote by p' the derivative of p w.r.t. x. Let $r := \mathrm{rem}(p'q, p, x)$ be the Euclidean remainder of $p'q$ divided by p. Let $s := \deg(p, x)$ and write $p = a_0x^s + \cdots + a_s$, $r = c_0x^{s-1} + \cdots + c_{s-1}$. The following $2s \times 2s$ matrix

$$(m_{ij}) = \begin{pmatrix} a_0 & a_1 & a_2 & \cdots & a_s & & & \\ 0 & c_0 & c_1 & \cdots & c_{s-1} & & & \\ & & \ddots & \ddots & \ddots & & & \\ & & a_0 & a_1 & a_2 & \cdots & a_s & \\ & & 0 & c_0 & c_1 & \cdots & c_{s-1} & \end{pmatrix}$$

is called the *generalized discrimination matrix* of p w.r.t. q. For $i = 1 \cdots s$, we denote by $\mathrm{gds}_i(p, q, x)$, the $2i$-th leading principal minor of the above matrix and call $\mathrm{gds}_1(p, q, x), \ldots, \mathrm{gds}_s(p, q, x)$ the *generalized discriminant sequence* of p w.r.t. q, denoted by $\mathrm{gds}(p, q, x)$. We write $\{\mathrm{gds}(p, q, x)\}$ the set consisting of the elements of $\mathrm{gds}(p, q, x)$.

Notation 1. Let p and q be two polynomials in $\mathbb{R}[x]$. Denote by $\mathrm{TaQ}(p, q)$ the number $\#\{x \mid p = 0,\ q > 0\} - \#\{x \mid p = 0,\ q < 0\}$, the *Tarski query* [2] of p w.r.t. q.

Remark 1. The elements in the generalized discriminant sequence of p w.r.t. q are in one-to-one correspondence (up to a power of a_0 and a power of -1) with the *signed subresultant coefficients* [2] of p and r. One can compute $\mathrm{TaQ}(p, q)$ merely from the signs of the elements in $\mathrm{gds}(p, q, x)$, see Theorem 4.32 in [2] or Theorem 3.2.1 in [42]: given two pairs of polynomials (p_i, q_i) $(i = 1, 2)$ with $\deg(p_1) = \deg(p_2) = s$, if $\mathrm{sign}(\mathrm{gds}_j(p_1, q_1, x)) = \mathrm{sign}(\mathrm{gds}_j(p_2, q_2, x))$ holds for all $j = 1, \ldots, s$, then $\mathrm{TaQ}(p_1, q_1) = \mathrm{TaQ}(p_2, q_2)$ holds.

We prove Lemmas 4 and 5 for completness; similar results appear in [41, 42]. Let $k > 0$ be an integer. Let p and $q_1, q_2, \ldots, q_k$ be polynomials from $\mathbb{R}[x]$ with $\gcd(p, q_j) = 1$ for each $j = 1, \ldots, k$. Lemma 4 shows that in this case, the numbers $\#\{x \mid p = 0,\, q_1\sigma_1 0, \ldots, q_k\sigma_k 0\}$ with $\sigma_1, \ldots, \sigma_k \in \{>, <\}$ can be computed from the numbers in $\{\mathrm{TaQ}(p, \prod_{j=1}^{k} q_j^{e_j}) \mid e_1, \ldots, e_k \in \{0, 1\}\}$ by solving a linear system with fixed coefficients.

Denote $\mathbf{M} := \left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right)$ and let $\mathbf{M}_1 := \mathbf{M}$. For $i = 1, \ldots, k-1$, denote by $\mathbf{M}_{i+1}$ the $2^i \times 2^i$ matrix obtained by replacing each element e of $\mathbf{M}$ with $e\mathbf{M}_i$. It is easy to deduce that $\det(\mathbf{M}_{i+1}) = 2^{2^i} \det(\mathbf{M}_i)^2$ from its block structure, which implies that all $\mathbf{M}_i$ $(i = 1, \ldots, k)$ are nonsingular.

Denote by $\mathbf{S}_1$ the list of constraints $[q_1 > 0, q_1 < 0]$, by $\mathbf{P}_1$ the polynomial list $[1, q_1]$. For $i = 1, \ldots, k-1$, denote by $\mathbf{S}_{i+1}$ the list of constraints

$$[\mathbf{S}_i[1] \wedge q_{i+1} > 0, \ldots, \mathbf{S}_i[2^i] \wedge q_{i+1} > 0, \mathbf{S}_i[1] \wedge q_{i+1} < 0, \ldots, \mathbf{S}_i[2^i] \wedge q_{i+1} < 0],$$

by $\mathbf{P}_{i+1}$ the polynomial list $[\mathbf{P}_i[1], \ldots, \mathbf{P}_i[2^i], \mathbf{P}_i[1] \cdot q_{i+1}, \ldots, \mathbf{P}_i[2^i] \cdot q_{i+1}]$. It is easy to deduce that $\mathbf{S}_i$ and $\mathbf{P}_i$ are of length 2^i.

Let $\mathbf{T}_k$ be $[\mathrm{TaQ}(p, \mathbf{P}_k[1]), \mathrm{TaQ}(p, \mathbf{P}_k[2]), \ldots, \mathrm{TaQ}(p, \mathbf{P}_k[2^k])]$. Let $\mathbf{N}_k$ be $[\#\{x \mid p = 0,\, \mathbf{S}_k[1]\}, \#\{x \mid p = 0,\, \mathbf{S}_k[2]\}, \ldots, \#\{x \mid p = 0,\, \mathbf{S}_k[2^k]\}]$. We observe that each of $\mathbf{T}_k$ and $\mathbf{N}_k$ is a list of 2^k non-negative integers.

Lemma 4 ([42])**.** Using the above notations $\mathbf{M}_k$, $\mathbf{T}_k$, $\mathbf{N}_k$ and viewing $\mathbf{T}_k$ and $\mathbf{N}_k$ as vectors, we have $\mathbf{N}_k = \mathbf{M}_k^{-1} \times \mathbf{T}_k$.

Proof. Consider the system of linear equations $\mathbf{M}_k \times \mathbf{X} = \mathbf{T}_k$ with $\mathbf{X}$ as unknown vector, one can verify that $\mathbf{X} = \mathbf{N}_k$ is the solution. Here, we only verify the base case, namely $k = 1$. Since $\gcd(p, q_1) = 1$, we have

$$\#\{x \mid p = 0,\, q_1 > 0\} + \#\{x \mid p = 0,\, q_1 < 0\} = \#\{x \mid p = 0\} = \mathrm{TaQ}(p, 1).$$

Moreover $\#\{x \mid p = 0,\, q_1 > 0\} - \#\{x \mid p = 0,\, q_1 < 0\}$ equals $\mathrm{TaQ}(p, q_1)$ by definition. □

Let p and q be two univariate polynomials of x with coefficients in $\mathbb{Q}[\mathbf{u}]$. The *signed pseudo-remainder* (see [2]) of p divided by q, denoted by $\mathrm{sPrem}(p, q, x)$, is the polynomial r satisfying $\mathrm{lc}(q)^e p = aq + r$, where $\deg(r, x) < \deg(q, x)$ and e is the smallest non-negative even integer greater than or equal to $\deg(p, x) - \deg(q, x) + 1$. In Definition 4, we reviewed the concepts of "generalized discriminant matrix (sequence)" of two univariate polynomials with real coefficients. We extend the definition to cover the case of two univariate polynomials p and q with coefficients in $\mathbb{Q}[\mathbf{u}]$ by replacing $r := \mathrm{rem}(p'q, p, x)$ with $r := \mathrm{sPrem}(p'q, p, x)$.

Lemma 5. Let p and q be two polynomials of x with coefficients in $\mathbb{Q}[\mathbf{u}]$. Let $p = a_0 x^s + \cdots + a_s$, where $a_0 \neq 0$. Suppose α_1 and α_2 are two points of $\mathbb{R}^d$ such that both $a_0(\alpha_1) \neq 0$ and $a_0(\alpha_2) \neq 0$ hold. If $\mathrm{sign}(\mathrm{gds}_j(p, q, x)(\alpha_1)) = \mathrm{sign}(\mathrm{gds}_j(p, q, x)(\alpha_2))$ hold for all $j = 1, \ldots, s$, then we have $\mathrm{TaQ}(p(\alpha_1), q(\alpha_1)) = \mathrm{TaQ}(p(\alpha_2), q(\alpha_2))$ holds.

Proof. Let $r := \mathrm{sPrem}(p'q, p, x)$. Then there exists a non-negative even integer e and a polynomial b such that $a_0^e p'q = bp + r$ holds. Therefore for any $\alpha \in \mathbb{R}^d$ such that $a_0(\alpha) \neq 0$, we have

$$p(\alpha)'q(\alpha) = \frac{b}{a_0^e}(\alpha)p(\alpha) + \frac{r}{a_0^e}(\alpha).$$

For $i = 1, 2$, denote $r_i := \mathrm{rem}(p(\alpha_i), q(\alpha_i))$. By the uniqueness of Euclidean reminder, we deduce that $r_i = \frac{r}{a_0^e}(\alpha_i)$ for $i = 1, 2$. By the specialization properties of computing the determinant of a polynomial matrix and the fact that e is an even number, we deduce that $\mathrm{sign}(\mathrm{gds}_j(p, q, x)(\alpha_i)) = \mathrm{sign}(\mathrm{gds}_j(p(\alpha_i), q(\alpha_i), x))$ holds for $i = 1, 2$ and $j = 1, \ldots, s$. Then the conclusion follows from (ii) of Remark 1. □

Lemma 6. Let p and $q_1, q_2, \ldots, q_k$ be polynomials of x with coefficients in $\mathbb{Q}[\mathbf{u}]$ and $\deg(p, x) = s$. Assume that p is squarefree and that p has no common factors with each of $q_1, q_2, \ldots, q_k$. Let D be the polynomial set consisting of the non-zero polynomials in $\bigcup_{e_1,\ldots,e_k \in \{0,1\}} \{\mathrm{gds}(p, \prod_{j=1}^k q_j^{e_j}, x)\}$. Suppose α_1, α_2 are two values of $\mathbf{u}$ such that $\mathrm{sign}(f(\alpha_1)) = \mathrm{sign}(f(\alpha_2)) \neq 0$ for each $f \in D$. Then the numbers $\#\{x | p(\alpha_1) = 0, q_1(\alpha_1) > 0, \ldots, q_k(\alpha_1) > 0\}$ and $\#\{x | p(\alpha_2) = 0, q_1(\alpha_2) > 0, \ldots, q_k(\alpha_2) > 0\}$ are equal.

Proof. Let q be any polynomial in $\{1, q_1, q_2, \ldots, q_k\}$. Then there exists a non-negative even integer e and polynomial b such that $\mathrm{lc}(p)^e p'q = bp + r$, where $\deg(r, x) < \deg(p, x)$. Since p is squarefree and p has no common factors with q, we deduce that $\gcd(p, r) = \gcd(p'q, p) = 1$ in $\mathbb{Q}(\mathbf{u})[x]$, which implies that $\mathrm{gds}_s(p, q) \neq 0$ and therefore belongs to D.

From $\mathrm{sign}(f(\alpha_1)) = \mathrm{sign}(f(\alpha_2)) \neq 0$ holds for each $f \in D$, we deduce

(1) According to Definition 4, $\mathrm{lc}(p)$ is a factor of each polynomial in D. Therefore, $\mathrm{lc}(p)(\alpha_i) \neq 0$ holds.

(2) For each $q \in \{q_1, \ldots, q_k\}$, we have $\mathrm{gds}_s(p, q, x)(\alpha_i) \neq 0$ holds, which implies that $\gcd(p(\alpha_i), \mathrm{sPrem}(p'q, p, x)(\alpha_i)) = 1$ by (1) and the specialization properties of computing the determinant of a polynomial matrix. So $\gcd(p(\alpha_i), p'(\alpha_i)q(\alpha_i)) = 1$, which implies that $\gcd(p(\alpha_i), q(\alpha_i)) = 1$.

(3) For all $e_1, \ldots, e_k \in \{0, 1\}$, $\mathrm{TaQ}(p(\alpha_1), \prod_{j=1}^k q_j^{e_j}(\alpha_1)) = \mathrm{TaQ}(p(\alpha_2), \prod_{j=1}^k q_j^{e_j}(\alpha_2))$ by Lemma 5.

For $i = 1, 2$, let $\mathbf{N}_{\alpha_i}$, $\mathbf{T}_{\alpha_i}$ be the $\mathbf{N}_k$ and $\mathbf{T}_k$ constructed as in Lemma 4 for the polynomials $p(\alpha_i), q_1(\alpha_i), \cdots, q_k(\alpha_i)$. Then we have $\mathbf{T}_{\alpha_1} = \mathbf{T}_{\alpha_2}$ by the above item (3). Therefore, we have $\mathbf{N}_{\alpha_1} = \mathbf{N}_{\alpha_2}$ by Lemma 4. Then the conclusion follows, since the two numbers are the first element of $\mathbf{N}_{\alpha_1}$ and $\mathbf{N}_{\alpha_2}$ respectively. □

We return to the pre-regular semi-algebraic system $[B_{\neq}, T, P_{>}]$ introduced at the beginning of this section. Recall that m and ℓ are the numbers of polynomials in T and P respectively. Let $\mathfrak{P}_{m+1} := P$ and $\mathfrak{P}_i := \{\mathrm{sPrem}(p, t_i, y_i) \mid p \in \mathfrak{P}_{i+1}\}$ for $i = m, \ldots, 2$. Note $\mathfrak{P}_i$ $(i = m+1, \ldots, 2)$ has at most ℓ elements and suppose that $\mathfrak{P}_2 = \{b_1, \ldots, b_k\}$ $(k \leq \ell)$. Let

$$\mathfrak{P}_1 := \bigcup_{(\alpha_1,\alpha_2,\ldots,\alpha_k)\in\{0,1\}^k} \{\mathrm{gds}(t_1, \prod_{i=1}^k b_i^{\alpha_i}, y_1)\} \setminus \{0\}.$$

Proposition 4. Assume that T is in generic position and let $D := B \cup \mathfrak{P}_1$. Then the set D is a fingerprint polynomial set of the pre-regular semi-algebraic system $[B_{\neq}, T, P_{>}]$.

Proof. First since the main degree of t_i, $2 \leq i \leq m$, is 1, by the relation between pseudo remainder and resultant, we conclude that $\mathfrak{P}_2$ only have variables $\mathbf{u}$ and y_1.

Let dp be the product of polynomials in D. By the definition of D, we know that the border polynomial of $[T, P]$ divides dp. By Proposition 1, for any $\alpha \in \mathbb{R}^d$ such that $dp(\alpha) \neq 0$, the regular system $[T, P]$ specializes well at α. On the other hand, by the definition of signed pseudo remainder, there exists even integers δ_i, $1 \leq i \leq \ell$, and polynomials q_{ij}, $1 \leq i \leq \ell, 2 \leq j \leq m$, such that $h_{T_{\geq y_2}}^{\delta_i} p_i = \sum_{j=2}^{m} q_{ij} t_j + b_i$ $(*)$.

Hence, for any $\beta = (\beta_1, \ldots, \beta_m)$ such that $T(\alpha, \beta) = 0$ and $P(\alpha, \beta) > 0$, we have $t_1(\alpha, \beta_1) = 0$ and $b_1(\alpha, \beta_1) > 0$. Similarly, for all β_1 such that $t_1(\alpha, \beta_1) = 0$ and $b_1(\alpha, \beta_1) > 0$, there exists a unique $\beta = (\beta_1, \ldots, \beta_m)$ with $T(\alpha, \beta) = 0$ and $P(\alpha, \beta) > 0$.

Therefore, for any $\alpha \in \mathbb{R}^d$ such that $dp(\alpha) \neq 0$, there is a 1-to-1 correspondence between the real solutions of $t_1(\alpha) = 0, \mathfrak{P}_2(\alpha) > 0$ and those of $[T(\alpha), P(\alpha)_>]$. On the other hand, since for any β such that $T(\alpha, \beta) = 0$, we have $p(\alpha, \beta) \neq 0$ for any $p \in P$, by relation $(*)$ we deduce that $t_1(\alpha)$ has no common factors with any $p(\alpha)$, where $p \in P$. The polynomial $t_1(\alpha)$ is clearly squarefree since $[T, P]$ specializes well at α. Thus it follows from Lemma 6 that the number of real solutions of $t_1 = 0, \mathfrak{P}_2 > 0$ is determined by signs of polynomials in D. Therefore, the number of real solutions of $[B_{\neq}, T, P_>]$ is also determined by signs of polynomials in D. Finally, D is an FPS of $[B_{\neq}, T, P_>]$. □

Theorem 6. Let δ and $\hbar$ be respectively the maximum total degree and the maximum coefficient size among all polynomials in P or T. Recall that ℓ and m denote the number of polynomials in P and T respectively. Then the following three properties hold:
(1) $\mathfrak{P}_1$ has at most $\delta 2^\ell$ polynomials,
(2) the total degree and, the coefficient bit-size of any polynomials in $\mathfrak{P}_1$ are upper bounded by $2\ell(\delta + 1)^{m+3}$ and $\ell^3 \delta^{\mathcal{O}(m^2)} n\hbar$ respectively;
(3) each polynomial in $\mathfrak{P}_1$ can be computed within $2^{\mathcal{O}(n)} \ell^{\mathcal{O}(n)} \delta^{\mathcal{O}(n)\mathcal{O}(m^2)} \hbar^2$ bit-operations.

Proof. Denote the total degree and coefficient bit-size of any polynomials in $\mathfrak{P}_{i+1}$ by Δ_i and $\bar{H}_i$ respectively, for $i = 2, \ldots, m$. Combining the estimates for pseudo-reminder and polynomial product recalled in Section 4, we have

$$\Delta_i \leq \delta(\Delta_{i+1} + 1) \;\text{ and }\; \bar{H}_i \leq (\Delta_{i+1} + 1)\left(\bar{H}_{i+1} + n\log(\Delta_{i+1})\right),$$

where $\Delta_{m+1} = \delta$, $\bar{H}_{m+1} = \hbar$. Therefore, for $i = 0, \ldots, m-1$, we have

$$\Delta_{m-i+1} \leq (\delta + 1)^{i+1} \;\text{ and }\; H_{m-i+1} < (\delta + 1)^{\frac{i^2 - i}{2}} \left(\hbar + i2n\log(\delta + 1)\right).$$

Thus, the total degree and coefficient size of polynomials in $\mathfrak{P}_2$ are upper bounded by $(\delta{+}1)^m$ and $(\delta{+}1)^{\frac{m^2}{2}} (\hbar + nm\log(\delta + 1))$. Applying the estimates of polynomial product, the total degrees and coefficient sizes of a product of k $(k \leq \ell)$ polynomials from $\mathfrak{P}_2$ are bounded over respectively by $\ell(\delta + 1)^m$ and $\ell^2(\delta + 1)^{\frac{m^2}{2}} \left(\hbar + mn\log(\delta + 1)\right)$. Since $\mathfrak{P}_2$ has k $(k \leq \ell)$ polynomials and $\deg(t_1) < \delta$, the set $\mathfrak{P}_1$ has at most $\delta 2^\ell$ polynomials.

Applying the estimates for the determinant of a matrix of multivariate polynomials in Section 4, each polynomial in $\mathfrak{P}_1$ has total degree and coefficient size upper bound $2\ell(\delta + 1)^{m+3}$ and $\ell^3 \delta^{\mathcal{O}(m^2)} n\hbar$ respectively, and can be computed in $2^{\mathcal{O}(n)} \ell^{\mathcal{O}(n)} \delta^{\mathcal{O}(n)\mathcal{O}(m^2)} \hbar^2$ bit operations starting from $\mathfrak{P}_2$.

Note that a pseudo-remainder can be computed as a determinant of a matrix of multivariate polynomials. So the computation of of each polynomial in $\mathfrak{P}_i$ $(i = m, \ldots, 2)$ is dominated by the above estimates on computing a polynomial of $\mathfrak{P}_1$ from $\mathfrak{P}_2$. Therefore, each polynomial in $\mathfrak{P}_1$ is computed within $2^{\mathcal{O}(n)} \ell^{\mathcal{O}(n)} \delta^{\mathcal{O}(n)\mathcal{O}(m^2)} \hbar^2$ bit operations. □

7. Algorithms

In this section, we present algorithms for LazyRealTriangularize and RealTriangularize that we have implemented. As a byproduct of RealTriangularize, we obtain an algorithm called SamplePoints which computes at least one sample point per connected component of a semi-algebraic set. Note that this SamplePoints algorithm is different from the one presented in our article [11], which is renamed as SampleOutHypersurface in this paper.

Algorithm 2: GeneratePreRegularSas($\mathfrak{S}$)

```
   Input: a semi-algebraic system $\mathfrak{S} = [F, N_{\geq}, P_{>}, H_{\neq}]$
   Output: a set of pre-regular semi-algebraic systems $[B_{i\neq}, T_i, P_{i>}]$, $i = 1 \ldots e$,
   such that $Z_{\mathbb{R}}(\mathfrak{S}) = \cup_{i=1}^{e} Z_{\mathbb{R}}(B_{i\neq}, T_i, P_{i>}) \cup_{i=1}^{e} Z_{\mathbb{R}}(\mathrm{sat}(T_i) \cup \{\Pi_{b \in B_i} b\}, N_{\geq}, P_{>}, H_{\neq})$.
 1 $\mathfrak{T} :=$ Triangularize$(F)$; $\mathfrak{T}' := \emptyset$
 2 for $p \in P \cup H$ do
 3     for $T \in \mathfrak{T}$ do
 4         for $C \in$ Regularize$(p, T)$ do
 5             if $p \notin \mathrm{sat}(C)$ then $\mathfrak{T}' := \mathfrak{T}' \cup \{C\}$
 6     $\mathfrak{T} := \mathfrak{T}'$; $\mathfrak{T}' := \emptyset$
 7 $\mathfrak{T} := \{[T, \emptyset] \mid T \in \mathfrak{T}\}$; $\mathfrak{T}' := \emptyset$
 8 for $p \in N$ do
 9     for $[T, N'] \in \mathfrak{T}$ do
10         for $C \in$ Regularize$(p, T)$ do
11             if $p \in \mathrm{sat}(C)$ then
12                 $\mathfrak{T}' := \mathfrak{T}' \cup \{[C, N']\}$
13             else
14                 $\mathfrak{T}' := \mathfrak{T}' \cup \{[C, N' \cup \{p\}]\}$
15     $\mathfrak{T} := \mathfrak{T}'$; $\mathfrak{T}' := \emptyset$
16 $\mathfrak{T} := \{[T, N', P, H] \mid [T, N'] \in \mathfrak{T}\}$
17 for $[T, N', P, H] \in \mathfrak{T}$ do
18     $B :=$ BorderPolynomialSet$(T, N' \cup P \cup H)$
19     output $[B, T, N' \cup P]$
```

Basic subroutines. The algorithms stated in this section rely on a few subroutines that we specify hereafter. For a zero-dimensional squarefree regular system $[T, P]$, the function call RealRootIsolate(T, P) [39] returns all the isolated real zeros of $[T, P_{>}]$. For $A \subset \mathbb{Q}[u_1, \ldots, u_d]$ and a point s of $\mathbb{Q}^d$ such that $p(s) \neq 0$ for all $p \in A$, the function call GenerateFormula(A, s) computes a formula $\wedge_{p \in A} (p\, \sigma_{p,s} > 0)$, where $\sigma_{p,s}$ is defined as $+1$ if $p(s) > 0$ and -1 otherwise. For a set of formulas G, the function call Disjunction(G) computes a logic formula Φ equivalent to the disjunction of the formulas in G.

Proof of Algorithm 2. Its termination is obvious. We prove its correctness. By the specification of Triangularize and Regularize, at line 16, we have

$$Z(F, P_{\neq} \cup H_{\neq}) \;=\; \cup_{[T,N',P,H] \in \mathfrak{T}} \, Z(\mathrm{sat}(T), P_{\neq} \cup H_{\neq}).$$

```
Algorithm 3: GenerateRegularSas($B, T, P$)
  Input: $\mathfrak{S} = [B_{\neq}, T, P_{>}]$, a pre-regular semi-algebraic system of $\mathbb{Q}[\mathbf{u}, \mathbf{y}]$, where
         $\mathbf{u} = u_1, \ldots, u_d$ and $\mathbf{y} = y_1, \ldots, y_{n-d}$.
  Output: A pair $(D, \mathcal{R})$ satisfying:
    (1) $D \subset \mathbb{Q}[\mathbf{u}]$ such that $\mathsf{factor}(B) \subseteq D$;
    (2) $\mathcal{R}$ is a finite set of regular semi-algebraic systems, such that we have:
        $\cup_{R \in \mathcal{R}} Z_{\mathbb{R}}(R) = Z_{\mathbb{R}}(D_{\neq}, T, P_{>})$.
  $D := \mathsf{factor}(B \setminus \mathbb{Q})$
  if $d = 0$ then
      if RealRootIsolate($T, P$) = [ ] then return $(D, \emptyset)$; else return $(D, \{[true, T, P]\})$
  while $true$ do
      $S :=$ SampleOutHypersurface($D, d$); $G_0 := \emptyset$; $G_1 := \emptyset$
      for $s \in S$ do
          if RealRootIsolate($T(s), P(s)$) = [ ] then
              $G_0 := G_0 \cup \{$GenerateFormula($D, s$)$\}$
          else
              $G_1 := G_1 \cup \{$GenerateFormula($D, s$)$\}$
      if $G_0 \cap G_1 = \emptyset$ then
          $\mathcal{Q} :=$ Disjunction($G_1$)
          if $\mathcal{Q} = false$ then return $(D, \emptyset)$; else return $(D, \{[\mathcal{Q}, T, P]\})$
      else
          select a subset $D' \subseteq \mathrm{oaf}(B) \setminus D$ by some heuristic method
          $D := D \cup D'$
```

Write $\cup_{[T,N',P,H] \in \mathfrak{T}}$ as $\cup_T$. Then we deduce that

$$Z_{\mathbb{R}}(F, N_{\geq}, P_{>}, H_{\neq}) \ = \ \cup_T \ Z_{\mathbb{R}}(\mathrm{sat}(T), N_{\geq}, P_{>}, H_{\neq}).$$

Between lines 17 and 19, for each $[T, N', P, H]$, we generate a pre-regular semi-algebraic system $[B_{\neq}, T, N'_{>} \cup P_{>}]$. By Corollary 1, we have

$$\begin{aligned} Z_{\mathbb{R}}(\mathrm{sat}(T), N_{\geq}, P_{>}, H_{\neq}) &= Z_{\mathbb{R}}(\mathrm{sat}(T), N'_{\geq}, P_{>}, H_{\neq}) \\ &= Z_{\mathbb{R}}(B_{\neq}, T, N'_{>} \cup P_{>}) \cup Z_{\mathbb{R}} \ (\mathrm{sat}(T) \cup \{\Pi_{b \in B} b\}, N_{\geq}, P_{>}, H_{\neq}), \end{aligned}$$

which implies that

$$Z_{\mathbb{R}}(\mathfrak{S}) \ = \ \cup_T \ (Z_{\mathbb{R}}(B_{\neq}, T, N'_{>} \cup P_{>}) \cup Z_{\mathbb{R}}(\mathrm{sat}(T) \cup \{\Pi_{b \in B} b\}, N_{\geq}, P_{>}, H_{\neq}))$$

holds. Therefore, Algorithm 2 satisfies its specification.

Proof of Algorithms 3 and 4. By the definition of oproj, Algorithm 4 terminates and satisfies its specification. By Theorem 5, $\mathrm{oaf}(B)$ is an FPS. Thus, by the definition of an FPS, Algorithm 3 terminates and satisfies its specification.

Proof of Algorithm 5. Its termination is obvious; we prove it is correct. Let R_i, $i = 1 \cdots t$ be the output. By the specification of each sub-algorithm, each R_i is a regular semi-algebraic system and we have $\cup_{i=1}^{t} Z_{\mathbb{R}}(R_i) \subseteq Z_{\mathbb{R}}(\mathfrak{S})$. Next we show that there exists

Algorithm 4: $\mathsf{SampleOutHypersurface}(A, k)$

```
Input: $A \subset \mathbb{Q}[x_1, \ldots, x_k]$ is a finite set of non-zero polynomials
Output: A finite subset of $\mathbb{Q}^k$ contained in $(\Pi_{p \in A}\, p) \neq 0$ and having a non-empty
        intersection with each connected component of $(\Pi_{p \in A}\, p) \neq 0$.
1 if $k = 1$ then
2     return one rational point from each connected component of $\Pi_{p \in A}\, p \neq 0$
3 else
4     $A_k := \{p \in A \mid \mathrm{mvar}(p) = x_k\}$; $A' := \mathrm{oproj}(A, x_k)$
5     for $s \in \mathsf{SampleOutHypersurface}(A', k-1)$ do
6         Collect in a set $S$ one rational point from each connected component of
          $\Pi_{p \in A_k} p(s, x_k) \neq 0$;
7         for $\alpha \in S$ do output $(s, \alpha)$
```

Algorithm 5: $\mathsf{LazyRealTriangularize}(\mathfrak{S})$

```
Input: a semi-algebraic system $\mathfrak{S} = [F, N_{\geq}, P_{>}, H_{\neq}]$
Output: a lazy triangular decomposition of $\mathfrak{S}$
1 $\mathfrak{T} := \mathsf{GeneratePreRegularSas}(F, N, P, H)$
2 for $[B, T, P'] \in \mathfrak{T}$ do
3     $(D, \mathcal{R}) = \mathsf{GenerateRegularSas}(B, T, P')$
4     if $\mathcal{R} \neq \emptyset$ then output $\mathcal{R}$
```

Algorithm 6: $\mathsf{RealTriangularize}(\mathfrak{S})$

```
Input: a semi-algebraic system $\mathfrak{S} = [F, N_{\geq}, P_{>}, H_{\neq}]$
Output: a triangular decomposition of $\mathfrak{S}$
1 $\mathfrak{T} := \mathsf{GeneratePreRegularSas}(F, N, P, H)$
2 for $[B, T, P'] \in \mathfrak{T}$ do
3     $(D, \mathcal{R}) = \mathsf{GenerateRegularSas}(B, T, P')$
4     if $\mathcal{R} \neq \emptyset$ then output $\mathcal{R}$
5     for $p \in D$ do
6         output $\mathsf{RealTriangularize}(F \cup \{p\}, N, P, H)$
```

an ideal $\mathcal{I} \subseteq \mathbb{Q}[\mathbf{x}]$, whose dimension is less than $\dim(Z(F, P_{\neq} \cup H_{\neq}))$ and such that $Z_{\mathbb{R}}(\mathfrak{S}) \setminus \cup_{i=1}^{t} Z_{\mathbb{R}}(R_i) \subseteq Z_{\mathbb{R}}(\mathcal{I})$ holds. At line 1, the specification of Algorithm 2 imply:

$$Z_{\mathbb{R}}(\mathfrak{S}) = \cup_T Z_{\mathbb{R}}(B_{\neq}, T, P'_{>}) \ \cup\ \cup_T Z_{\mathbb{R}}(\mathrm{sat}(T) \cup \{\Pi_{b \in B}\ b\}, N_{\geq}, P_{>}, H_{\neq}).$$

At line 3, by the specification of Algorithm 3, for each B, we compute a set D such that $\mathsf{factor}(B) \subseteq D$ and

$$\cup_T Z_{\mathbb{R}}(D_{\neq}, T, P'_{>}) = \cup_{i=1}^{t} Z_{\mathbb{R}}(R_i) \tag{2}$$

both hold. Following the strategy used in Algorithm 2, based on Corollary 1, we have

$$Z_{\mathbb{R}}(\mathfrak{S}) = \cup_T Z_{\mathbb{R}}(D_{\neq}, T, P'_{>}) \ \cup\ \cup_T Z_{\mathbb{R}}(\mathrm{sat}(T) \cup \{\Pi_{p \in D}\ p\}, N_{\geq}, P_{>}, H_{\neq}). \tag{3}$$

Combining the relations (2) and (3) together, we obtain

$$Z_{\mathbb{R}}(\mathfrak{S}) = \cup_T Z_{\mathbb{R}}(R_i) \ \cup\ \cup_T Z_{\mathbb{R}}(\mathrm{sat}(T) \cup \{\Pi_{p \in D}\, p\}, N_{\geq}, P_{>}, H_{\neq}).$$

Therefore, the following relations hold

$$\begin{aligned} Z_{\mathbb{R}}(\mathfrak{S}) \setminus \cup_{i=1}^{t} Z_{\mathbb{R}}(R_i) &\subseteq \cup_T Z_{\mathbb{R}}(\mathrm{sat}(T) \cup \{\Pi_{p\in D}\, p\}, N_{\geq}, P_{>}, H_{\neq}) \\ &\subseteq Z_{\mathbb{R}}\left(\cap_T\left(\mathrm{sat}(T) \cup \{\Pi_{p\in D}\, p\}\right)\right). \end{aligned}$$

Define $\mathcal{I} = \cap_T\left(\mathrm{sat}(T) \cup \{\Pi_{p\in D}\, p\}\right)$. Since each $p \in D$ is regular modulo $\mathrm{sat}(T)$, we have $\dim(\mathcal{I}) < \dim\left(\cap_T \mathrm{sat}(T)\right) \leq \dim(Z(F, P_{\neq} \cup H_{\neq}))$. So all R_i form a lazy triangular decomposition of $\mathfrak{S}$. □

Proof of Algorithm 6. For its termination, it is sufficient to prove that there are only finitely many recursive calls to RealTriangularize. Indeed, if $[F, N, P, H]$ is the input of a call to RealTriangularize then each of the immediate recursive calls takes $[F \cup \{p\}, N, P, H]$ as input, where p belongs to the set D of some pre-regular semi-algebraic system $[D_{\neq}, T, P_{>}]$. Since p is regular (and non-zero) modulo $\mathrm{sat}(T)$ we have: $\langle F\rangle \subsetneq \langle F \cup \{p\}\rangle$. Therefore, the algorithm terminates by the ascending chain condition on ideals of $\mathbb{Q}[\mathbf{x}]$. The correctness of Algorithm 6 follows from that of its sub-algorithms. □

Implementation remark for LazyRealTriangularize. Our software implementation (within the `RegularChains` library in Maple) of Algorithm 5 returns the necessary information for completing a full triangular decomposition of the input semi-algebraic system $\mathfrak{S}$. This is achieved simply by returning $[F \cup \{p\}, N, P, H]$ for each $p \in D$, for each D.

For an input semi-algebraic system $\mathfrak{S}$ Algorithm 7 computes a sample point set of $\mathfrak{S}$, see Definition 5, thus producing at least one point per connected component of $Z_{\mathbb{R}}(\mathfrak{S})$.

Definition 5. Let S be a semi-algebraic set of $\mathbb{R}^n$. A finite subset A of $\mathbb{R}^n$ is called a *sample point set* of S if the following conditions hold:

(i) every point of A belongs to some connected component of S,

(2) every connected component of S has a nonempty intersection with A.

Algorithm 7: SamplePoints($\mathfrak{S}$)

```
Input:  a semi-algebraic system $\mathfrak{S} = [F, N_{\geq}, P_{>}, H_{\neq}]$
Output: A sample point set of $\mathfrak{S}$.
$\mathfrak{T}$ := GeneratePreRegularSas($F, N, P, H$)
for $[B, T, P'] \in \mathfrak{T}$ do
    for $s \in$ SampleOutHypersurface($B$) do
        for $\alpha \in$ RealRootIsolate($T(s), P'(s)$) do
            output $(s, \alpha)$
    for $p \in B$ do
        output SamplePoints($F \cup \{p\}, N, P, H$)
```

Lemma 7. Let S, S_1 and S_2 be nonempty semi-algebraic sets of $\mathbb{R}^n$. Assume that $S = S_1 \cup S_2$. Let A_1 (resp. A_2) be a sample point set of S_1 (resp. S_2). Then $A_1 \cup A_2$ is a sample point set of S.

Proof. First, any point of $A_1 \cup A_2$ obviously belongs to S and therefore belongs to some connected component of S. Secondly, we want to prove that each connected component

Table 1 Notations for Tables 2 and 3

symbol	meaning
#e	number of equations in the input system
#v	number of variables in the input equations
d	maximum total degree of an input equation
G	Groebner:-Basis (plex order) in Maple
T	Triangularize in `RegularChains` library of Maple
ST	Squarefree Triangularize in `RegularChains` library of Maple
LR	LazyRealTriangularize implemented in Maple
R	RealTriangularize implemented in Maple
S	SamplePoints implemented in Maple
Q	Qepcad b 1.61
$> 1h$	computation does not complete within 1 hour
FAIL	Qepcad b failed due to prime list exhausted

of S contains at least one point of $A_1 \cup A_2$. We prove this by contradiction. Suppose C is a connected component of S that does not contain any point of $A_1 \cup A_2$ $(*)$. Let $p \in C$. Then p belongs to some connected component D of S_1 or S_2. Let q be a point of $A_1 \cup A_2$ such that q belongs to D. Then there exists a path $L(p, q)$ connecting p and q, which is contained in D and hence contained in S. So p and q belongs to the same connected component of S, which implies that $q \in C$ holds. This is a contradiction to $(*)$. □

Proof of Algorithm 7. The proof of its termination is exactly the same as that of RealTriangularize. It correctness follows from Lemma 7 and Theorem 1.

8. Experimentation

We have implemented our algorithms on top of the `RegularChains` library in Maple. Hereafter, we report on experimental results using well known benchmark examples from the literature. The test examples are available at `www.orcca.on.ca/~cchen/issac10.txt`.

Table 1. Table 1 summarizes the notations used in Tables 2 and 3. Tables 2 and 3 demonstrate benchmarks running in Maple 15, using an Intel Core 2 Quad CPU (2.40GHz) with 3.0GB memory. The timings are in seconds and the time-out is 1 hour.

Table 2. The systems in this group involve equations only. We list the running times for computing a triangular decomposition of the input algebraic variety as well as a lazy and a full triangular decomposition of the corresponding real variety. We also provide the running times for computing lexicographical Gröbner bases with the Maple function `Groebner:-Basis`. The data illustrate the performance of LazyRealTriangulrize, RealTriangulrize and SamplePoints.

Table 3. The systems in this table are from quantifier elimination problems. Most of them involve both equations and inequalities. We provide the timings for computing (1) a lazy

Table 2 Timings for varieties

system	#v/#e/d	G	T	ST	LR	R	S
Hairer-2-BGK	13/11/4	24.64	2.05	2.08	2.96	4.20	5.55
Collins-jsc02	5/4/3	$> 1h$	0.52	0.52	1.81	560.92	10.82
Leykin-1	8/6/4	101.44	4.00	4.02	4.39	5.46	5.72
8-3-config-Li	12/7/2	110.24	5.96	6.01	7.38	417.90	446.29
Lichtblau	3/2/11	126.35	0.31	0.32	3.55	$> 1h$	$> 1h$
Cinquin-3-3	4/3/4	64.84	0.70	0.76	2.34	$> 1h$	57.23
Cinquin-3-4	4/3/5	$> 1h$	3.47	3.43	15.19	$> 1h$	$> 1h$
DonatiTraverso-rev	4/3/8	159.95	1.89	2.23	3.34	3.02	2.98
Cheaters-homotopy-1	7/3/7	2498.78	0.65	451.33	$> 1h$	$> 1h$	$> 1h$
hereman-8.8	8/6/6	$> 1h$	12.92	22.24	$> 1h$	$> 1h$	110.34
L	12/4/3	$> 1h$	0.79	0.80	1.12	14.94	18.16
dgp6	17/19/2	27.38	48.62	49.62	51.75	62.99	70.74
dgp29	5/4/15	85.70	0.20	0.20	0.37	0.38	0.33

Table 3 Timings for semi-algebraic systems

system	#v/#e/d	T	ST	LR	R	S	Q
BM05-1	4/2/3	0.28	0.28	0.65	1.15	1.19	8.16
BM05-2	4/2/4	0.29	0.29	3.50	$> 1h$	$> 1h$	FAIL
Solotareff-4b	5/4/3	0.91	0.93	1.98	881.15	14.42	$> 1h$
Solotareff-4a	5/4/3	0.71	0.74	1.63	4.00	3.12	FAIL
putnam	6/4/2	0.27	0.30	0.76	1.65	1.70	$> 1h$
MPV89	6/3/4	0.23	0.29	0.89	2.75	2.42	$> 1h$
IBVP	8/5/2	0.58	0.62	1.26	14.23	13.89	$> 1h$
Lafferriere37	3/3/4	0.33	0.38	0.69	0.72	0.62	2.3
Xia	6/3/4	0.46	0.46	2.20	209.65	168.49	$> 1h$
SEIT	11/4/3	0.70	0.71	32.67	$> 1h$	1355.81	$> 1h$
p3p-isosceles	7/3/3	0.35	0.35	$> 1h$	$> 1h$	$> 1h$	$> 1h$
p3p	8/3/3	0.37	0.40	$> 1h$	$> 1h$	$> 1h$	FAIL
Ellipse	6/1/3	0.18	0.19	0.96	$> 1h$	$> 1h$	$> 1h$

triangular decomposition, (2) a full triangular decomposition and (3) sample points of the corresponding semi-algebraic systems as well as the timings for solving the quantifier elimination problem via Qepcad b [6] (in non-interactive mode). Our tools complete the computations for most of the systems. However, one should note that the output of our tools is not a solution to the posed quantifier elimination probem. We note also that our tools are more effective for systems counting more equations than inequalities.

We conclude this section by reporting on an experimental comparison of `SamplePoints` versus related software tools. Among the software that we can access, we could find only one software function with the same specifications as `SamplePoints`, that is, a function computing a sample point set, see Definition 5, for an arbitrary semi-algebraic system. This function is the `SemialgebraicComponentInstances` command in MATHEMATICA. We have tested the function `SemialgebraicComponentInstances` in MATHEMATICA 8 for the systems (26 in total) listed in Table 2 and Table 3. We have found that this command succeeded for 9 of them, within the same resource limit and the same machine as described above, while `SamplePoints` could solve 19 of those systems. Among the 9 systems that `SemialgebraicComponentInstances` could solve, `SamplePoints` failed on 3 of them.

9. Illustrative examples

We consider examples arising in the study of dynamical systems and program verification. We apply the `RegularChains` library implementation of the algorithms of Section 7.

9.1. Program verification

Recent advances in program verification indicate that various problems, for instance, termination analysis of linear programs [37], reachability computation of linear hybrid systems [28], and invariant generation [19, 34] can be reduced to solving semi-algebraic systems. Tools for real algebraic computation such as REDLOG [23] QEPCAD [18, 25, 6], and DISCOVERER [42] have therefore been applied to program verification [19, 28].

We consider here Example 3.5 from [28]. This problem reduces to determine the set

$$\{(y_1, y_2) \in \mathbb{R}^2 \mid (\exists a \in \mathbb{R})(\exists z \in \mathbb{R})\ (0 \le a) \wedge (z \ge 1) \wedge (h_1 = 0) \wedge (h_2 = 0)\}$$

where $h_1 = 3\,y_1 - 2\,a(-z^4 + z)$ and $h_2 = 2\,y_2 z^2 - a(z^4 - 1)$. In order words, one wishes to compute the projection of the semi-algebraic set defined by $(0 \le a) \wedge (z \ge 1) \wedge (h_1 = 0) \wedge (h_2 = 0)$ onto the (y_1, y_2)-plane. This question can be answered by running the RealTriangularize command on the semi-algebraic set for the variable ordering $a > z > y_1 > y_2$. We obtain the five following regular semi-algebraic systems R_1 to R_5 (unspecified R_i^P and R_i^Q are empty):

$$R_1^T = \begin{cases} \left(z^4 - 1\right) a - 2\,z^2 y_2 \\ 4\,y_2\,z^5 + 4\,y_2\,z^4 + (3\,y_1 + 4\,y_2)\,z^3 + 3\,y_1\,z^2 + 3\,y_1\,z + 3\,y_1 \end{cases}$$

$$R_1^Q = \begin{cases} (y_1 + y_2 < 0) \wedge (y_1 < 0) \wedge (0 < y_2) \\ 3y_1^5 - 6y_2y_1^4 - 63y_2^2y_1^3 + 192y_2^3y_1^2 + 112y_2^4y_1 + 16y_2^5 \neq 0 \end{cases} \qquad R_1^P = \begin{cases} z > 1 \end{cases}$$

$$R_2^T = \begin{cases} a \\ y_1 \\ y_2 \end{cases} \qquad R_3^T = \begin{cases} z - 1 \\ y_1 \\ y_2 \end{cases} \qquad R_4^T = \begin{cases} a \\ z - 1 \\ y_1 \\ y_2 \end{cases}$$

$$R_2^P = \begin{cases} z > 1 \end{cases} \qquad R_3^P = \begin{cases} 0 < a \end{cases}$$

$$R_5^T = \begin{cases} \left(z^4 - 1\right) a - 2\, z^2 y_2 \\ t_z \\ 3\, y_1{}^5 - 6\, y_2\, y_1{}^4 - 63\, y_2{}^2 y_1{}^3 + 192\, y_2{}^3 y_1{}^2 + 112\, y_2{}^4 y_1 + 16\, y_2{}^5 \end{cases}$$

$$R_5^Q = \left\{ 0 < y_2 \qquad R_5^P = \left\{ z > 1 \right.\right.$$

where

$$\begin{aligned} t_z = {} & (369252163868\, y_1{}^4 - 2508200686544\, y_2\, y_1{}^3 + 4300300820416\, y_2{}^2 y_1{}^2 + 2761812320448\, y_2{}^3 y_1 \\ & + 406754520832\, y_2{}^4) z^4 + (-180672905280\, y_2{}^4 - 1228579249664\, y_2{}^3 y_1 - 1922937082240\, y_2{}^2 y_1{}^2 \\ & + 1092105551100\, y_2\, y_1{}^3 - 157082832940\, y_1{}^4) z^3 + (-815128066608\, y_2{}^4 - 5538434025360\, y_2{}^3 y_1 \\ & - 8644620182000\, y_2{}^2 y_1{}^2 + 4979116186797\, y_2\, y_1{}^3 - 728379335938\, y_1{}^4) z^2 + (-316725331280\, y_2{}^4 \\ & - 276096356865\, y_1{}^4 + 1914148321163\, y_2\, y_1{}^3 - 3371008535808\, y_2{}^2 y_1{}^2 - 2153737071904\, y_2{}^3 y_1) z \\ & - 1030979306368\, y_2{}^4 - 10923966861712\, y_2{}^2 y_1{}^2 + 6315633355800\, y_2\, y_1{}^3 - 7003676730320\, y_2{}^3 y_1 \\ & - 923425115541\, y_1{}^4. \end{aligned}$$

The projection on the (y_1, y_2)-plane of $Z_{\mathbb{R}}(R_2) \cup Z_{\mathbb{R}}(R_3) \cup Z_{\mathbb{R}}(R_4)$ is clearly equal to the $(y_1, y_2) = (0, 0)$ point. Properties (iii) of Definition 1 implies that the projection on the (y_1, y_2)-plane of $Z_{\mathbb{R}}(R_1)$ is given by $Z_{\mathbb{R}}(R_1^Q)$. For R_5, we observe that the polynomial of R_5^T with main variable y_1, say t_{y_1} is delineable above $0 < y_2$ (By Theorem 1). Using a sample point we check that t_{y_1} admits a single real root. It follows that the projection on the (y_1, y_2)-plane of $Z_{\mathbb{R}}(R_5)$ is given by:

$$(0 < y_2) \wedge (3\, y_1{}^5 - 6\, y_2\, y_1{}^4 - 63\, y_2{}^2 y_1{}^3 + 192\, y_2{}^3 y_1{}^2 + 112\, y_2{}^4 y_1 + 16\, y_2{}^5 = 0).$$

To conclude, we have completed the projection of the semi-algebraic set onto the (y_1, y_2)-plane, which can be simplified as $(y_1 < 0 \wedge y_2 > 0 \wedge y_1 + y_2 < 0) \vee (y_1 = 0 \wedge y_2 = 0)$.

9.2. Dynamical systems

In [29], Laurent proposed a model for the dynamics of diseases of the central nervous system caused by prions, such as scrapie in sheep and goat, and “mad cow disease” or Creutzfeldt-Jacob disease in humans. The model is based on the protein-only hypothesis, which assumes that infection can be spread by particular proteins (prions) that can exist in two isomeric forms. The normal form PrP^C is harmless, while the infectious form PrP^{Sc} catalyzes a transformation from the normal form to itself.

The dynamical system system ruling this transformation is given by

$$\frac{\mathrm{d}x}{\mathrm{d}t} = k_1 - k_2 x - ax\frac{(1 + by^n)}{1 + cy^n}$$
$$\frac{\mathrm{d}y}{\mathrm{d}t} = ax\frac{(1 + by^n)}{1 + cy^n} - k_4 y$$

where x and y are respectively the concentrations of PrP^C and PrP^{Sc}. The symbols b, c, n, a, k_4, k_1 are biological constants which can be set as follows: $b = 2$, $c = 1/20$, $n = 4$, $a = 1/10$, $k_4 = 50$ and $k_1 = 800$. Hence, we obtain a dynamical system with only one (positive) parameter k_2:

$$\begin{aligned}\frac{\mathrm{d}x}{\mathrm{d}t} &= \frac{16000 + 800y^4 - 20k_2x - k_2xy^4 - 2x - 4xy^4}{20 + y^4}\\ \frac{\mathrm{d}y}{\mathrm{d}t} &= \frac{2(x + 2xy^4 - 500y - 25y^5)}{20 + y^4}\end{aligned} \tag{4}$$

Since $20 + y^4$ is always positive, the semi-algebraic system describing the equilibria is:

$$\mathcal{S} := \begin{cases} 16000 + 800y^4 - 20k_2x - k_2xy^4 - 2x - 4xy^4 = 0 \\ 2(x + 2xy^4 - 500y - 25y^5) = 0 \\ k_2 > 0 \end{cases}$$

Applying LazyRealTriangularize to this system, yields the following regular semi-algebraic system (and unevaluated recursive calls)

$$\begin{cases} (2y^4 + 1)x - 500y - 25y^5 = 0 \\ (k_2 + 4)y^5 - 64y^4 + (20k_2 + 2)y - 32 = 0 \\ (k_2 > 0) \ \wedge \ (R_1 \neq 0) \end{cases}$$

where

$$\begin{aligned} R_1 = {}& 100000k_2^8 + 1250000k_2^7 + 5410000k_2^6 + 8921000k_2^5 - 9161219950k_2^4 \\ & - 5038824999k_2^3 - 1665203348k_2^2 - 882897744k_2 + 1099528405056. \end{aligned}$$

Through the computation of sample points, we easily obtain the following observation. Whenever $R_1 > 0$ holds, System (4) has 1 equilibrium, while $R_1 < 0$ implies that System (4) has 3 equilibria.

Now we study the stability of those equilibria. To this end, we consider the two Hurwitz determinants of $\mathcal{S}$, which are (forgetting their denominators $(20 + y^4)^2$):

$$\begin{aligned} \Delta_1 &= 54y^8 + 40k_2y^4 + 2082y^4 - 312xy^3 + 20040 + k_2y^8 + 400k_2, \\ a_2 &= 20000k_2 + 2000 + 50k_2y^8 + 200y^8 + 2000k_2y^4 - 312k_2xy^3 + 4100y^4. \end{aligned}$$

Adding to $\mathcal{S}$ the constraints $\{\Delta_1 > 0, a_2 > 0\}$ we obtain a new semi-algebraic system $\mathcal{S}'$. Applying LazyRealTriangularize to $\mathcal{S}'$ in conjunction with sample point computations brings the following conclusion. If $R_1 > 0$, then System (4) has 1 asymptotically stable hyperbolic equilibria; If $R_1 < 0$ and $R_2 \neq 0$, then System (4) has 2 asymptotically stable hyperbolic equilibria, where R_2 is

$$\begin{aligned} & 10004737927168k_2^9 + 624166300700672k_2^8 + 7000539052537600k_2^7 + 45135589467012800k_2^6 \\ & - 840351411856453750k_2^5 - 5009800435224844 6875k_2^4 - 2738816898945500000 0k_2^3 \\ & - 8675209266696000000k_2^2 + 10296091735680000000k_2 + 59325460641024000000. \end{aligned}$$

To further investigate the number of asymptotically stable hyperbolic equilibria on the hypersurface $R_2 = 0$ and the equilibria when $R_1 = 0$, one can apply SamplePoints on $\mathcal{S}'$, which produces 14 points. From those, all the possible equilibrium configurations of the dynamical system can be determined.

10. Discussion and concluding remarks

Given a semi-algebraic system $\mathfrak{S}$ the algorithm RealTriangularize (resp. LazyRealTriangularize), as stated in Section 7, returns a full (resp. lazy) triangular decomposition of $\mathfrak{S}$. Consider $R = [\mathcal{Q}, T, P_{>}]$ an output regular semi-algebraic system and assume that T admits $x_1 < \cdots < x_d$ as free variables, for $d > 0$. Let C be a connected component of the semi-algebraic set defined by $\mathcal{Q}$ in $\mathbb{R}^d$. Theorem 1 states that, above C, the set $Z_{\mathbb{R}}(R)$ consists of finitely many disjoint graphs of continuous functions where each of these graphs is locally homeomorphic to the hypercube $(0,1)^d$. Therefore R can be regarded as a parameterization of $Z_{\mathbb{R}}(R)$.

This situation is similar to that of triangular decomposition of algebraic sets. Indeed, consider an input polynomial system $F \in \mathbf{k}[\mathbf{x}]$, for a field $\mathbf{k}$, to which the algorithm Triangularize is applied. Consider also an output regular chain T with $x_1 < \cdots < x_d$ as free variables, for $d > 0$. Then T represents a generic zero for each irreducible component of $V(\mathrm{sat}(T))$; moreover each of these irreducible components has dimension d.

The complexity results of Sections 4 and 6 together with the experimental results of Section 8 suggest that the notions and algorithms presented in this paper are promising tools for manipulating semi-algebraic sets symbolically. In the sequel of this section, we would like to address the following natural question: would there be an alternative and competitive algorithm implementing the specifications of LazyRealTriangularize while relying on existing tools from the literature?

One direct approach for computing a lazy triangular decomposition of the semi-algebraic system $\mathfrak{S}$ could be the following.

(i) Decompose $\mathfrak{S}$ into pre-regular semi-algebraic systems, using Algorithm 2.

(ii) For each output pre-regular semi-algebraic system $[B_{\neq}, T, P_{>}]$ compute a CAD of the complement of the hypersurface defined by B in the parameter space, where this CAD produces for each cell a sample point s and a Tarski formula Φ defining that cell.

(iii) For each $[B_{\neq}, T, P_{>}]$ for each (s, Φ) associated with $[B_{\neq}, T, P_{>}]$, if the specialized system $[T(s), P_{>}(s)]$ has real solutions then output $[\Phi, T, P_{>}]$.

In our approach we modify Step (ii) (and Step (iii)) and avoid the computation of a full CAD by reducing to the following quantifier elimination problem:

$$\exists \mathbf{y}(B(\mathbf{u}) \neq 0, T(\mathbf{u}, \mathbf{y}) = 0, P(\mathbf{u}, \mathbf{y}) > 0).$$

See Section 5 for details. When B is a fingerprint polynomial set, we solve this problem by computing (at least) one sample point in each connected component of the complement of the hypersurface defined by B in the parameter space. Then, the properties of an FPS yield the Tarski formulas from the polynomials in the FPS. When B is not a fingerprint polynomial set, we replace B by a superset D of B, which is an FPS.

A first advantage of our approach is that the concept of an FPS is independent of the elimination procedure (CAD or other). Actually, we have described two strategies for FPS construction: one based on open augmented projection (Section 5) and one based on generalized discriminant sequences (Section 6). A second advantage is that when T is in generic position, an FPS of $[B_{\neq}, T, P_{>}]$ can be computed in singly exponential time w.r.t. the number of variables. It is worth noticing that this case occurs very frequently in practice. Another important practical observation is the fact that, often, a fairly small subset of the theoretical FPS (the set $\mathrm{oaf}(B)$ in Theorem 5 and the set D in Proposition 4) is already an FPS. We take advantage of this latter observation in our implementation.

Regarding the construction and the use of an FPS, we conclude with two remarks. First, in our implementation and as suggested by Algorithm 3, an FPS is constructed by an incremental process starting from B. A related procedure appears in [4] where a CAD augmented projection is computed incrementally so as to produce a projection-definable CAD. One difference is that, in the FPS construction based on open augmented projection, the considered cells (in the space of the free variables of T) are all open. In the case of the augmented projection construction [4] cells of lower dimension may need to be considered as well. Secondly, we observe that, in principle, Algorithm 4 may be replaced by any procedure computing at least one rational point per connected component of the complement of a hypersurface. Despite of its doubly exponential running time, we have verified experimentally that our implementation of Algorithm 4 is competitive with other tools, such as MAPLE's command `RootFinding:- WitnessPoints.`

References

[1] P. Aubry, D. Lazard, and M. Moreno Maza. On the theories of triangular sets. *J. Symb. Comput.*, 28(1-2):105–124, 1999.

[2] S. Basu, R. Pollack, and M-F. Roy. *Algorithms in real algebraic geometry.* Springer-Verlag, 2006.

[3] F. Boulier, C. Chen, F. Lemaire, and M. Moreno Maza. Real root isolation of regular chains. In *Proc. ASCM'09.*

[4] C. W. Brown. Guaranteed solution formula construction. In *Proc. ISSAC'99*, pages 137–144, 1999.

[5] C. W. Brown. Improved projection for cylindrical algebraic decomposition. *J. Symb. Comput.*, 32(5):447–465, 2001.

[6] C. W. Brown. QEPCAD B: a program for computing with semi-algebraic sets using cads. *SIGSAM Bull.*, 37(4):97–108, 2003.

[7] C. W. Brown and J. H. Davenport. The complexity of quantifier elimination and cylinrical algebraic decomposition. In *Proc. ISSAC'07*, pages 54–60, 2007.

[8] C. W. Brown and S. McCallum. On using bi-equational constraints in cad construction. In *ISSAC'05*, pages 76–83, 2005.

[9] P. Bürgisser, M. Clausen, and M. A. Shokrollahi. *Algebraic Complexity Theory.* Springer, 1997.

[10] B. Caviness and J. Johnson, editors. *Quantifier Elimination and Cylindical Algebraic Decomposition, Texts and Mongraphs in Symbolic Computation.* Springer, 1998.

[11] C. Chen, J.H. Davenport, J. May, M. Moreno Maza, B. Xia, and R. Xiao. Triangular decomposition of semi-algebraic systems. In *Proc. of ISSAC'10*, ACM Press, pages 187–194, 2010.

[12] C. Chen, J. H. Davenport, M. Moreno Maza, B. Xia, and R. Xiao. Computing with semi-algebraic sets represented by triangular decomposition. In *Proc. of ISSAC'11*, 2011.

[13] C. Chen, O. Golubitsky, F. Lemaire, M. Moreno Maza, and W. Pan. Comprehensive triangular decomposition. In *Proc. of CASC'07*, volume 4770 of *Lecture Notes in Computer Science*, pages 73–101, 2007.

[14] C. Chen and M. Moreno Maza. Algorithms for computing triangular decompositions of polynomial systems. In *Proc. of ISSAC'11*, 2011.

[15] C. Chen, M. Moreno Maza, B. Xia, and L. Yang. Computing cylindrical algebraic decomposition via triangular decomposition. In *ISSAC'09*, pages 95–102, 2009.

[16] J.S. Cheng, X.S. Gao, and C.K. Yap. Complete numerical isolation of real zeros in zero-dimensional triangular systems. In *ISSAC '07*, pages 92–99, 2007.

[17] G. E. Collins. Quantifier elimination for real closed fields by cylindrical algebraic decomposition. *Springer Lecture Notes in Computer Science*, 33:515–532, 1975.

[18] G. E., Collins and H. Hong. Partial cylindrical algebraic decomposition for quantifier elimination. J. of Symbolic Computation, 12:299328, 1991.

[19] M. Colón, S. Sankaranarayanan and H.B. Sipma. Linear invariant generation using nonlinear constraint solving. In CAV'03, LNCS 2725, pp. 420432, 2003.

[20] X. Dahan, A. Kadri, and É. Schost. Bit-size estimates for triangular sets in positive dimension. Technical report, University of Western Ontario, 2009.

[21] X. Dahan, M. Moreno Maza, É. Schost, W. Wu, and Y. Xie. Lifting techniques for triangular decompositions. In *ISSAC'05*, pages 108–115, 2005.

[22] J.H. Davenport, Y. Siret, and E. Tournier. *Computer Algebra.* Academic Press, 1988.

[23] A. Dolzman and T. Sturm. REDLOG: Computer algebra meets computer logic. ACM SIGSAM Bulletin, 31(2):29.

[24] H. Hong and J. R. Sendra. Computation of variant results, B. Caviness and J. Johnson, eds, *Quantifier Elimination and Cylindrical Algebraic Decomposition*, Texts and Monographs in Symbolic Computation. Springer Verlag, 1998.

[25] H. Hong *et al.* QEPCAD B, `www.usna.edu/Users/cs/qepcad/`.

[26] M. Kalkbrener. A generalized euclidean algorithm for computing triangular representations of algebraic varieties. *J. Symb. Comp.*, 15:143–167, 1993.

[27] T. Krick, L. M. Pardo, and M. Sombra. Sharp estimates for the arithmetic Nullstellensatz. *Duke Math. J.*, 109(3):521–598, 2001.

[28] Gerardo Lafferriere, George J. Pappas, Sergio Yovine. Symbolic Reachability Computation for Families of Linear Vector Fields. J. Symb. Comput. 32(3): 231-253 (2001)

[29] M. Laurent. Prion diseases and the protein only hypothesis: a theoretical dynamic study. *Biochem. J.*, 318, 1996

[30] X. Li, M. Moreno Maza, and W. Pan. Computations modulo regular chains. In *ISSAC'09*, pages 239–246, 2009.

[31] M. Moreno Maza. On triangular decompositions of algebraic varieties. MEGA-2000, Bath, UK. `http://www.csd.uwo.ca/`∼moreno/books-papers.html

[32] P. Philippon. Sur des hauteurs alternatives III. *J. Math. Pures Appl.*, 74(4):345–365, 1995.

[33] J. Renegar. On the computational complexity and geometry of the first-order theory of the reals. parts I–III. *J. Symb. Comput.*, 13(3):255–352, 1992.

[34] S. Sankaranarayanan, H.B. Sipma, and Z. Manna. Non-linear loop invariant generation using Gröbner bases. In ACM POPL'04, pp. 318329, 2004.

[35] A. Strzeboński. Solving systems of strict polynomial inequalities. *J. Symb. Comput.*, 29(3):471–480, 2000.

[36] Á. Szántó. *Computation with polynomial systems.* PhD thesis, Cornell University, 1999.

[37] A. Tiwari. Termination of linear programs. In CAV'04, LNCS 3114, pp. 387–390, 2004.

[38] W. T. Wu. A zero structure theorem for polynomial equations solving. *MM Research Preprints*, 1:2–12, 1987.

[39] B. Xia and T. Zhang. Real solution isolation using interval arithmetic. *Comput. Math. Appl.*, 52(6-7):853–860, 2006.

[40] R. Xiao. *Parametric Polynomial System Solving.* PhD thesis, Peking University, Beijing, 2009.

[41] L. Yang, X. Hou, and B. Xia. A complete algorithm for automated discovering of a class of inequality-type theorems. *Science in China, Series* **F**, 44(6):33–49, 2001.

[42] L. Yang and B. Xia. *Automated proving and discovering inequalities.* Science Press, Beijing, 2008.

[43] L. Yang and B. Xia. Real solution classifications of a class of parametric semi-algebraic systems. In *A3L'05*, pages 281–289, 2005.