



HAL
open science

Computing points in connected components defined by a real inequation: algorithms, complexity and implementations, Part I

Jérémy Berthomieu, Edern Gillot, Mohab Safey El Din

► To cite this version:

Jérémy Berthomieu, Edern Gillot, Mohab Safey El Din. Computing points in connected components defined by a real inequation: algorithms, complexity and implementations, Part I. 2026. ⟨hal-05612270v1⟩

HAL Id: hal-05612270

<https://hal.science/hal-05612270v1>

Preprint submitted on 5 May 2026 (v1), last revised 26 May 2026 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

Computing points in connected components defined by a real inequation: algorithms, complexity and implementations, Part I

Jérémy Berthomieu¹, Edern Gillot¹, and Mohab Safey El Din¹

¹Sorbonne Université, CNRS, LIP6, Paris, France

May 5, 2026

Abstract

We consider the problem of computing sample points in each connected component of a semi-algebraic set defined by the non-vanishing or the positivity of an n -variate polynomial of degree d , with rational coefficients of bit size bounded by τ .

Such a problem is a basic routine in effective real algebraic geometry, used in higher-level algorithms for solving polynomial systems over the reals and finds many applications in sciences.

We design a probabilistic algorithm for solving this problem, which is based on reductions to different routines for solving zero-dimensional polynomial systems. It assumes that the input polynomial satisfies sufficiently generic properties (namely, smoothness of its defining hypersurface). This is done through the computations of critical points of well-chosen maps to capture the connected components of the semi-algebraic set under study.

We derive a bit complexity estimate for the cost of this algorithm, which is, in terms of the Bézout bound $d(d-1)^{n-1}$, essentially cubic for obtaining parametrisations of the sought-for real points. Moreover, we also consider the case of obtaining rational approximations of those points, which are precise enough to lie in the same connected components as their exact counterparts, which yields a cost that is essentially quartic in the Bézout bound. In these complexity estimates, we take into account the degree structure of the input polynomial and its partial derivatives, allowing for a more refined bit complexity when the partial derivative of the input polynomial have degree lower than expected. We also analyse the probability of success of those algorithms.

We report on practical experiments, benchmarking with random dense input polynomials as well as polynomials coming from applications, which were out of reach of the state-of-the-art implementations, and hence illustrate the practical efficiency of these new algorithms.

1 Introduction

1.1 Background

Computing sample points of connected components of real solutions to real polynomial systems with constraints is a basic subroutine of real algebraic geometry. It allows in particular to decide the emptiness of the solution set of such a system, a problem that

is known to be NP-hard [24, Appendix A.7]. The computation of these sample points also finds applications in fields such as robotics [12], biology [22], optimisation [23], program verification [31], or combinatorics [36]. There is therefore a need for efficient algorithms, both in theory and in practice, for computing these sample points. Moreover, as examples arising from applications are typically structured, having for instance symmetries, invariance, multi- or weighted-homogeneity, these algorithms should make use of these structures to improve computations. In this paper, we focus on the particular case of semi-algebraic sets defined by a single polynomial inequation: sets of the form $S := \{\mathbf{x} \in \mathbb{R}^n \mid f(\mathbf{x}) \neq 0\}$, where f is a polynomial with rational coefficients in n variables.

Given such a polynomial f of degree d , the number of connected components of S is known to be at most $O(d)^n$, by the *Oleinik–Petrovski–Thom–Milnor bound* (original works in [47, 49, 63], see [5] for a modern statement). In particular, any algorithm computing at least one point per connected component of S will have complexity at least exponential in n .

This problem can be seen as an instance of *real quantifier elimination*, and as such is known to be solvable algorithmically since Tarski [62]. The first algorithm whose complexity is elementary recursive to solve this problem is the *cylindrical algebraic decomposition* (CAD) given by Collins [13], and performs $d^{2^{O(n)}}$ arithmetic operations. Moreover, if the bitsize of the coefficients of f is at most τ , the CAD performs $\tau d^{2^{O(n)}}$ bit operations [5, 7]. The CAD is a classical algorithm for solving quantifier elimination problems, and is available in various computer algebra systems, such as **Maple** [45] and **Mathematica** [37]. However, its doubly exponential complexity in n makes it impractical to use when n is large; for instance, on examples of fixed degree 4 and small bitsize, its runtime becomes prohibitive whenever n exceeds 4.

This doubly exponential complexity in the number of variables contrasts with the Oleinik–Petrovski–Thom–Milnor bound, which is singly exponential in n . To bridge this gap, the *critical point* method was developed [32, 33, 35], and eventually resulted in Basu, Pollack and Roy’s algorithm [5, Theorem 13.22], which performs $d^{O(n)}$ arithmetic operations, and $\tau d^{O(n)}$ bit operations, and is thus polynomial in the Oleinik–Petrovski–Thom–Milnor bound. However, the constant hidden by the big-Oh notation is actually too large for practical implementation [2], due to the manipulation of infinitesimals, and it does not make use of any structure inherent to the input system.

To resolve this, new, *probabilistic*, algorithms have been developed [3, 4, 42, 43, 53, 55], where a random change of variables is performed to ensure certain properties with high probability. On one hand, one could reduce the problem of computing sample points in a semi-algebraic set to one in an *algebraic* set, by introducing a new variable x_{n+1} and taking $x_{n+1}f - 1$ as input. In this case, using \tilde{O} to denote the omission of logarithmic factors, the state-of-the-art complexity result is in $\tilde{O}(\log(1/\epsilon)(\tau + \log(1/\epsilon))d^{3n+6})$ bit operations [18, 19, 55], where $0 < \epsilon < 1$ is an input parameter controlling the probability of success. Note that this result is only applicable on inputs satisfying some regularity

assumption. On the other hand, algorithms specific to the semi-algebraic case have also been developed, yielding a state-of-the-art complexity result of $\tilde{O}\left(\binom{n+d}{d}8^n d^{2n+1}\right)$ arithmetic operations [43], without analysis of the bit complexity. Both these approaches enjoy better practical results than the CAD.

In the following paragraphs, we recall the notion of height (a measure for the bit size of polynomials), as well as certain data structures used to represent polynomials with rational coefficients and finite sets of algebraic points, which are needed to express our main result.

1.2 Bit size and data structures

Throughout this paper, we use \log to denote logarithms in base 2.

Height. We define the (logarithmic) *height* of a non-zero rational number $q = u/v$ as $\text{ht}(q) := \max(\log|u|, \log(v))$, where $u \in \mathbb{Z}$, $v \in \mathbb{N} \setminus \{0\}$, and $\gcd(u, v) = 1$, and formally define $\text{ht}(0) := 0$. For a non-zero polynomial $g \in \mathbb{Q}[x_1, \dots, x_n]$, if $v \in \mathbb{N} \setminus \{0\}$ is the minimal common denominator of all non-zero coefficients of g , then the *height* of the polynomial g is defined as $\text{ht}(g) := \max(\log(v), \log\|vg\|_\infty)$, where $\|\cdot\|_\infty$ denotes the maximum of the absolute values of the coefficients of the polynomial. Knowing the height of a polynomial with rational coefficients, along with its degree, allows us to obtain an upper bound on the size of its bit representation.

Straight-line programs. Given polynomials $g_1, \dots, g_r \in \mathbb{Q}[x_1, \dots, x_n]$, we say that a sequence Γ of elementary operations $+, -, \times$ is a *straight-line program evaluating the polynomials* g_1, \dots, g_r if Γ evaluates g_1, \dots, g_r from the input variables x_1, \dots, x_n . The *length*, or *size*, of a straight-line program is the number of elementary operations it performs. This representation of polynomials is commonly used in polynomial system solving [3, 27–30, 38, 54, 55].

Zero-dimensional rational parametrisations. Suppose that we are given a zero-dimensional algebraic set $Q \subset \overline{\mathbb{Q}}^n$ defined over \mathbb{Q} (that is, a finite set of algebraic numbers). A *zero-dimensional rational parametrisation* $\mathbb{Q} = (w, v_1, \dots, v_n, \nu)$ of Q consists of univariate polynomials $w, v_1, \dots, v_n \in \mathbb{Q}[t]$ for a new variable t , and a \mathbb{Q} -linear form ν in n variables, such that:

- w is monic, squarefree, and $\deg(w) > \deg(v_i)$ for all i ,
- $\nu(v_1, \dots, v_n) \equiv tw' \pmod{w}$,
- and $Q = \left\{ \left(\frac{v_1(t)}{w'(t)}, \dots, \frac{v_n(t)}{w'(t)} \right) \in \mathbb{C}^n \mid w(t) = 0 \right\}$.

The constraint on ν says that the roots of w are precisely the values taken by ν on the points of Q , and implies that ν takes distinct values at distinct points of Q ; it is therefore referred to as a *separating* linear form *associated* to \mathbb{Q} . This data structure is a staple of

effective algebra, being first introduced by Kronecker and Macaulay [41, 44] and having been used in many algorithms since [1, 26, 28–30, 51, 52, 54, 55].

Note that, although the denominators w' are not strictly necessary to represent the sets in this way, they allow for precise control of the heights of the polynomials of the parametrisation [1, 30, 52], which we exploit in this work. By convention, we denote by $\mathbb{Q} = (w \equiv 1)$ the parametrisation that represents the empty set. Moreover, we define the *degree* of a parametrisation as the degree of its defining polynomial w .

1.3 Main result

Our first contribution to the problem is a new Monte-Carlo algorithm for computing points per connected components of semi-algebraic sets defined by a single polynomial inequation, in the case where the hypersurface defined by the vanishing of this polynomial is smooth.

Our algorithm relies on the use of efficient multi-homogeneous polynomial system solving techniques [54] for solving zero-dimensional polynomial systems, and thus inherits their probabilistic aspects. As such, our algorithm implicitly relies on an *oracle* \mathbb{O} , which takes as input a positive integer b , and returns a prime number in the set $\{b + 1, \dots, 2b\}$ uniformly at random among the primes in this interval. Such an oracle exists, see for example [25, Section 18.4].

We adapt the algorithm from [55] to compute points on the boundaries of each connected component of the semi-algebraic set S under study, and obtain from them parametrisations of finite sets of points meeting all connected components of S , by computing well-chosen translations of these critical points. Once the parametrisations of the sample points are obtained, it suffices to compute rational approximations of those points of low bit length, but sufficiently precise to lie in the same connected component as their exact counterpart. This idea is not new (see e.g. [5, Theorem 13.16]), but we apply recent results concerning rational parametrisations (chiefly due to [46]) in order to make explicit the exponent of d .

Concerning the computation of these rational approximations, our first result concerns the case where the input polynomial f has a general multi-homogeneous structure:

Theorem 1. *Suppose that $f \in \mathbb{Q}[\mathbf{x}_1, \dots, \mathbf{x}_m]$ is squarefree, where $\mathbf{x}_i = (x_{1+\sum_{j=1}^{i-1} n_j}, \dots, x_{n_i+\sum_{j=1}^{i-1} n_j})$, $n = n_1 + \dots + n_m$, and $\deg_{\mathbf{x}_i}(f) \leq d_i$. Suppose that $d_1 \leq \dots \leq d_m$, and that f has height τ . Assume that the hypersurface $V(f) \subset \mathbb{C}^n$ is smooth, and that Γ is a straight-line program of length L evaluating f . Let $S := \{\mathbf{x} \in \mathbb{R}^n \mid f(\mathbf{x}) \neq 0\}$ and let $0 < \epsilon < 1$. Then there exists an algorithm which takes Γ and ϵ as inputs, and returns one of the following outputs:*

- either a set of at most $2n\Delta + 1$ points of \mathbb{Q}^n ,
- or fail.

With probability at least $1 - \epsilon$, the first outcome occurs, and the computed set of points intersects every connected component of S . In any case, the algorithm performs at most

$$\tilde{O}(\log(1/\epsilon)n^3\Delta\mathbf{H}(L + nd + n^3) + nd^5\Delta^3(\mathbf{H} + n\Delta))$$

bit operations, where:

- $d := \deg(f)$,
- for $1 \leq c \leq m$, $1 \leq k \leq n$, $\ell > k$, and constants $a_{i,j}$ and $b_{i,j,k}$,

$$\delta_{c,\ell,k} := \deg_{\mathbf{x}_c} \left(\frac{\partial f}{\partial x_\ell} + \sum_{i=1}^k \left(\sum_{j=1}^{n-k} a_{i,(j+k)} b_{j,\ell-k,k} \right) \frac{\partial f}{\partial x_i} \right),$$

- $\gamma \in \tilde{O}(n \log(1/\epsilon) + n^2 + \tau + d)$,
- Δ is the sum of the coefficients of the polynomial

$$(d_1\theta_1 + \cdots + d_m\theta_m) \prod_{i=2}^n (\delta_{1,i,1}\theta_1 + \cdots + \delta_{m,i,1}\theta_m) \pmod{\langle \theta_1^{n_1+1}, \dots, \theta_m^{n_m+1} \rangle},$$

which is bounded from above by $d_1^{n_1} \cdots d_m^{n_m} \frac{n!}{n_1! \cdots n_m!}$,

- \mathbf{H} is the sum of the coefficients of the polynomial

$$\begin{aligned} & (\gamma\zeta + d_1\theta_1 + \cdots + d_m\theta_m) \prod_{i=2}^n (\gamma\zeta + \delta_{1,i,1}\theta_1 + \cdots + \delta_{m,i,1}\theta_m) \\ & \pmod{\langle \zeta^2, \theta_1^{n_1+1}, \dots, \theta_m^{n_m+1} \rangle}. \end{aligned}$$

The oracle \mathbb{O} is called n times. Upon success, every rational coordinate in the output has height in

$$\tilde{O}(d^5\Delta^2(\mathbf{H} + n\Delta)).$$

Note that the squarefreeness of f is a mild assumption, as the non-vanishing of a polynomial defines the same semi-algebraic set as the non-vanishing of its squarefree part. As $\Delta \leq d_1^{n_1} \cdots d_m^{n_m} \frac{n!}{n_1! \cdots n_m!}$, the complexity result stated above can be subexponential in n in some cases, for instance when some of the d_i 's are equal to 1.

Our second result is a consequence of the first one, when the input polynomial f does not have any particular associated multi-degree structure.

Theorem 2. *Suppose that $f \in \mathbb{Q}[x_1, \dots, x_n]$ is squarefree, of degree d and height τ , and that $\deg\left(\frac{\partial f}{\partial x_1}\right) \leq \cdots \leq \deg\left(\frac{\partial f}{\partial x_n}\right)$. Suppose that the hypersurface $V(f) \subset \mathbb{C}^n$ is smooth.*

Let $S := \{\mathbf{x} \in \mathbb{R}^n \mid f(\mathbf{x}) \neq 0\}$. Suppose that Γ is a straight-line program of length L evaluating f . Let $0 < \epsilon < 1$, and define

$$\mathbf{D} := d \times \deg\left(\frac{\partial f}{\partial x_2}\right) \times \cdots \times \deg\left(\frac{\partial f}{\partial x_n}\right).$$

Then there exists an algorithm which takes Γ and ϵ as inputs, and returns one of the following outputs:

- either a set of at most $2n\mathbf{D} + 1$ points of \mathbb{Q}^n ,
- or fail.

With probability at least $1 - \epsilon$, the first outcome occurs, and the computed set of points intersects every connected component of S . In any case, the algorithm performs at most

$$\tilde{O}(n^2 \mathbf{D}^2 (n \log(1/\epsilon) + n^2 + \tau + d)(\log(1/\epsilon)n^2(L + nd + n^3) + d^5 \mathbf{D}^2))$$

bit operations. The oracle \mathbb{O} is called n times. Upon success, every rational coordinate in the output has height in

$$\tilde{O}(nd^5 \mathbf{D}^3 (n \log(1/\epsilon) + n^2 + \tau + d)).$$

The assumption on the degrees of the partial derivatives of f is also mild, since it can always be satisfied by re-labelling the variables.

Our third main result concerns the complexity of computing only the rational parametrisations of the points per connected components of S , without computing the rational approximations, which yields the following result:

Theorem 3. *Under the notations of Theorem 1 and Theorem 2, there exists an algorithm which takes Γ and ϵ as inputs, and returns one of the following outputs:*

- either a point of \mathbb{Q}^n and $2n$ zero-dimensional rational parametrisations,
- or fail.

With probability at least $1 - \epsilon$, the first outcome occurs, and the union of the point and the zeroes of the parametrisations contains at least one point per connected component of S . In any case, the algorithm performs at most

$$\tilde{O}(\log(1/\epsilon)n^3 \mathbf{\Delta} \mathbf{H} (L + nd + n^3) + nd^3 \mathbf{\Delta}^2 (\mathbf{H} + n\mathbf{\Delta}))$$

bit operations in the multi-homogeneous setting, or

$$\tilde{O}(n^2 \mathbf{D}^2 (n \log(1/\epsilon) + n^2 + \tau + d)(\log(1/\epsilon)n^2(L + nd + n^3) + d^3 \mathbf{D}))$$

bit operations in the setting of Theorem 2. The oracle \mathbb{O} is called n times. Upon success, every polynomial in each parametrisation has degree at most $\mathbf{\Delta}$, respectively \mathbf{D} , and height in $\tilde{O}(d^3 \mathbf{\Delta} (\mathbf{H} + n\mathbf{\Delta}))$, respectively $\tilde{O}(nd^3 \mathbf{D}^2 (n \log(1/\epsilon) + n^2 + \tau + d))$.

Moreover, the quantities $\mathbf{\Delta}$ and \mathbf{D} are (multi-homogeneous) Bézout bounds on the number of complex solutions to a polynomial system involving f and its partial derivatives. As such, these quantities are bounded above by $d(d-1)^{n-1}$, and [Theorems 1 to 3](#) imply that the bit complexity of computing rational parametrisations of points per connected components of S is essentially cubic in $d(d-1)^{n-1}$, whereas the bit complexity of computing the actual rational sample points is essentially quartic in $d(d-1)^{n-1}$.

We have implemented this algorithm and observed practical improvements over the state-of-the-art algorithms on several examples. Notably, on examples where the structure can be leveraged, e.g. where \mathbf{D} is much lower than $d(d-1)^{n-1}$, we observe competitive timings even compared to state-of-the-art numerical solvers. See [Section 7](#) for more details.

The results used to show the correctness of our algorithm are, due to its nature, inherently geometric, and are interesting on their own. For this reason, we have grouped all such results in [Section 4](#).

1.4 Organisation of the paper

We start by fixing notation and recalling basic notions in [Section 2](#). We then describe the algorithm and its subroutines in [Section 3](#). [Section 4](#) provides proofs of correctness of these subroutines and of the main algorithm, `GENERICPOINTPERCONNECTEDCOMPONENT`. We analyse the probability of success of the algorithm in [Section 5](#), and then analyse its bit cost in [Section 6](#). Finally, [Section 7](#) is devoted to practical implementations.

2 Notation and preliminaries

2.1 Basic notions

We begin by recalling some basic algebraic and geometric notions which we use in this paper. Those are standard, and we refer interested readers to [\[14, 17, 48, 59\]](#) for more details.

Algebraic sets. Let \mathbb{K} be a field and $\overline{\mathbb{K}}$ denote its algebraic closure. A \mathbb{K} -*algebraic set* $W \subset \overline{\mathbb{K}}^n$ is the set of common solutions, in $\overline{\mathbb{K}}^n$, to n -variate polynomial equations, where those polynomials have coefficients in \mathbb{K} . When the base field is clear from context, we simply refer to those sets as *algebraic sets*. If explicit polynomials $g_1, \dots, g_s \in \mathbb{K}[x_1, \dots, x_n]$ are given, we denote by $V(g_1, \dots, g_s) \subset \overline{\mathbb{K}}^n$ the algebraic set defined by $g_1 = \dots = g_s = 0$. Conversely, the *ideal*, in $\mathbb{K}[x_1, \dots, x_n]$, associated to a set $W \subset \overline{\mathbb{K}}^n$ is the set of n -variate polynomials that vanish on all points of W , and is denoted $I(W)$.

An algebraic set $W \subset \overline{\mathbb{K}}^n$ is said to be *irreducible* when $W = W_1 \cup W_2$, with W_1, W_2 algebraic sets, implies $W = W_1$ or $W = W_2$. Any algebraic set W can be decomposed

into a finite union of irreducible algebraic sets, called the *irreducible components* of W , which are uniquely defined up to order. An algebraic set is said to be a *hypersurface* when it is defined by a single polynomial.

For an algebraic set W , its *dimension*, denoted $\dim(W)$, is the Krull dimension of the coordinate ring of W . By convention, we set $\dim(\emptyset) := -1$; zero-dimensional algebraic sets are precisely non-empty finite algebraic sets. An algebraic set $W = V(g_1, \dots, g_s)$ is said to be *d-equidimensional* when all its irreducible components have the same dimension d .

Zariski topology and genericity. The *Zariski topology* on \mathbb{K}^n is a topology where the closed sets are of the form $V(F) \cap \mathbb{K}^n$, where F is any set of n -variate polynomials with coefficients in \mathbb{K}^n . The *Zariski closure* of some set $X \subset \mathbb{K}^n$, denoted \overline{X}^Z , is the smallest Zariski closed set containing X . Note that this is typically different from the Euclidean closure \overline{X} .

A property P depending on parameters $\mathbf{p} \in \mathbb{K}^n$ is said to be *generic* if there exists a non-empty Zariski open set $U \subseteq \mathbb{K}^n$ such that $P(\mathbf{p})$ is true for all $\mathbf{p} \in U$.

Quantitative aspects. The *degree* of a zero-dimensional algebraic set is its cardinality. For an irreducible algebraic set W , its *degree* (in the sense of [34]), denoted $\deg(W)$, is the finite number of points contained in the intersection of W with $\dim(W)$ generic hyperplanes. For an arbitrary algebraic set, its degree is the sum of the degrees of its irreducible components of highest dimension $\dim(W)$. If the algebraic set is a hypersurface, its degree is simply the degree of its defining polynomial, and we formally define $\deg(\emptyset) = 1$.

For a d -equidimensional algebraic set $W \subset \mathbb{C}^n$ defined over \mathbb{Q} , its *height* (in the sense of [15, 16, 40, 50]) is a measure of the bit complexity of its representation. Formally, it is defined as

$$\text{ht}(W) := \sum_{p \text{ prime}} \ell_p(C_W) + m(C_W, d+1, n+1) + (d+1) \deg(W) \sum_{i=1}^n \frac{1}{2^i},$$

where C_W denotes any Chow form of W defined over \mathbb{Q} [40, Section 1], $m(f, r, n)$ denotes the (r, n) -Mahler measure of f over the complex sphere of dimension n [40, Section 1], and ℓ_p denotes the p -adic height [40, Section 1]. As a formal definition of all these quantities would be rather technical, and not the chief topic of this paper, we refer interested readers to e.g. [15, Section 3] for a comprehensive definition.

Smoothness, projections and critical points. Assume that W is a d -equidimensional algebraic set. The *Zariski tangent space* to W at $\mathbf{x} \in W$, denoted $T_{\mathbf{x}}W \subset \overline{\mathbb{K}}^n$, is the vector space defined by

$$\mathbf{v} \in T_{\mathbf{x}}W \iff \text{grad}(g)(\mathbf{x}) \cdot \mathbf{v} = 0 \text{ for all } g \in I(W).$$

If, as a vector space, $\dim(\mathbb{T}_{\mathbf{x}}W) = d$, then \mathbf{x} is said to be a *regular* or *smooth* point of W . Otherwise, \mathbf{x} is said to be a *singular* point of W . The algebraic set W is said to be *smooth* if it does not have any singular point. Furthermore, if $G = \{g_1, \dots, g_s\}$ generates $I(W)$, then at any regular point $\mathbf{x} \in W$, $\text{Jac}_{\mathbf{x}}(G)$ has rank $n - d$ and its right kernel is $\mathbb{T}_{\mathbf{x}}W$, where $\text{Jac}_{\mathbf{x}}(G)$ is the Jacobian matrix of G evaluated at \mathbf{x} .

Let $W \subset \mathbb{C}^n$ be a d -equidimensional algebraic set, and let $\phi : \mathbb{C}^n \rightarrow \mathbb{C}^m$ be a polynomial map. A point \mathbf{x} of W is said to be a *critical point* of the restriction of ϕ to W if it is a regular point of W such that $d\phi(\mathbb{T}_{\mathbf{x}}W) \neq \mathbb{C}^m$, where $d\phi$ denotes the *differential map* of the restriction of ϕ to W , defined as $d\phi : \mathbb{T}_{\mathbf{x}}W \rightarrow \mathbb{C}^m, \mathbf{u} \mapsto \text{Jac}_{\mathbf{x}}(\phi) \cdot \mathbf{u}$.

We denote by $\text{crit}(\phi, W)$ the set of critical points of ϕ on W . A *critical value* of ϕ on W is a value $\mathbf{y} \in \mathbb{C}^m$ such that $\exists \mathbf{x} \in \text{crit}(\phi, W)$ with $\phi(\mathbf{x}) = \mathbf{y}$. As the above definition of critical points does not allow us to directly compute them, we recall the following lemma:

[56, Lemma A.2] *Let W be a d -equidimensional algebraic set, and suppose that $G = \{g_1, \dots, g_s\}$ generates $I(W)$. Let $\phi : \mathbb{C}^n \rightarrow \mathbb{C}^m$ be a polynomial map. Then*

$$\text{crit}(\phi, W) = \left\{ \mathbf{x} \in W \mid \text{rank}(\text{Jac}_{\mathbf{x}}(G)) = n - d, \quad \text{rank} \begin{bmatrix} \text{Jac}_{\mathbf{x}}(G) \\ \text{Jac}_{\mathbf{x}}(\phi) \end{bmatrix} < n - d + m \right\}.$$

Throughout this paper, we chiefly focus on critical points of projection maps onto some coordinates, which we denote as follows, for $1 \leq i \leq n$:

$$\pi_i : \mathbb{C}^n \rightarrow \mathbb{C}^i, \quad \pi_i(x_1, \dots, x_n) = (x_1, \dots, x_i).$$

Change of variables. For a matrix $\mathbf{A} \in \mathbb{K}^{n \times n}$ and a polynomial $g \in \mathbb{K}[x_1, \dots, x_n]$, we define

$$g^{\mathbf{A}} := g(\mathbf{A}\mathbf{X}) \in \mathbb{K}[x_1, \dots, x_n],$$

where \mathbf{X} denotes the column vector $(x_1, \dots, x_n)^{\text{T}}$, where the T superscript denotes transpose. Similarly, if $G = \{g_1, \dots, g_s\}$, we define $G^{\mathbf{A}} := (g_1^{\mathbf{A}}, \dots, g_s^{\mathbf{A}})$. Finally, for an algebraic set $W \subset \overline{\mathbb{K}}^n$ and \mathbf{A} invertible, we define

$$W^{\mathbf{A}} := \{ \mathbf{A}^{-1}\mathbf{x} \mid \mathbf{x} \in W \}.$$

Note that this notation is consistent, in the sense that $V(G^{\mathbf{A}}) = V(G)^{\mathbf{A}}$.

Part of the randomness of the algorithm comes from choosing a “suitable” \mathbf{A} to change variables before performing computations. We show that part of this “suitability” arises from the very structure of \mathbf{A} , and as such, we introduce notation to denote the parts of \mathbf{A} of interest. We denote by $a_{i,j}$ the entries of \mathbf{A} . For $0 \leq k \leq n$, we define the matrices C_k and B_k as the right blocks of \mathbf{A} of sizes $k \times (n - k)$ and $(n - k) \times (n - k)$ respectively:

$$\mathbf{A} = \begin{bmatrix} * & C_k \\ * & B_k \end{bmatrix}.$$

Finally, we denote by $b_{i,j,k}$ the (i,j) -th entry of B_k^{-1} , when it exists. In particular, $b_{i,j,0}$ denotes the (i,j) -th entry of \mathbf{A}^{-1} , when it exists.

We also sometimes need to consider the entries of the change-of-variables matrix as indeterminates. When this is the case, we refer to those variables as \mathfrak{A} , and a polynomial in those variables with coefficients in \mathbb{K} as an element of $\mathbb{K}[\mathfrak{A}]$.

2.2 Notation and assumptions

Throughout this paper, we denote by $f \in \mathbb{Q}[x_1, \dots, x_n]$ the input polynomial defining the semi-algebraic set $S := \{\mathbf{x} \in \mathbb{R}^n \mid f(\mathbf{x}) \neq 0\}$. We denote by $V := V(f) \subset \mathbb{C}^n$ the hypersurface defined by the vanishing of f .

The algorithm requires the use of an arbitrary point of \mathbb{Q}^{n-1} , which is denoted $\boldsymbol{\sigma}$ in the remainder of this paper. We sometimes need to consider the coordinates of $\boldsymbol{\sigma}$ as indeterminates. When this is the case, we denote them \mathfrak{S} .

There are four key assumptions that frequently appear in our statements. To simplify notation, we define and name them here as follows:

- **(S)**: The hypersurface V is smooth, and f is squarefree.
- **(A1)**: We have $\mathbf{A} \in \text{GL}_n(\mathbb{C}) \setminus \mathcal{Z}$, where \mathcal{Z} is the proper Zariski closed set of [55, Theorem 1]. This assumption is here to guarantee that the computed critical points are in finite number.
- **(A2)**: The matrices B_k are invertible for all $1 \leq k \leq n$. This assumption is here to guarantee that the polynomials determining the critical points we compute are well-defined.
- **(B)**: We have $\boldsymbol{\sigma} \in \mathbb{C}^{n-1} \setminus \mathcal{V}$, where \mathcal{V} is the proper Zariski closed set of [19, Theorem 2.2]. This assumption is here to guarantee that the smoothness of the boundary of the semi-algebraic set S is preserved under instantiation of variables to these values.

Assumption **(S)** is the central assumption of this paper. We require f to be squarefree for the ideal generated by f to be radical, in order to apply [56, Lemma A.2] for critical points computations. As stated before, this is a mild assumption, as both f and its squarefree part define the same semi-algebraic set S .

Assumptions **(A1)**, **(A2)** and **(B)**, on the other hand, explicitly state the sufficiently generic conditions that \mathbf{A} and $\boldsymbol{\sigma}$ need to satisfy in order to be considered “suitable” for our algorithm. We show that these conditions suffice for the smooth-case algorithm in Section 4, as well as analyse the probability that randomly chosen $\boldsymbol{\sigma}$ and \mathbf{A} with bounded entries satisfy those conditions, in Section 5.

3 Algorithm

We start with a brief overview of the main algorithm. It takes as inputs a straight-line program Γ evaluating $f \in \mathbb{Q}[x_1, \dots, x_n]$ satisfying **(S)**, and a real number ϵ between 0 and 1, which allows to control the probability of success of the algorithm. The parameter ϵ comes into play when choosing random integers for the change of coordinates and the specialisation values: they are chosen uniformly at random within a set whose size is a multiple of ϵ^{-1} . Therefore, the smaller ϵ is, the larger the bitsize of the quantities involved, but the lower the chance of failure. The algorithm is based on four key steps:

1. Project $V \cap \mathbb{R}^n$ on a randomly chosen line, such that the critical points of this projection are finite, and such that the image of $V \cap \mathbb{R}^n$ under this projection is closed.

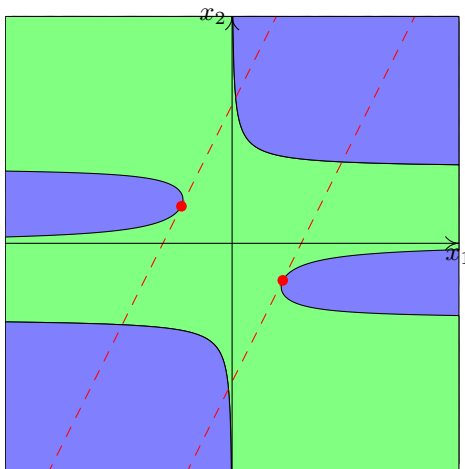


Figure 1: Illustration of the first step of algorithm applied to $f = 4x_1(x_2^3 - x_2) - 1$

2. Compute a rational parametrisation of these critical points, and obtain “sufficiently precise” isolation intervals for these points, in a transverse direction to $V \cap \mathbb{R}^n$.

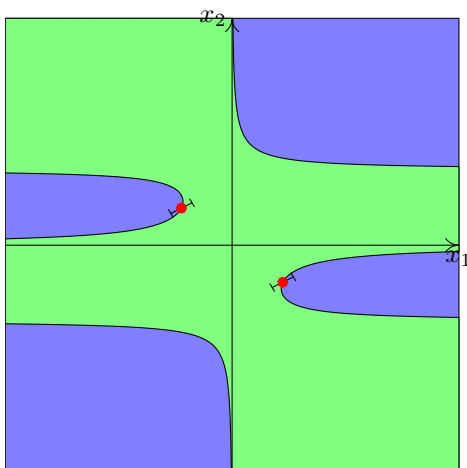


Figure 2: Illustration of the second step of algorithm applied to $f = 4x_1(x_2^3 - x_2) - 1$

3. For each endpoint of these intervals, compute a sufficiently small n -dimensional box around them, such that f never vanishes at any point in the box. Pick one rational point in each such box.

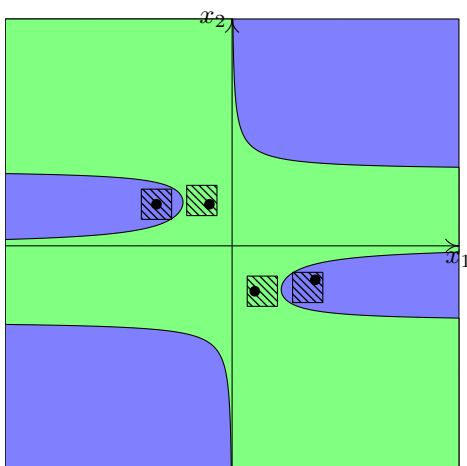


Figure 3: Illustration of the third step of algorithm applied to $f = 4x_1(x_2^3 - x_2) - 1$ (Exaggerated box size for visual clarity)

4. Instantiate x_1 to a randomly chosen value, say σ_1 , in f and perform the above steps with the newly obtained polynomial as input.

We now introduce the subroutines we use in the main algorithm.

- **Polar** takes as inputs the straight-line program Γ evaluating f , a matrix $\mathbf{A} \in \text{GL}_n(\mathbb{Q})$ satisfying **(A2)**, an integer $1 \leq k \leq n$ and the specialisation point $\sigma \in \mathbb{Q}^{n-1}$. It outputs

a straight-line program evaluating the polynomials \mathcal{P}_k , defined as

$$\mathcal{P}_k := \left(\sum_{i=1}^n b_{1,i,0} x_i - \sigma_1, \dots, \sum_{i=1}^n b_{k-1,i,0} x_i - \sigma_{k-1}, f, \right. \\ \left. \frac{\partial f}{\partial x_{k+1}} + \sum_{i=1}^k \left(\sum_{j=1}^{n-k} a_{i,(j+k)} b_{j,1,k} \right) \frac{\partial f}{\partial x_i}, \dots, \frac{\partial f}{\partial x_n} + \sum_{i=1}^k \left(\sum_{j=1}^{n-k} a_{i,(j+k)} b_{j,n-k,k} \right) \frac{\partial f}{\partial x_i} \right)$$

where $a_{i,j}$ denotes the entries of \mathbf{A} and $b_{i,j,k}$ of B_k^{-1} .

• **ZeroDimParam** takes as inputs a straight-line program evaluating a zero-dimensional polynomial system and $k \in \mathbb{N} \setminus \{0\}$. When $k = 1$, it outputs a zero-dimensional rational parametrisation of its regular solutions with probability of success greater than or equal to $1 - (11/32)$. Otherwise, it either outputs a parametrisation of a proper subset of the solutions, or **fail**. When $k > 1$, it repeats the procedure k times, and hence has a probability of success greater than or equal to $1 - (11/32)^k$, since bad outputs can be ruled out by their degree. Such a subroutine can be found in [54, Algorithm 2].

Subroutine 1 CRITICALPOINTS($\Gamma, \mathbf{A}, k, \sigma, r$)

Input: $\Gamma, \mathbf{A}, k, \sigma, r$, where Γ is a straight-line program evaluating $f \in \mathbb{Q}[x_1, \dots, x_n]$, $\mathbf{A} \in \mathbb{Q}^{n \times n}$, $1 \leq k \leq n$, $\sigma \in \mathbb{Q}^{n-1}$ and $r \in \mathbb{N} \setminus \{0\}$.

Requirement: f, \mathbf{A} and σ satisfy **(S)**, **(A1)**, **(A2)** and **(B)**.

Output: A zero-dimensional rational parametrisation $\mathcal{Q} = (w, \mathbf{v}, \nu)$.

- 1: $\Lambda \leftarrow \text{Polar}(\Gamma, \mathbf{A}, k, \sigma)$;
 - 2: $\mathcal{Q} \leftarrow \text{ZeroDimParam}(\Lambda, r)$;
 - 3: **return** \mathcal{Q}
-

• **ExtensionIsolation** takes as inputs a univariate polynomial $p \in \mathbb{Q}[t]$ and a bivariate polynomial $q \in \mathbb{Q}[t, u]$. Denoting by $t_1 < \dots < t_j$ the real algebraic roots of p and by $u_{i,1} < \dots < u_{i,\ell_i}$ the real roots of $q(t_i, u)$, and assuming $q(t_i, u)$ is squarefree for all i , it outputs j lists consisting, for any $1 \leq i \leq j$, of ℓ_i disjoint intervals with rational endpoints $[q_{i,k}^-, q_{i,k}^+]$ such that $u_{i,k} \in [q_{i,k}^-, q_{i,k}^+]$.

Subroutine 2 ISOLATION($\Gamma, \mathbf{a}, \mathbb{Q}$)

Input: $\Gamma, \mathbf{a}, \mathbb{Q}$, where Γ is a straight-line program evaluating $f \in \mathbb{Q}[x_1, \dots, x_n]$, $\mathbf{a} \in \mathbb{Q}^n$, and $\mathbb{Q} = (w, \mathbf{v}, \nu)$ is a zero-dimensional rational parametrisation, where $\mathbf{v} = (v_1, \dots, v_n)$.

Requirement: f satisfies **(S)**, $w \neq 1$, and for any ξ parametrised by \mathbb{Q} , $\xi \in V$ and the line parametrised by $\xi + \mathbf{a}u$ does not intersect V tangentially at ξ .

Output: Two zero-dimensional rational parametrisations \mathcal{P}^- and \mathcal{P}^+ .

- 1: $d \leftarrow \deg(f)$;
 - 2: $\mathcal{F} \leftarrow \text{ExtensionIsolation}\left(w(t), (w'(t))^d f\left(\frac{1}{w'(t)}\mathbf{v}(t) + u\mathbf{a}\right)\right)$;
 - 3: $\lambda \leftarrow \min\{1, \min\{|e| : e \in \{\text{endpoints of each interval in } \mathcal{F}\} \wedge e \neq 0\}\}/2$;
 - 4: $\mathcal{P}^- \leftarrow (w, \mathbf{v} - \lambda w' \mathbf{a}, \nu)$;
 - 5: $\mathcal{P}^+ \leftarrow (w, \mathbf{v} + \lambda w' \mathbf{a}, \nu)$;
 - 6: **return** $\mathcal{P}^-, \mathcal{P}^+$
-

• **LowerBoundValue** & **UpperBoundValue** both take as inputs a zero-dimensional rational parametrisation \mathbb{Q} and a polynomial f with rational coefficients. They respectively output a positive lower and upper bound on the non-zero absolute values the polynomial f takes at any real solution to the parametrisation \mathbb{Q} .

• **ParamApproximation** takes as inputs a zero-dimensional rational parametrisation \mathbb{Q} and an integer r , and outputs rational approximations of all real solutions to \mathbb{Q} , whose coordinates are accurate to precision 2^{-r} .

Subroutine 3 APPROXIMATION(Γ, \mathbb{Q})

Input: Γ, \mathbb{Q} , where Γ is a straight-line program evaluating $f \in \mathbb{Q}[x_1, \dots, x_n]$, and $\mathbb{Q} = (w, \mathbf{v}, \nu)$ is a zero-dimensional rational parametrisation, where $\mathbf{v} = (v_1, \dots, v_n)$.

Requirement: f is non-zero on every point parametrised by \mathbb{Q} .

Output: A finite subset \mathcal{A} of \mathbb{Q}^n .

- 1: $d \leftarrow \deg(f)$;
 - 2: $c \leftarrow \|f\|_\infty$;
 - 3: $m \leftarrow \text{LowerBoundValue}(\mathbb{Q}, f)$;
 - 4: $\mathcal{M} \leftarrow \max_{1 \leq i \leq n} \text{UpperBoundValue}(\mathbb{Q}, x_i) + 1$;
 - 5: $d \leftarrow \min\left\{1, m \left(\binom{n+d}{d} \mathcal{M}^d c\right)^{-1}\right\}$;
 - 6: $\mathcal{A} \leftarrow \text{ParamApproximation}(\mathbb{Q}, \lceil -\log(d) \rceil)$;
 - 7: **return** \mathcal{A}
-

We can now describe the main algorithm, **Algorithm 1**. The algorithm essentially reduces to three subroutines; **CRITICALPOINTS**, which computes a parametrisation of the critical points; **ISOLATION**, which computes the endpoints of the isolation intervals, and **APPROXIMATION**, which computes sufficiently precise rational approximations.

Algorithm 1 GENERICPOINTPERCONNECTEDCOMPONENT(Γ, ϵ)

Input: Γ, ϵ where Γ is a straight-line program evaluating $f \in \mathbb{Q}[x_1, \dots, x_n]$, and $0 < \epsilon < 1$.

Requirement: f satisfies **(S)**.

Output: A finite subset of \mathbb{Q}^n .

```
1:  $d \leftarrow \deg(f)$ ;
2: Build  $X := \left\{1, \dots, \left\lceil 3\epsilon^{-1} \left(5n^3(2d)^{2n} + \frac{n^2-n}{2}\right)\right\rceil\right\}$  and draw  $\mathbf{A} \in X^{n \times n}$  at random;
3: Build  $Y := \{1, \dots, \lceil 3\epsilon^{-1}nd^{2n} \rceil\}$  and draw  $\boldsymbol{\sigma} \in Y^{n-1}$  at random;
4: for  $k \in \{1, \dots, n\}$  do
5:    $\mathbb{Q}_k = (w_k, \mathbf{v}_k, \nu_k) \leftarrow \text{CRITICALPOINTS}(\Gamma, \mathbf{A}, k, \boldsymbol{\sigma}, \lceil \log(3n\epsilon^{-1}) \rceil)$ ;
6:   if  $w_k \neq 1$  then
7:      $\mathcal{P}_k^-, \mathcal{P}_k^+ \leftarrow \text{ISOLATION}(\Gamma, (a_{1,k}, \dots, a_{n,k}), \mathbb{Q}_k)$ ;
8:      $\mathcal{A}_k^- \leftarrow \text{APPROXIMATION}(\Gamma, \mathcal{P}_k^-)$ ;
9:      $\mathcal{A}_k^+ \leftarrow \text{APPROXIMATION}(\Gamma, \mathcal{P}_k^+)$ ;
10:     $\mathcal{A}_k \leftarrow \mathcal{A}_k^- \cup \mathcal{A}_k^+$ ;
11:   else
12:      $\mathcal{A}_k \leftarrow \emptyset$ ;
13: return  $\bigcup_{k=1}^n \mathcal{A}_k \cup \{\mathbf{A}(\sigma_1, \dots, \sigma_{n-1}, 0)^\top\}$ 
```

In the illustrated example, the first three steps correspond to [Algorithm 1](#), and the last step corresponds to the **for** loop of [Algorithm 1](#).

4 Correctness

4.1 Correctness of **CRITICALPOINTS**

In this subsection, we show that, under assumptions **(S)**, **(A1)**, **(A2)** and **(B)**, **CRITICALPOINTS** correctly outputs a zero-dimensional rational parametrisation \mathbb{Q} which encodes the sought-for critical points.

We begin by some intuitive geometric results on the boundary of a connected component of the semi-algebraic set S .

Lemma 4.1.1. *For any non-isolated point $\mathbf{y} \in V \cap \mathbb{R}^n$, there exists a neighbourhood \mathcal{N}_1 of \mathbf{y} such that $\mathcal{N}_1 \cap V \cap \mathbb{R}^n$ is connected.*

Proof. Let us fix a non-isolated $\mathbf{y} \in V \cap \mathbb{R}^n$. Then, by the local conic structure theorem [\[10, Theorem 9.3.6\]](#), there exists $e \in \mathbb{R}$, $e > 0$, and a semi-algebraic homeomorphism ϕ such that:

- $\phi : \overline{B}_n(\mathbf{y}, e) \rightarrow \overline{B}_n(\mathbf{y}, e)$, where $\overline{B}_n(\mathbf{y}, e)$ denotes the closed n -dimensional ball of radius e centred at \mathbf{y} ,
- $\|\phi(\mathbf{x}) - \mathbf{y}\| = \|\mathbf{x} - \mathbf{y}\|$ for every $\mathbf{x} \in \overline{B}_n(\mathbf{y}, e)$,

- ϕ restricted to $S^{n-1}(\mathbf{y}, e)$ is the identity map, where $S^{n-1}(\mathbf{y}, e)$ denotes the boundary of $\overline{B}_n(\mathbf{y}, e)$,
- and $\phi^{-1}(V \cap \mathbb{R}^n \cap \overline{B}_n(\mathbf{y}, e))$ is the cone with vertex \mathbf{y} and basis $V \cap \mathbb{R}^n \cap S^n(\mathbf{y}, e)$.

In particular, since $\phi^{-1}(V \cap \mathbb{R}^n \cap \overline{B}_n(\mathbf{y}, e))$ is a cone, it is connected. Moreover, since homeomorphisms preserve connectivity, it follows that $\phi(\phi^{-1}(V \cap \mathbb{R}^n \cap \overline{B}_n(\mathbf{y}, e))) = V \cap \mathbb{R}^n \cap \overline{B}_n(\mathbf{y}, e)$ is also connected. Letting $\mathcal{N}_1 := \overline{B}_n(\mathbf{y}, e)$ finishes the proof. \square

Lemma 4.1.2. *Suppose that f satisfies **(S)**, and let $S^+ := \{\mathbf{x} \in \mathbb{R}^n \mid f(\mathbf{x}) > 0\}$. Then, for any $\mathbf{y} = (y_1, \dots, y_n) \in V \cap \mathbb{R}^n$, there exists a neighbourhood \mathcal{N}_2 of \mathbf{y} such that $\mathcal{N}_2 \cap S^+$ is connected.*

Proof. Since f satisfies **(S)**, the gradient of f at \mathbf{y} is non-zero, and hence there exists $1 \leq i \leq n$ such that its i -th coordinate is non-zero. Without loss of generality, we assume for simplicity that $i = n$.

Then, by the semi-algebraic Implicit Function Theorem [10, Corollary 2.9.8], there exist open semi-algebraic neighbourhoods $U_1 \subseteq \mathbb{R}^{n-1}$ and $U_2 \subseteq \mathbb{R}$ of (y_1, \dots, y_{n-1}) and y_n respectively, and a map $\phi : U_1 \rightarrow U_2$ such that $\phi(y_1, \dots, y_{n-1}) = y_n$ and $f(\mathbf{x}) = 0 \iff x_n = \phi(x_1, \dots, x_{n-1})$ for every $\mathbf{x} \in U_1 \times U_2$. Let us now define

$$A^+ := \{\mathbf{x} \in U_1 \times U_2 \mid x_n > \phi(x_1, \dots, x_{n-1})\}.$$

Let \mathcal{N}_1 be the neighbourhood of **Theorem 4.1.1**, and let $\boldsymbol{\alpha}, \boldsymbol{\beta} \in A^+ \cap \mathcal{N}_1$ be distinct. If $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ can be path connected, then $A^+ \cap \mathcal{N}_1$ is connected (since we are working in \mathbb{R}^n), which by the definition of ϕ implies that $(U_1 \times U_2) \cap \mathcal{N}_1 \cap S^+$ is connected, and then defining $\mathcal{N}_2 := (U_1 \times U_2) \cap \mathcal{N}_1$ finishes the proof. It therefore suffices to show that $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are path connected in $A^+ \cap \mathcal{N}_1$.

By definition of \mathcal{N}_1 and by **Theorem 4.1.1**, $V \cap \mathbb{R}^n \cap \mathcal{N}_1$ is connected. Hence, there exists a path $(\gamma_1(t), \dots, \gamma_n(t))$, for $t \in [0, 1]$, from $(\alpha_1, \dots, \alpha_{n-1}, \phi(\alpha_1, \dots, \alpha_{n-1}))$ to $(\beta_1, \dots, \beta_{n-1}, \phi(\beta_1, \dots, \beta_{n-1}))$. We then consider the path $(\gamma_1(t), \dots, \gamma_{n-1}(t), \gamma_n(t) + \alpha_n t + \beta_n(1-t))$. It is a path from $\boldsymbol{\alpha}$ to $\boldsymbol{\beta}$, which lies in $A^+ \cap \mathcal{N}_1$, since $\gamma_n(t) + \alpha_n t + \beta_n(1-t) > \phi(\gamma_1(t), \dots, \gamma_{n-1}(t))$. Hence, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are path connected in $A^+ \cap \mathcal{N}_1$, as required. \square

Lemma 4.1.3. *Suppose that f satisfies **(S)** and let C be a connected component of S . Then, each connected component of the boundary $\overline{C} \setminus C$ of C is a connected component of the real algebraic set $V \cap \mathbb{R}^n$.*

Proof. Without loss of generality, suppose that f is positive on C , that is, C is a connected component of S^+ . Let $\mathbf{x} \in \overline{C}$, and suppose for a contradiction that $f(\mathbf{x}) < 0$. Then, there exists an open ball \mathcal{B} around \mathbf{x} such that f is negative on every point of that ball,

by continuity of f . In particular, $B \cap C = \emptyset$, which contradicts the fact that \mathbf{x} belongs to the Euclidean closure of C . Therefore, $\mathbf{x} \in \overline{C}$ implies that $f(\mathbf{x}) \geq 0$, and hence

$$\overline{C} \setminus C \subseteq \{\mathbf{x} \in \mathbb{R}^n \mid f(\mathbf{x}) \geq 0 \wedge f(\mathbf{x}) \not\equiv 0\} = V \cap \mathbb{R}^n.$$

Hence, a connected component of $\overline{C} \setminus C$ necessarily lies in $V \cap \mathbb{R}^n$. Let $B \subset V \cap \mathbb{R}^n$ be a connected component of $\overline{C} \setminus C$. Then, there exists a connected component A of $V \cap \mathbb{R}^n$ which contains B , since otherwise B itself would not be connected. To prove the claim, it thus suffices to show that $B = A$.

Let $\mathbf{x} \in B \subset A$ and $\mathbf{y} \in A$. Since A is connected, there exists a continuous path $\gamma : [0, 1] \rightarrow A$ such that $\gamma(0) = \mathbf{x}$ and $\gamma(1) = \mathbf{y}$. For any $t \in [0, 1]$, by [Theorems 4.1.1](#) and [4.1.2](#), there exists a neighbourhood $\mathcal{N}_t := \mathcal{N}_{1,t} \cap \mathcal{N}_{2,t}$ around $\gamma(t)$ such that $\mathcal{N}_t \cap V \cap \mathbb{R}^n$ and $\mathcal{N}_t \cap S^+$ are connected. We now consider the sets $\gamma([0, 1]) \cap \mathcal{N}_t$. These sets are open (as \mathcal{N}_t 's are), non-empty (as they contain $\gamma(t)$), and cover $\gamma([0, 1])$, which is compact (as $[0, 1]$ is). Hence, there exists a finite covering $\cup_{t \in \{t_0 < \dots < t_k\}} \mathcal{N}_t \cap \gamma([0, 1])$ of $\gamma([0, 1])$ with $t_0 = 0$ and $t_k = 1$. For $1 \leq i \leq k$, pick $\mathbf{z}_i \in \mathcal{N}_{t_{i-1}} \cap \mathcal{N}_{t_i} \cap \gamma([0, 1])$, and formally define \mathbf{z}_0 as \mathbf{x} and \mathbf{z}_{k+1} as \mathbf{y} . If we show that $\mathbf{z}_i \in B$ implies $\mathbf{z}_{i+1} \in B$ for all i , this implies, since $\mathbf{z}_0 = \mathbf{x} \in B$, that $\mathbf{z}_{k+1} = \mathbf{y}$ lies in B , which proves the claim. It therefore suffices to show that $\mathbf{z}_i \in B \implies \mathbf{z}_{i+1} \in B$.

Let us suppose that $\mathbf{z}_i \in B$ for some i . Then, as both $\mathcal{N}_{t_i} \cap V \cap \mathbb{R}^n$ and $\mathcal{N}_{t_i} \cap S^+$ are connected, it follows that any point of $\mathcal{N}_{t_i} \cap \gamma([0, 1])$ lies in the same connected component of the boundary of the same connected component of S^+ as \mathbf{z}_i . In particular, as $\mathbf{z}_i \in B$ by assumption, and as $\mathbf{z}_{i+1} \in \mathcal{N}_{t_i}$ by construction, it follows that $\mathbf{z}_{i+1} \in B$, as required. \square

We recall here some of the results from [\[55\]](#): [\[55, Theorem 1 and Proposition 4\]](#) *There exists a proper Zariski closed set \mathcal{Z} of $\mathrm{GL}_n(\mathbb{C})$ such that, for any $\mathbf{A} \in \mathrm{GL}_n(\mathbb{C}) \setminus \mathcal{Z}$ and $1 \leq k \leq n$, $V^{\mathbf{A}}$ is in Noether position with respect to π_k and, for any connected component C of $V^{\mathbf{A}} \cap \mathbb{R}^n$, the boundary of $\pi_k(C)$ is contained in $\pi_k(\mathrm{crit}(\pi_k, V^{\mathbf{A}}) \cap C)$.*

Lemma 4.1.4. *Suppose that f satisfies [\(S\)](#). Let B be a connected component of $V \cap \mathbb{R}^n$, and let $\mathcal{Z} \subset \mathbb{C}^{n \times n}$ be the proper Zariski closed set of [\[55, Theorem 1\]](#). Then, for any $\mathbf{A} \in \mathrm{GL}_n(\mathbb{C}) \setminus \mathcal{Z}$ and for any $1 \leq k \leq n$, $\pi_k(B^{\mathbf{A}})$ is closed for the Euclidean topology, where $B^{\mathbf{A}}$ denotes the connected component of $V^{\mathbf{A}} \cap \mathbb{R}^n$ corresponding to B after application of the change of variables \mathbf{A} .*

Proof. Since changing variables by \mathbf{A} preserves connectivity, we deduce that $B^{\mathbf{A}}$ is a connected component of $V^{\mathbf{A}} \cap \mathbb{R}^n$, and hence, for any \mathbf{A} and k as above, by [\[55, Theorem 1 and Proposition 4\]](#), we have

$$\overline{\pi_k(B^{\mathbf{A}})} \setminus \pi_k(B^{\mathbf{A}}) \subseteq \pi_k(\mathrm{crit}(\pi_k, V^{\mathbf{A}}) \cap B^{\mathbf{A}}) \subseteq \pi_k(B^{\mathbf{A}})$$

and hence $\overline{\pi_k(B^{\mathbf{A}})} \subseteq \pi_k(B^{\mathbf{A}})$, which implies that the image of $B^{\mathbf{A}}$ by π_k is closed for the Euclidean topology. \square

Note that, with our notation, the required condition on \mathbf{A} for [Theorem 4.1.4](#) is precisely that \mathbf{A} satisfies [\(A1\)](#). For the convenience of the reader, we now recall the statement of [\[19, Theorem 2.2\]](#):

[\[19, Theorem 2.2\]](#): Suppose that f satisfies [\(S\)](#), and let \mathcal{Z} be as in [Theorem 4.1.4](#). There exists a non-zero polynomial $g \in \mathbb{C}[\mathfrak{S}]$ of degree at most nd^{2n} such that, for any $\mathbf{A} \in \text{GL}_n(\mathbb{C}) \setminus \mathcal{Z}$, $1 \leq k \leq n$, and $\boldsymbol{\sigma} \in \mathbb{C}^{n-1} \setminus \mathcal{V}$, where $\mathcal{V} := \text{V}(g)$, the system of equations

$$x_1 - \sigma_1 = \cdots = x_{k-1} - \sigma_{k-1} = f^{\mathbf{A}} = \frac{\partial f^{\mathbf{A}}}{\partial x_{k+1}} = \cdots = \frac{\partial f^{\mathbf{A}}}{\partial x_n} = 0$$

has finitely many solutions, none of which are singular.

Note again that, with our notation, the required conditions on \mathbf{A} and $\boldsymbol{\sigma}$ for [\[19, Theorem 2.2\]](#) to be applicable are precisely that \mathbf{A} satisfies [\(A1\)](#) and that $\boldsymbol{\sigma}$ satisfies [\(B\)](#). By [\[56, Lemma A.2\]](#), we have that the solutions to the systems of [\[19, Theorem 2.2\]](#) are precisely the critical points of $V^{\mathbf{A}}$ under projection by π_k , for which the first $k-1$ coordinates have been instantiated to $\sigma_1, \dots, \sigma_{k-1}$, which under our notation is the set $\text{crit}(\pi_k, V^{\mathbf{A}}) \cap \pi_{k-1}^{-1}(\sigma_1, \dots, \sigma_{k-1})$. We could therefore use this system as the starting point of [Algorithm 1](#); however, we instead choose to start with the related following system:

Lemma 4.1.5. Suppose that \mathbf{A} is invertible and satisfies [\(A2\)](#), and let \mathcal{P}_k denote the polynomials obtained in the output of `Polar`. Then, for any $1 \leq k \leq n$, we have

$$\boldsymbol{\xi} \in \text{V} \left(x_1 - \sigma_1, \dots, x_{k-1} - \sigma_{k-1}, f^{\mathbf{A}}, \frac{\partial f^{\mathbf{A}}}{\partial x_{k+1}}, \dots, \frac{\partial f^{\mathbf{A}}}{\partial x_n} \right) \iff \mathbf{A}\boldsymbol{\xi} \in \text{V}(\mathcal{P}_k).$$

Proof. By the (multivariate) chain rule, we have $\frac{\partial f^{\mathbf{A}}}{\partial x_j} = \sum_{i=1}^n a_{i,j} \left(\frac{\partial f}{\partial x_i} \right)^{\mathbf{A}}$, for all $1 \leq j \leq n$. Therefore, by definition of the Jacobian matrix, we obtain $\text{Jac}_{\mathbf{A}\mathbf{x}}(f)^{\mathbf{A}} = \text{Jac}_{\mathbf{x}}(f^{\mathbf{A}})$. Let us fix $1 \leq k \leq n$. As \mathbf{A} satisfies [\(A2\)](#), B_k^{-1} exists, and hence

$$\begin{aligned} & \boldsymbol{\xi} \in \text{V} \left(\frac{\partial f^{\mathbf{A}}}{\partial x_{k+1}}, \dots, \frac{\partial f^{\mathbf{A}}}{\partial x_n} \right) \\ \iff & \frac{\partial f^{\mathbf{A}}}{\partial x_{k+1}}(\boldsymbol{\xi}) = \cdots = \frac{\partial f^{\mathbf{A}}}{\partial x_n}(\boldsymbol{\xi}) = 0 \\ \iff & \begin{bmatrix} \frac{\partial f}{\partial x_1}(\mathbf{A}\boldsymbol{\xi}) & \cdots & \frac{\partial f}{\partial x_n}(\mathbf{A}\boldsymbol{\xi}) \end{bmatrix} \begin{bmatrix} C_k \\ B_k \end{bmatrix} = [0 \quad \cdots \quad 0] \\ \iff & \begin{bmatrix} \frac{\partial f}{\partial x_1}(\mathbf{A}\boldsymbol{\xi}) & \cdots & \frac{\partial f}{\partial x_n}(\mathbf{A}\boldsymbol{\xi}) \end{bmatrix} \begin{bmatrix} C_k \\ B_k \end{bmatrix} B_k^{-1} = [0 \quad \cdots \quad 0] \\ \iff & \begin{bmatrix} \frac{\partial f}{\partial x_1}(\mathbf{A}\boldsymbol{\xi}) & \cdots & \frac{\partial f}{\partial x_n}(\mathbf{A}\boldsymbol{\xi}) \end{bmatrix} \begin{bmatrix} C_k B_k^{-1} \\ I_{n-k} \end{bmatrix} = [0 \quad \cdots \quad 0] \\ \iff & \begin{bmatrix} \frac{\partial f}{\partial x_{k+1}}(\mathbf{A}\boldsymbol{\xi}) & \cdots & \frac{\partial f}{\partial x_n}(\mathbf{A}\boldsymbol{\xi}) \end{bmatrix} + \begin{bmatrix} \frac{\partial f}{\partial x_1}(\mathbf{A}\boldsymbol{\xi}) & \cdots & \frac{\partial f}{\partial x_k}(\mathbf{A}\boldsymbol{\xi}) \end{bmatrix} C_k B_k^{-1} = [0 \quad \cdots \quad 0] \end{aligned}$$

$$\begin{aligned} \iff \mathbf{A}\boldsymbol{\xi} \in V \left(\frac{\partial f}{\partial x_{k+1}} + \sum_{i=1}^k \left(\sum_{j=1}^{n-k} a_{i,(j+k)} b_{j,1,k} \right) \frac{\partial f}{\partial x_i}, \dots, \right. \\ \left. \frac{\partial f}{\partial x_n} + \sum_{i=1}^k \left(\sum_{j=1}^{n-k} a_{i,(j+k)} b_{j,n-k,k} \right) \frac{\partial f}{\partial x_i} \right). \end{aligned}$$

On the other hand, as \mathbf{A} is invertible, $B_0^{-1} = \mathbf{A}^{-1}$ exists, and by substitution we have

$$\begin{aligned} \boldsymbol{\xi} \in V(x_1 - \sigma_1, \dots, x_{k-1} - \sigma_{k-1}) \\ \iff \mathbf{A}\boldsymbol{\xi} \in V \left(\sum_{i=1}^n b_{1,i,0} x_i - \sigma_1, \dots, \sum_{i=1}^n b_{k-1,i,0} x_i - \sigma_{k-1} \right). \end{aligned}$$

Since we also have $\boldsymbol{\xi} \in V(f^{\mathbf{A}}) \iff \mathbf{A}\boldsymbol{\xi} \in V(f)$ by definition, we conclude by combining those results. \square

In [Algorithm 1](#) of `GENERICPOINTPERCONNECTEDCOMPONENT`, we use the system introduced in [Theorem 4.1.5](#) as the starting system to solve rather than the more intuitive system of [\[19, Theorem 2.2\]](#). This is because the partial derivatives of $f^{\mathbf{A}}$ are generically of maximal degree $d - 1$ regardless of the structure of f for any variable, whereas the partial derivatives of f could have much lower degree in some variables. This input system therefore preserves part of the structure associated to f , and comes at the cost of an extra condition [\(A2\)](#) that \mathbf{A} must satisfy. Combining all these results yields:

Lemma 4.1.6. *Suppose that f satisfies [\(S\)](#), that \mathbf{A} satisfies [\(A1\)](#) and [\(A2\)](#), and that $\boldsymbol{\sigma}$ satisfies [\(B\)](#). Let us fix $1 \leq k \leq n$ and $r \in \mathbb{N} \setminus \{0\}$, and let Γ be a straight-line program evaluating f . If the call to `ZeroDimParam` is successful after r attempts, then `CRITICALPOINTS` applied to Γ , \mathbf{A} , k , $\boldsymbol{\sigma}$ and r correctly returns a zero-dimensional rational parametrisation of the set $\mathbf{A}(\text{crit}(\pi_k, V^{\mathbf{A}}) \cap \pi_{k-1}^{-1}(\sigma_1, \dots, \sigma_{k-1}))$.*

Proof. As f satisfies [\(S\)](#), \mathbf{A} satisfies [\(A1\)](#) and $\boldsymbol{\sigma}$ satisfies [\(B\)](#), we can apply [\[19, Theorem 2.2\]](#) to obtain that the set of solutions to the system

$$x_1 - \sigma_1 = \dots = x_{k-1} - \sigma_{k-1} = f^{\mathbf{A}} = \frac{\partial f^{\mathbf{A}}}{\partial x_{k+1}} = \dots = \frac{\partial f^{\mathbf{A}}}{\partial x_n} = 0$$

is finite and does not contain singular points. By [\[56, Lemma A.2\]](#), this solution set is precisely the set $\text{crit}(\pi_k, V^{\mathbf{A}}) \cap \pi_{k-1}^{-1}(\sigma_1, \dots, \sigma_{k-1})$. Since \mathbf{A} is invertible (as it satisfies [\(A1\)](#)) and satisfies [\(A2\)](#), we can apply [Theorem 4.1.5](#), to deduce that the set of solutions to the system

$$\begin{aligned} \left(\sum_{i=1}^n b_{1,i,0} x_i - \sigma_1, \dots, \sum_{i=1}^n b_{k-1,i,0} x_i - \sigma_{k-1}, f, \frac{\partial f}{\partial x_{k+1}} + \sum_{i=1}^k \left(\sum_{j=1}^{n-k} a_{i,(j+k)} b_{j,1,k} \right) \frac{\partial f}{\partial x_i}, \right. \\ \left. \dots, \frac{\partial f}{\partial x_n} + \sum_{i=1}^k \left(\sum_{j=1}^{n-k} a_{i,(j+k)} b_{j,n-k,k} \right) \frac{\partial f}{\partial x_i} \right) \end{aligned}$$

is the set $\mathbf{A}(\text{crit}(\pi_k, V^{\mathbf{A}}) \cap \pi_{k-1}^{-1}(\sigma_1, \dots, \sigma_{k-1}))$. In particular, it must also be finite and not contain singular points.

Since \mathbf{A} satisfies **(A2)**, we can apply **Polar** on **Subroutine 1**, which, given Γ , \mathbf{A} , k and σ , computes a straight-line program Λ that evaluates this system. As it has finitely many solutions, we can apply **ZeroDimParam** on **Subroutine 1** to Λ and r , which, under assumption of success, computes a zero-dimensional rational parametrisation \mathcal{Q} of the regular solutions to this system. Since \mathbf{A} satisfies **(A1)** and σ satisfies **(B)**, by [19, Theorem 2.2], its solutions are all regular, and \mathcal{Q} is therefore a parametrisation of the entire set $\mathbf{A}(\text{crit}(\pi_k, V^{\mathbf{A}}) \cap \pi_{k-1}^{-1}(\sigma_1, \dots, \sigma_{k-1}))$. Since the output of **CRITICALPOINTS** is precisely \mathcal{Q} (see **Subroutine 1**), we conclude that **CRITICALPOINTS** correctly returns a zero-dimensional rational parametrisation of the set $\mathbf{A}(\text{crit}(\pi_k, V^{\mathbf{A}}) \cap \pi_{k-1}^{-1}(\sigma_1, \dots, \sigma_{k-1}))$ when all the aforementioned assumptions are made, as required. \square

4.2 Correctness of **ISOLATION**

Recall that $V = V(f) \subset \mathbb{C}^n$ and that $S := \{\mathbf{x} \in \mathbb{R}^n \mid f(\mathbf{x}) \neq 0\}$. In this subsection, we show that, if f satisfies **(S)**, if the zero-dimensional rational parametrisation \mathcal{Q} parametrises a non-empty finite set Q , and if \mathbf{a} denotes a direction vector not tangent to $V \cap \mathbb{R}^n$ at any point of Q , then **ISOLATION** correctly outputs two zero-dimensional rational parametrisations, which encode points in each connected component of S whose boundary contains a point of Q . We begin by a small lemma concerning the behaviour of the boundary of S .

Lemma 4.2.1. *Suppose that f satisfies **(S)**. Then, any point in the boundary of S lies in the boundary of at most two distinct connected components of S .*

Proof. Let $\xi \in \overline{S} \setminus S$. Since f satisfies **(S)**, we can apply **Theorem 4.1.2** to conclude that there exists a neighbourhood \mathcal{N}^+ of ξ such that $\mathcal{N}^+ \cap S^+$ is connected. Therefore, there can only be at most one connected component of S such that ξ lies in its boundary and f is positive on it. By applying the same argument to $-f$, we also deduce the existence of a neighbourhood \mathcal{N}^- of ξ such that $\mathcal{N}^- \cap S^-$ is connected, and hence there can only be at most one connected component of S such that ξ lies in its boundary and f is negative on it. Because f is non-zero and has a fixed sign on any connected component of S , we conclude that there can only be at most two distinct connected components of S having ξ in their boundary. \square

Let us now introduce some notation. Suppose that we have $\xi \in V \cap \mathbb{R}^n$ and $\mathbf{a} \in \mathbb{Q}^n$ non-zero. We define, for a new variable u , $\ell_{\xi, \mathbf{a}} := \xi + \mathbf{a}u \in \mathbb{R}[u]^n$. This vector of univariate polynomials defines, as we let u vary in \mathbb{R} , the set of points lying on the affine line passing through ξ in the direction of \mathbf{a} , as illustrated below.

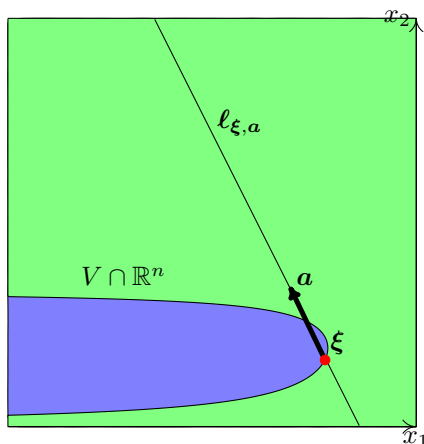


Figure 4: Illustration of $\ell_{\xi,a}$ on a two-dimensional example.

Let us now define the univariate polynomial $f_{\xi,a}(u) := f(\xi + au) \in \mathbb{R}[u]$. By construction, the roots of $f_{\xi,a}$ are the relative distances between ξ and the intersection points of $V \cap \mathbb{R}^n$ and $\ell_{\xi,a}$. In particular, 0 is a root of $f_{\xi,a}$, corresponding to ξ itself. Let us denote by $\{\zeta_1, \dots, \zeta_r\}$ the non-zero roots of $f_{\xi,a}$, when any exists. Finally, if $\xi \in V$, let us define $T_\xi V$ as the tangent hyperplane to V at ξ , a well-defined object when f satisfies **(S)**. We use the notation $\langle \mathbf{a}, \mathbf{b} \rangle$ to indicate the usual dot product of two vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$.

Lemma 4.2.2. *Suppose that f satisfies **(S)**. Let C be a connected component of S . Suppose that $\xi \in \overline{C} \setminus C$, and that $\ell_{\xi,a}$ does not lie in $T_\xi V$. Let us define $\delta := \min\{1, \min_{1 \leq i \leq r} |\zeta_i|\}$. Then, for any $0 < z < \delta$, either $\xi + z\mathbf{a}$ or $\xi - z\mathbf{a}$ lies in C .*

Proof. Let C and ξ be as in the statement. Since we suppose that $\ell_{\xi,a} \not\subset T_\xi V$, we have $\langle \text{Jac}_\xi(f), \mathbf{a} \rangle \neq 0$. Suppose without loss of generality that $\langle \text{Jac}_\xi(f), \mathbf{a} \rangle > 0$ (otherwise, consider $f_{\xi,-a}$). By Taylor expansion of $f_{\xi,a}$ at 0, we have

$$f_{\xi,a}(u) = f_{\xi,a}(0) + u f'_{\xi,a}(0) + o(u^2) = 0 + u \langle \text{Jac}_\xi(f), \mathbf{a} \rangle + o(u^2) = u \langle \text{Jac}_\xi(f), \mathbf{a} \rangle + o(u^2).$$

In particular, there exists a sufficiently small positive $\delta \in \mathbb{R}$ such that, for any $u \in]-\delta, \delta[$, $f_{\xi,a}(u)$ and u have the same sign, since $\langle \text{Jac}_\xi(f), \mathbf{a} \rangle$ is a positive constant.

Therefore, if we suppose that we have computed such a $\delta \in \mathbb{R}$, then, for any $0 < z < \delta$, $f_{\xi,a}(z) > 0$ and $f_{\xi,a}(-z) < 0$, and in particular $\xi + z\mathbf{a}$ and $\xi - z\mathbf{a}$ lie in different connected components of S that contain ξ in their boundary. Now, because f satisfies **(S)**, we can apply [Theorem 4.2.1](#) to conclude that there are at most two distinct connected components of S which have ξ in their boundary. Since there are exactly two such components, and since C is one of them by definition, we conclude that either $\xi + z\mathbf{a}$ or $\xi - z\mathbf{a}$ lies in C .

It therefore remains to show that the real number $m := \min\{1, \min_{1 \leq i \leq r} |\zeta_i|\}$ as defined in the statement is a suitable choice for δ . By definition, $m > 0$, and 0 is the only root of $f_{\xi, \mathbf{a}}$ in the interval $] - m, m[$. In particular, by continuity, $f_{\xi, \mathbf{a}}$ has constant sign on $] - m, 0[$ and on $]0, m[$, which means that m is a suitable choice for δ , as required. \square

Lemma 4.2.3. *Suppose that f satisfies **(S)**. Suppose that $\mathbb{Q} = (w, \mathbf{v}, \nu)$ is a zero-dimensional rational parametrisation of a non-empty subset $Q \subset V$. Let $\mathbf{a} \in \mathbb{Q}^n$ such that, for each $\xi \in Q$, $\ell_{\xi, \mathbf{a}}$ is not tangent to V at ξ . Let Γ be a straight-line program evaluating f . Then **ISOLATION** applied to Γ , \mathbf{a} and \mathbb{Q} correctly returns two zero-dimensional rational parametrisations \mathcal{P}^- and \mathcal{P}^+ of non-empty sets P^- and P^+ , such that $P^- \cup P^+$ intersects every connected component of S whose boundary contains a point of Q , and such that f does not vanish at any point of $P^- \cup P^+$.*

Proof. Let us denote by d the degree of f . Let $t_1 < \dots < t_r$ be the real roots of $w(t)$. By definition of a rational parametrisation, the set Q is equal to $\left\{ \left(\frac{1}{w'(t_i)} \mathbf{v}(t_i) \right)_{1 \leq i \leq r} \right\}$. Let us define $\xi_i := \frac{1}{w'(t_i)} \mathbf{v}(t_i)$. Following our previous notation, the set

$$\{\xi_i + u\mathbf{a}\}_{1 \leq i \leq r} \subset (\mathbb{Q}[u])^n$$

is therefore the set of parametrisations of the lines $\ell_{\xi_i, \mathbf{a}}$, and the set of intersection points of these lines with $V \cap \mathbb{R}^n$ is given by the real roots of the polynomials $f_{\xi_i, \mathbf{a}}(u) \in \mathbb{R}[u]$ for $1 \leq i \leq r$. Let $\zeta_{i,1}, \dots, \zeta_{i,s_i}$ be the non-zero real roots of $f_{\xi_i, \mathbf{a}}$, and define $\delta_i := \min\{1, \min_{1 \leq j \leq s_i} |\zeta_{i,j}|\}$. Let us define

$$f_{\mathbb{Q}, \mathbf{a}}(t, u) := f \left(\frac{1}{w'(t)} \mathbf{v}(t) + u\mathbf{a} \right) \in \mathbb{Q}(t)[u]$$

so that, for all $1 \leq i \leq r$, we have $f_{\mathbb{Q}, \mathbf{a}}(t_i, u) = f_{\xi_i, \mathbf{a}}(u)$. By definition of a rational parametrisation, $w(t)$ is squarefree, and hence $w'(t_i) \neq 0$ for all i . In particular, the polynomial $(w'(t))^d f_{\mathbb{Q}, \mathbf{a}} \in \mathbb{Q}[t, u]$ has, when t is instantiated to t_i , the same roots as $f_{\mathbb{Q}, \mathbf{a}}(t_i, u)$. We work with $(w'(t))^d f_{\mathbb{Q}, \mathbf{a}}$ in **ISOLATION** instead of $f_{\mathbb{Q}, \mathbf{a}}$, as it is a polynomial.

Because $w \in \mathbb{Q}[t]$ and $(w'(t))^d f_{\mathbb{Q}, \mathbf{a}}(t, u) \in \mathbb{Q}[t, u]$, and because $(w'(t_i))^d f_{\mathbb{Q}, \mathbf{a}}(t_i, u)$ is squarefree for all i by construction (otherwise, either f would have a square factor, impossible by **(S)**, or w' , and hence w , would have one, impossible by the definition of a zero-dimensional rational parametrisation), we can apply **ExtensionIsolation** on **Subroutine 2**, which, given these two polynomials, returns a tuple \mathcal{F} of r lists of isolation intervals for each root of $f_{\mathbb{Q}, \mathbf{a}}(t_i, u) = f_{\xi_i, \mathbf{a}}(u)$, for $1 \leq i \leq r$. Let E be the set of endpoints of all intervals present in \mathcal{F} , and, as done on **Subroutine 2**, define $\lambda := (1/2) \min\{1, \min\{|e| : e \in E \wedge e \neq 0\}\}$. By construction, for any $1 \leq i \leq r$, we have $0 < \lambda < \delta_i$. Note that λ does not depend on i .

Let C be a connected component of S whose boundary contains an element of Q , say ξ_i for some $1 \leq i \leq r$. Because f satisfies **(S)** and $\ell_{\xi_i, \mathbf{a}}$ is not tangent to V at ξ_i by

assumption, we can apply [Theorem 4.2.2](#) to deduce that either $\xi_i + \lambda \mathbf{a}$ or $\xi_i - \lambda \mathbf{a}$ lies in C . Therefore, if we define the sets

$$P^- := \{\xi_i - \lambda \mathbf{a}\}_{1 \leq i \leq r} \quad \text{and} \quad P^+ := \{\xi_i + \lambda \mathbf{a}\}_{1 \leq i \leq r},$$

then $P^- \cup P^+$ is non-empty, contains at least one point of every connected component of S whose boundary intersects Q , and does not contain any root of f by definition of λ . Moreover, the parametrisations \mathcal{P}^- and \mathcal{P}^+ computed on [Subroutine 2](#) respectively encode P^- and P^+ . Since the output of [ISOLATION](#) on [Subroutine 2](#) is precisely $\mathcal{P}^-, \mathcal{P}^+$, we conclude that, provided Q is not empty, [ISOLATION](#) correctly returns two zero-dimensional rational parametrisations of non-empty sets P^- and P^+ , such that their union intersects every connected component of S whose boundary contains a point of Q , and such that f does not vanish at any point, as required. \square

4.3 Correctness of [APPROXIMATION](#)

In this subsection, we show that, given a polynomial f and a zero-dimensional rational parametrisation \mathbb{Q} of a set Q that does not contain any root of f , [APPROXIMATION](#) correctly outputs rational approximations \mathbf{r} of $\mathbf{q} \in Q$ that lie in the same connected component of S as \mathbf{q} .

We begin by introducing some notation. Given a polynomial $f \in \mathbb{R}[x_1, \dots, x_n]$, we define $\|f\|_\infty$ as the maximum of the absolute values of its coefficients, and, given $\mathbf{x} \in \mathbb{R}^n$, we define $\|\mathbf{x}\|_\infty := \max_{1 \leq i \leq n} |x_i|$. Given a point $\mathbf{x} \in \mathbb{R}^n$ and $r > 0$, we also define

$$\mathcal{B}(\mathbf{x}, r) := \{\mathbf{y} \in \mathbb{R}^n : \|\mathbf{x} - \mathbf{y}\|_\infty \leq r\},$$

that is, the n -dimensional box of side-length $2r$ centred at \mathbf{x} .

Lemma 4.3.1. *Let $f \in \mathbb{Q}[x_1, \dots, x_n]$ have positive degree and $Q \subset \mathbb{R}^n$ be non-empty and finite, and assume that f does not vanish at any point of Q . Let:*

- $d = \deg(f)$,
- $\mathbf{c} = \|f\|_\infty$,
- $\mathcal{M} \geq \max_{\mathbf{q} \in Q} \{\|\mathbf{q}\|_\infty\} + 1$,
- $0 < m \leq \min_{\mathbf{q} \in Q} |f(\mathbf{q})|$,
- and

$$d := \min \left\{ 1, m \left(\binom{n+d}{d} \mathcal{M}^d \mathbf{c} \right)^{-1} \right\}.$$

Then f does not vanish at any point of $\mathcal{B}(\mathbf{q}, d)$, for any $\mathbf{q} \in Q$.

Proof. First, we note that, for d, c, \mathcal{M} and m as in the statement, d is well-defined, since all aforementioned quantities are positive. Suppose that we have shown that, for any $\mathbf{q} \in Q$, we have

$$\max_{\mathbf{p} \in \mathfrak{B}(\mathbf{q}, d)} |f(\mathbf{p}) - f(\mathbf{q})| < m. \quad (1)$$

Then, by definition of m and since $f(\mathbf{q}) \neq 0$ by assumption, for any $\mathbf{p} \in \mathfrak{B}(\mathbf{q}, d)$, we have $0 \leq |f(\mathbf{q})| - m < |f(\mathbf{p})| < |f(\mathbf{q})| + m$, and in particular f does not vanish on $\mathfrak{B}(\mathbf{q}, d)$. It therefore suffices to prove (1).

We begin by showing (1) for $\mathbf{q} = \mathbf{0} := (0, \dots, 0)$, and then reduce the general case to this one. Let $f = \sum_{0 \leq |\alpha| \leq d} f_\alpha \mathbf{x}^\alpha$. In this case, we have

$$\begin{aligned} \max_{\mathbf{p} \in \mathfrak{B}(\mathbf{0}, d)} |f(\mathbf{p}) - f(\mathbf{0})| &\leq \max_{\mathbf{p} \in \mathfrak{B}(\mathbf{0}, d)} \sum_{1 \leq |\alpha| \leq d} |f_\alpha| |\mathbf{p}^\alpha| \\ &\leq \sum_{1 \leq |\alpha| \leq d} |f_\alpha| \max_{\mathbf{p} \in \mathfrak{B}(\mathbf{0}, d)} |\mathbf{p}^\alpha| = \sum_{1 \leq |\alpha| \leq d} |f_\alpha| d^{|\alpha|} = d^d \sum_{1 \leq |\alpha| \leq d} |f_\alpha| \\ &< d^d \binom{n+d}{d} \|f\|_\infty \end{aligned} \quad (2)$$

$$\leq d \binom{n+d}{d} c, \quad (3)$$

$$\leq m \mathcal{M}^{-d} \binom{n+d}{d}^{-1} \leq m \quad (4)$$

where (2) follows from the fact that f has $\left(\binom{n+d}{d} - 1\right)$ non-constant terms, (3) from the definition of c and the fact that $0 < d \leq 1$, and (4) by construction of d and the fact that $\mathcal{M} \geq 1$. In particular, we obtain

$$\max_{\mathbf{p} \in \mathfrak{B}(\mathbf{0}, d)} |f(\mathbf{p}) - f(\mathbf{0})| < m$$

as required. Suppose now that $\mathbf{q} \in Q$ is arbitrary. Then, since translation by \mathbf{q} preserves distances, we have

$$\max_{\mathbf{p} \in \mathfrak{B}(\mathbf{q}, d)} |f(\mathbf{p}) - f(\mathbf{q})| = \max_{\mathbf{p} \in \mathfrak{B}(\mathbf{0}, d)} |f(\mathbf{p} - \mathbf{q}) - f(\mathbf{0})| < d \binom{n+d}{d} \|f(\mathbf{x} - \mathbf{q})\|_\infty$$

by applying the previously obtained result for $\mathbf{0}$ up to (2) and the fact that $0 < d \leq 1$. Now, we also have

$$\|f(\mathbf{x} - \mathbf{q})\|_\infty \leq c \mathcal{M}^d \binom{n+d}{d},$$

since $\mathcal{M} > \|\mathbf{q}\|_\infty$ and f has at most $\binom{n+d}{d}$ terms. Therefore, combining this result with the definition of d yields

$$\max_{\mathbf{p} \in \mathfrak{B}(\mathbf{q}, d)} |f(\mathbf{p}) - f(\mathbf{q})| < d c \mathcal{M}^d \binom{n+d}{d}^2 = m$$

which is precisely (1) for an arbitrary $\mathbf{q} \in Q$, as required. \square

With [Theorem 4.3.1](#) at hand, the correctness of [APPROXIMATION](#) readily follows:

Lemma 4.3.2. *Suppose that Γ is a straight-line program evaluating $f \in \mathbb{Q}[x_1, \dots, x_n]$ of positive degree, and that $\mathbb{Q} = (w, \mathbf{v}, \nu)$ is a zero-dimensional rational parametrisation which parametrises a non-empty set $Q = \{\mathbf{q}_1, \dots, \mathbf{q}_s\} \subset \mathbb{R}^n$. For any $1 \leq i \leq s$, suppose that $f(\mathbf{q}_i) \neq 0$, and let C_i be the connected component of S in which \mathbf{q}_i lies. Then [APPROXIMATION](#) applied to Γ and \mathbb{Q} correctly returns a finite set of rational approximations $\{\mathbf{r}_1, \dots, \mathbf{r}_s\} \subset \mathbb{Q}^n$ of $\{\mathbf{q}_1, \dots, \mathbf{q}_s\}$, such that $\mathbf{r}_i \in C_i$ for all $1 \leq i \leq s$.*

Proof. Let $d = \deg(f)$ and $c = \|f\|_\infty$, as computed in [Subroutine 3](#), which are both positive as f has positive degree. The number m computed in [Subroutine 3](#) satisfies by construction $0 < m \leq \min_{1 \leq i \leq s} |f(\mathbf{q}_i)|$, and \mathcal{M} computed in [Subroutine 3](#) satisfies $\mathcal{M} \geq \max_{1 \leq i \leq s} \{\|\mathbf{q}_i\|_\infty\} + 1 > 0$. Since f has positive degree, since Q is finite and non-empty, and since f does not vanish on any point of Q , we can apply [Theorem 4.3.1](#) to deduce f does not vanish at any point of $\mathcal{B}(\mathbf{q}_i, d)$ for any $\mathbf{q}_i \in Q$, where d is the quantity computed in [Subroutine 3](#).

In particular, if C_i is the connected component of S in which \mathbf{q}_i lies, any rational approximation \mathbf{r}_i of \mathbf{q}_i that lies in $\mathcal{B}(\mathbf{q}_i, d)$ will also lie in C_i , since the boundary of C_i lies in $V \cap \mathbb{R}^n$, which does not intersect $\mathcal{B}(\mathbf{q}_i, d)$. It therefore suffices to compute a rational approximation \mathbf{r}_i of \mathbf{q}_i that lies in $\mathcal{B}(\mathbf{q}_i, d)$ for all $\mathbf{q}_i \in Q$. By definition, to have $\mathbf{r}_i \in \mathcal{B}(\mathbf{q}_i, d)$, it must satisfy $\|\mathbf{r}_i - \mathbf{q}_i\|_\infty \leq d$, and hence have all its coordinates accurate to precision at least $2^{\log(d)}$.

The set \mathcal{A} computed in [Subroutine 3](#) is precisely a set of rational approximations of points of Q accurate to precision $2^{\log(d)}$, and therefore contains rational approximations $\{\mathbf{r}_1, \dots, \mathbf{r}_s\}$ of $\{\mathbf{q}_1, \dots, \mathbf{q}_s\}$ such that $\mathbf{r}_i \in C_i$. Since the output of [APPROXIMATION](#) is precisely \mathcal{A} on [Subroutine 3](#), we conclude that [APPROXIMATION](#) correctly returns a set $\{\mathbf{r}_1, \dots, \mathbf{r}_s\} \subset \mathbb{Q}^n$ such that $\mathbf{r}_i \in C_i$, as required. \square

4.4 Correctness of [GENERICPOINTPERCONNECTEDCOMPONENT](#)

We now prove the correctness of [GENERICPOINTPERCONNECTEDCOMPONENT](#), using the results obtained in the previous subsections.

Theorem 4.4.1. *Let $0 < \epsilon < 1$. Suppose that f satisfies assumption [\(S\)](#), that the matrix \mathbf{A} chosen in [Algorithm 1](#) satisfies [\(A1\)](#) and [\(A2\)](#), that the point $\boldsymbol{\sigma}$ chosen in [Algorithm 1](#) satisfies [\(B\)](#), and that all calls to [ZeroDimParam](#) are successful. Let Γ be a straight-line program evaluating f . Then [GENERICPOINTPERCONNECTEDCOMPONENT](#) applied to Γ and ϵ correctly returns a finite subset of \mathbb{Q}^n which contains at least one point per connected component of S .*

Proof. If S is empty, there is nothing to prove, so let us suppose that $S \neq \emptyset$. If $V \cap \mathbb{R}^n = \emptyset$, then S contains a single connected component, equal to \mathbb{R}^n . Since [GENERICPOINTPERCONNECTEDCOMPONENT](#) always returns at least one point by construction, this case is covered, and we can hence assume without loss of generality that $V \cap \mathbb{R}^n \neq \emptyset$. Moreover,

if f is constant, S is also the whole space (since we assume it is not empty), and we hence also assume without loss of generality that f has positive degree.

As S and $V \cap \mathbb{R}^n$ are not empty, let C be a connected component of S and B be a connected component of its boundary, and denote by $C^{\mathbf{A}}$ the corresponding connected component of $S^{\mathbf{A}}$, and by $B^{\mathbf{A}}$ the corresponding connected component of its boundary. Let us assume for now that $\pi_1(C^{\mathbf{A}}) \neq \mathbb{R}$, which implies $\pi_1(B^{\mathbf{A}}) \neq \mathbb{R}$. We now show that, in this case, the first iteration of the `for` loop, for $k = 1$, computes a finite set $\mathcal{A}_1 \subset \mathbb{Q}^n$ that contains at least one point of C .

Because, by assumption, f satisfies **(S)**, \mathbf{A} satisfies **(A1)** and **(A2)**, σ satisfies **(B)**, and because we assume that all calls to `ZeroDimParam` are successful, we can apply [Theorem 4.1.6](#) to conclude that `CRITICALPOINTS` applied to $\Gamma, \mathbf{A}, 1, \sigma, \lceil \log(3n\epsilon^{-1}) \rceil$, as in [Algorithm 1](#), correctly returns a zero-dimensional rational parametrisation $\mathbb{Q}_1 = (w_1, \mathbf{v}_1, \nu_1)$ of the finite set $\mathbf{A}(\text{crit}(\pi_1, V^{\mathbf{A}}))$. As f satisfies **(S)**, we can apply [Theorem 4.1.3](#) to deduce that B is a connected component of $V \cap \mathbb{R}^n$. Because f satisfies **(S)** and \mathbf{A} satisfies **(A1)**, we can apply [Theorem 4.1.4](#) to deduce that $\pi_1(B^{\mathbf{A}})$ is closed for the Euclidean topology. Since $\pi_1(B^{\mathbf{A}}) \neq \mathbb{R}$, we therefore deduce that $B^{\mathbf{A}} \cap \text{crit}(\pi_1, V^{\mathbf{A}}) \neq \emptyset$. In particular, $\text{crit}(\pi_1, V^{\mathbf{A}})$ itself is non-empty, and hence $w_1 \neq 1$.

We are thus in the situation of [Algorithm 1](#) of `GENERICPOINTPERCONNECTEDCOMPONENT`. As it is non-empty, let $\xi \in B^{\mathbf{A}} \cap \text{crit}(\pi_1, V^{\mathbf{A}})$, so that $\mathbf{A}\xi$ is one of the points parametrised by \mathbb{Q}_1 . Because ξ is a critical point of π_1 restricted to $V^{\mathbf{A}}$, the line $\ell_{\xi, \mathbf{e}_1}^{\mathbf{A}}$, defined by its parametrisation $\xi + \mathbf{e}_1 u$ for $\mathbf{e}_1 = (1, 0, \dots, 0)^{\mathbf{T}}$ and a new variable u , is normal to $V^{\mathbf{A}}$ at ξ . Therefore, by applying the change of variables \mathbf{A} and defining $\mathbf{a}_1 := (a_{1,1}, \dots, a_{n,1})^{\mathbf{T}}$, we deduce that the line ℓ_{ξ, \mathbf{a}_1} , defined by its parametrisation $\mathbf{A}(\xi + \mathbf{e}_1 u) = \mathbf{A}\xi + \mathbf{a}_1 u$, is not in the tangent hyperplane $\mathbf{T}_{\xi} V$.

Therefore, \mathbb{Q}_1 is a zero-dimensional rational parametrisation of the non-empty set $\mathbf{A}(\text{crit}(\pi_1, V^{\mathbf{A}}))$, on which f does not vanish, and the line ℓ_{ξ, \mathbf{a}_1} is not tangent to V at ξ . Because f satisfies **(S)** and \mathbf{a}_1 does not depend on the choice of ξ , we can apply [Theorem 4.2.3](#) to conclude that `ISOLATION` applied to Γ, \mathbf{a}_1 and \mathbb{Q}_1 as in [Algorithm 1](#) correctly returns two zero-dimensional rational parametrisations \mathcal{P}_1^- and \mathcal{P}_1^+ of non-empty sets P_1^- and P_1^+ , whose union intersects every connected component having a point of $\mathbf{A}(\text{crit}(\pi_1, V^{\mathbf{A}}))$ in its boundary. In particular, as $\xi \in B^{\mathbf{A}} \cap \text{crit}(\pi_1, V^{\mathbf{A}})$, we have $\mathbf{A}\xi \in B \cap \mathbf{A} \text{crit}(\pi_1, V^{\mathbf{A}})$, and hence $P_1^- \cup P_1^+$ contains at least one point of C . Moreover, [Theorem 4.2.3](#) also implies that f does not vanish at any point of $P^- \cup P^+$.

Since neither P_1^- and P_1^+ are empty, and f is non-zero at every point of $P_1^- \cup P_1^+$, we can apply [Theorem 4.3.2](#) to conclude that `APPROXIMATION` applied to Γ and \mathcal{P}^- and \mathcal{P}^+ , as in [Algorithm 1](#), correctly returns sets \mathcal{A}_1^- and \mathcal{A}_1^+ of rational approximations of points of P_1^- and P_1^+ that lie in the same connected components of S as their exact counterparts. In particular, since $P_1^- \cup P_1^+$ contains a point of C , so does $\mathcal{A}_1 := \mathcal{A}_1^- \cup \mathcal{A}_1^+$. Therefore, the set \mathcal{A}_1 computed on [Algorithm 1](#) of `GENERICPOINTPERCONNECTEDCOMPONENT` is a finite subset of \mathbb{Q}^n that contains at least one point of C , as required.

Let us now suppose that $\pi_1(C^{\mathbf{A}}) = \mathbb{R}$. In this case, $C^{\mathbf{A}}$ has non-empty intersection with every fibre of π_1 , and, in particular, it therefore suffices to compute a point of $C^{\mathbf{A}} \cap \pi_1^{-1}(\sigma_1)$ to obtain a point of $C^{\mathbf{A}}$, and hence of C . If we suppose now that $\pi_2(C^{\mathbf{A}} \cap \pi_1^{-1}(\sigma_1)) \neq \mathbb{R}$, then this implies $\pi_2(B^{\mathbf{A}} \cap \pi_1^{-1}(\sigma_1)) \neq \mathbb{R}$. Because f satisfies **(S)**, \mathbf{A} satisfies **(A1)** and **(A2)**, σ satisfies **(B)**, and because we assume that all calls to `ZeroDimParam` are successful, we can apply the same argument as before to deduce that the `for` loop in the $k = 2$ case computes a finite set $\mathcal{A}_2 \subset \mathbb{Q}^n$ that contains at least one point of C . Otherwise, we have $\pi_2(C^{\mathbf{A}} \cap \pi_1^{-1}(\sigma_1)) = \mathbb{R}$, and it hence suffices to consider $C^{\mathbf{A}} \cap \pi_2^{-1}(\sigma_1, \sigma_2)$. We repeat this procedure, incrementing k each time.

By the end of the n -th and last iteration of the `for` loop, we have therefore obtained finite sets $\mathcal{A}_1, \dots, \mathcal{A}_n \subset \mathbb{R}^n$ such that, for any connected component C of S , either $\bigcup_{i=1}^n \mathcal{A}_i$ contains a point of C , or C satisfies $\pi_n(C^{\mathbf{A}} \cap \pi_{n-1}^{-1}(\sigma)) = \mathbb{R}$. In the latter case, this implies in particular that $C^{\mathbf{A}} \cap \pi_n^{-1}(\sigma_1, \dots, \sigma_{n-1}, 0) = C^{\mathbf{A}} \cap (\sigma_1, \dots, \sigma_{n-1}, 0)$ is non-empty, and therefore that $(\sigma_1, \dots, \sigma_{n-1}, 0) \in C^{\mathbf{A}}$ and hence that $\mathbf{A}(\sigma_1, \dots, \sigma_{n-1}, 0)^{\mathbf{T}} \in C$. We therefore conclude that, for any connected component C of S , the finite set $\bigcup_{i=1}^n \mathcal{A}_i \cup \{\mathbf{A}(\sigma_1, \dots, \sigma_{n-1}, 0)^{\mathbf{T}}\} \subset \mathbb{Q}^n$ contains at least one point of C . As this set is precisely the output of `GENERICPOINTPERCONNECTEDCOMPONENT` on [Algorithm 1](#), we conclude that `GENERICPOINTPERCONNECTEDCOMPONENT` correctly returns a finite subset of \mathbb{Q}^n that contains at least one point per connected component of S , as required. \square

5 Probability of success

In this section, we compute the probability of success of `GENERICPOINTPERCONNECTEDCOMPONENT` for a given polynomial f that satisfies **(S)** and a given $0 < \epsilon < 1$.

Our probabilistic analysis follows from ideas and results by [\[18, 19, 40\]](#), where we aim to find the degree of a polynomial describing a hypersurface which contains the closed Zariski sets we wish to avoid. Once this degree obtained, we can compute the probability of a randomly chosen point to not cancel this polynomial, and hence of not being in the aforementioned closed Zariski sets, using the so-called *DeMillo-Lipton-Schwartz-Zippel Lemma* [\[58\]](#), which we recall below for the reader's convenience.

DeMillo-Lipton-Schwartz-Zippel Lemma [\[58\]](#): *Let R be an integral domain and $p \in R[x_1, \dots, x_n]$ be a non-zero polynomial. Let $T \subset R$ be finite such that $|T| > \deg(p)$, and let (x_1, \dots, x_n) be a tuple of values selected uniformly at random from T . Then the probability that $p(x_1, \dots, x_n) = 0$ is bounded above by $\frac{\deg(p)}{|T|}$.*

By [Theorem 4.4.1](#), as the given f is assumed to satisfy **(S)**, `GENERICPOINTPERCONNECTEDCOMPONENT` succeeds when we have correctly chosen $\mathbf{A} \in \mathbb{Q}^{n \times n}$ satisfying **(A1)** and **(A2)**, $\sigma \in \mathbb{Q}^{n-1}$ satisfying **(B)**, and when all calls to `ZeroDimParam` are successful. The probabilities of correctly satisfying **(A1)** and **(B)** were already studied in [\[19\]](#), and we recall its key results:

[\[19, Theorem 2.1\]](#): *There exists a non-zero polynomial $h \in \mathbb{C}[\mathfrak{A}]$ of degree at most*

$5n^3(2d)^{2n}$ such that, if $h(\mathbf{A}) \neq 0$, then \mathbf{A} satisfies **(A1)**.

[19, Theorem 2.2]: There exists a non-zero polynomial $g \in \mathbb{C}[\mathfrak{S}]$ of degree at most nd^{2n} such that, if $g(\boldsymbol{\sigma}) \neq 0$, then $\boldsymbol{\sigma}$ satisfies **(B)**.

Combining these results with the DeMillo-Lipton-Schwartz-Zippel Lemma yields the following propositions. Note that all these proofs follow a similar structure to [19, Section 5].

Proposition 5.1. *Suppose that $0 < \epsilon < 1$ and that the $n \times n$ matrix \mathbf{A} is randomly drawn with coefficients within the set $X := \left\{1, \dots, \left\lceil 3\epsilon^{-1} \left(5n^3(2d)^{2n} + \frac{n^2-n}{2}\right) \right\rceil\right\}$. Then, the probability that \mathbf{A} satisfies **(A1)** and **(A2)** is at least $1 - \epsilon/3$.*

Proof. Let ϵ and X be as in the statement. For \mathbf{A} to satisfy **(A2)**, the determinant of each bottom right $(n-k) \times (n-k)$ submatrix of \mathbf{A} must be non-zero, for $1 \leq k \leq n$. As each such determinant is a non-zero polynomial of degree at most $n-k$ in the variables \mathfrak{A} , we conclude, by taking the product of all these polynomials, that there exists a non-zero polynomial $p \in \mathbb{C}[\mathfrak{A}]$ of degree at most $\sum_{k=1}^n (n-k) = \frac{n^2-n}{2}$ such that \mathbf{A} satisfies **(A2)** whenever $p(\mathbf{A}) \neq 0$.

On the other hand, by [19, Theorem 2.1], there exists a non-zero $h \in \mathbb{C}[\mathfrak{A}]$ of degree at most $5n^3(2d)^{2n}$ such that, whenever $h(\mathbf{A}) \neq 0$, \mathbf{A} satisfies **(A1)**. Therefore, there exists a non-zero polynomial $q := ph \in \mathbb{C}[\mathfrak{A}]$ of degree at most $5n^3(2d)^{2n} + \frac{n^2-n}{2}$ such that, whenever $q(\mathbf{A}) \neq 0$, \mathbf{A} satisfies **(A1)** and **(A2)**.

Thus, if we pick \mathbf{A} with arbitrary coefficients in X , by the DeMillo-Lipton-Schwartz-Zippel Lemma, since $0 < \epsilon < 1$, the probability that a randomly chosen \mathbf{A} cancels q is

$$\mathbb{P}[q(\mathbf{A}) = 0] \leq \frac{\deg q}{|X|} = \frac{5n^3(2d)^{2n} + \frac{n^2-n}{2}}{\left\lceil 3\epsilon^{-1} \left(5n^3(2d)^{2n} + \frac{n^2-n}{2}\right) \right\rceil} \leq \frac{\epsilon}{3}.$$

Hence, the probability that a randomly drawn $\mathbf{A} \in X^{n \times n}$ satisfies **(A1)** and **(A2)** is at least $1 - \epsilon/3$, as required. \square

Proposition 5.2. *Suppose that $0 < \epsilon < 1$ and that the point $\boldsymbol{\sigma}$ is randomly drawn with coefficients within the set $Y := \{1, \dots, \lceil 3\epsilon^{-1}nd^{2n} \rceil\}$. Then the probability that $\boldsymbol{\sigma}$ satisfies **(B)** is at least $1 - \epsilon/3$.*

Proof. Let ϵ and Y be as in the statement. By [19, Theorem 2.2], there exists non-zero polynomial $g \in \mathbb{C}[\mathfrak{S}]$ of degree at most nd^{2n} such that, if $g(\boldsymbol{\sigma}) \neq 0$, then $\boldsymbol{\sigma}$ satisfies **(B)**. By the DeMillo-Lipton-Schwartz-Zippel Lemma, since $0 < \epsilon < 1$, the probability that a randomly chosen $\boldsymbol{\sigma}$ cancels g is

$$\mathbb{P}[g(\boldsymbol{\sigma}) = 0] \leq \frac{\deg g}{|Y|} = \frac{nd^{2n}}{\lceil 3\epsilon^{-1}nd^{2n} \rceil} \leq \frac{\epsilon}{3}.$$

Therefore, the probability that a randomly drawn $\sigma \in Y^{n-1}$ satisfies **(B)** is at least $1 - \epsilon/3$. \square

On the other hand, the probability of success of a single call to `ZeroDimParam` was studied in [54, Theorem 1], and is at least $21/32$. Combining all these results yields:

Theorem 5.3. *Suppose that $0 < \epsilon < 1$, that $f \in \mathbb{Q}[x_1, \dots, x_n]$ satisfies **(S)**, and that Γ is a straight-line program evaluating f . Then, the probability of success of `GENERICPOINTPERCONNECTEDCOMPONENT` applied to Γ and ϵ is at least $1 - \epsilon$.*

Proof. Because f satisfies **(S)**, by Theorem 4.4.1, `GENERICPOINTPERCONNECTEDCOMPONENT` is correct when the matrix \mathbf{A} chosen in Algorithm 1 satisfies **(A1)** and **(A2)**, the point σ chosen in Algorithm 1 satisfies **(B)**, and when all calls to `ZeroDimParam` are correct.

As, in Algorithm 1, the matrix \mathbf{A} is picked with randomly drawn coefficients in the set $X := \left\{1, \dots, \left\lceil 3\epsilon^{-1} \left(5n^3(2d)^{2n} + \frac{n^2-n}{2}\right) \right\rceil\right\}$, we can apply Theorem 5.1 to deduce that it has a probability of at least $1 - \epsilon/3$ of satisfying **(A1)** and **(A2)**. Similarly, in Algorithm 1, the point σ is picked with randomly draw coefficients in the set $Y := \left\{1, \dots, \left\lceil 3\epsilon^{-1}nd^{2n} \right\rceil\right\}$, we can apply Theorem 5.2 to deduce that it has a probability of at least $1 - \epsilon/3$ of satisfying **(B)**. Finally, the probability of success of a single call to `ZeroDimParam` is at least $\frac{21}{32} > \frac{1}{2}$, by [54, Theorem 1], and we need to call it on n distinct systems, once per iteration of the `for` loop. Hence, if we call `ZeroDimParam` $\lceil \log(3n\epsilon^{-1}) \rceil$ times per iteration, the probability that it returns a correct result in every iteration is therefore at least

$$\left(1 - 2^{-\lceil \log(3n\epsilon^{-1}) \rceil}\right)^n \geq \left(1 - \frac{\epsilon}{3n}\right)^n \geq 1 - \frac{\epsilon}{3}$$

by Bernoulli's inequality, since $0 < \epsilon < 1$.

Therefore, the probability that \mathbf{A} satisfies **(A1)** and **(A2)**, σ satisfies **(B)**, and that all calls to `ZeroDimParam` are correct, since $0 < \epsilon < 1$, is at least

$$\left(1 - \frac{\epsilon}{3}\right) \left(1 - \frac{\epsilon}{3}\right) \left(1 - \frac{\epsilon}{3}\right) \geq 1 - \epsilon.$$

Because f satisfies **(S)**, by applying Theorem 4.4.1, we therefore conclude that `GENERICPOINTPERCONNECTEDCOMPONENT` correctly returns a finite subset of \mathbb{Q}^n containing at least one point per connected component of S with probability at least $1 - \epsilon$, as required. \square

6 Bit complexity: proofs of Theorems 1 to 3

We now analyse the bit complexity of `GENERICPOINTPERCONNECTEDCOMPONENT`. We use O and \tilde{O} to indicate the bit cost of an operation with and without logarithmic factors respectively. For the remainder of this section, we suppose, to simplify the statements of the results, that the degree of the input polynomial is at least 2, since computing points per connected components in the linear case reduces to elementary computations.

6.1 Bit Complexity of **CRITICALPOINTS**

We begin with the bit complexity of **CRITICALPOINTS**. As its core is the subroutine **ZeroDimParam**, we recall for convenience the main result of [54] regarding its bit complexity:

[54, Theorem 1]: *Let (f_1, \dots, f_n) be a tuple of polynomials in $\mathbb{Q}[\mathbf{x}_1, \dots, \mathbf{x}_m]$, where $\mathbf{x}_i = (x_{i,1}, \dots, x_{i,n_i})$, and $n = n_1 + \dots + n_m$. Suppose that $V(f_1, \dots, f_n)$ is zero-dimensional. Suppose that $\deg_{\mathbf{x}_j}(f_i) \leq \tilde{d}_{i,j}$, that $\text{ht}(f_i) \leq s_i$, and that there exists a straight-line program Γ of length L evaluating (f_1, \dots, f_n) , using constants of heights at most b . Then there exists an algorithm **ZeroDimParam** which produces either:*

- a zero-dimensional rational parametrisation of the non-singular complex solutions to $f_1 = \dots = f_n = 0$,
- or a zero-dimensional rational parametrisation of a subset of the solutions,
- or fail.

The first outcome occurs with probability at least $21/32$, and in any case, the algorithm performs

$$\tilde{O}(Lb + \mathcal{C}_n(\mathbf{d})\mathcal{H}_n(\boldsymbol{\eta}, \mathbf{d})(L + n\mathfrak{d} + n^2)n(\log(s) + n))$$

bit operations, where

- $\mathfrak{d} := \max_{1 \leq i \leq n} (\tilde{d}_{i,1} + \dots + \tilde{d}_{i,m})$,
- $s := \max_{1 \leq i \leq n} (s_i)$,
- $\boldsymbol{\eta} := \left(s_i + \sum_{j=1}^m \log(n_j + 1) \tilde{d}_{i,j} \right)_{1 \leq i \leq n}$,
- $\mathcal{C}_n(\mathbf{d})$ is the sum of the coefficients of the polynomial

$$\prod_{i=1}^n (\tilde{d}_{i,1}\theta_1 + \dots + \tilde{d}_{i,m}\theta_m) \pmod{\langle \theta_1^{n_1+1}, \dots, \theta_m^{n_m+1} \rangle},$$

- and $\mathcal{H}_n(\boldsymbol{\eta}, \mathbf{d})$ is the sum of the coefficients of the polynomial

$$\prod_{i=1}^n (\eta_i \zeta + \tilde{d}_{i,1}\theta_1 + \dots + \tilde{d}_{i,m}\theta_m) \pmod{\langle \zeta^2, \theta_1^{n_1+1}, \dots, \theta_m^{n_m+1} \rangle}.$$

The algorithm calls an oracle \mathcal{O} , taking as input a positive integer B and returning a prime number in $\{B + 1, \dots, 2B\}$, with the input $s\mathfrak{d}^{O(n)}$. Moreover, upon success, any polynomial in the output has degree at most $\mathcal{C}_n(\mathbf{d})$ and height in $\tilde{O}(\mathcal{H}_n(\boldsymbol{\eta}, \mathbf{d}) + n\mathcal{C}_n(\mathbf{d}))$.

This result gives us the bit complexity **ZeroDimParam** subroutine, provided that we know the length of the input straight-line program. This is given to us by the following result:

Baur-Strassen Theorem [6, Theorem 1]: Let $f \in \mathbb{Q}[x_1, \dots, x_n]$, and suppose that there exists a straight-line program Γ of length L evaluating f . Then, there exists a straight-line program Λ of length at most $3L$ evaluating the polynomials $\left(f, \frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n}\right)$.

We now have all the tools to express the bit complexity of **CRITICALPOINTS**. To apply [54, Theorem 1] in its full generality, we suppose in this section that the input polynomial f has a multi-homogeneous structure, that is, $f \in \mathbb{Q}[\mathbf{x}_1, \dots, \mathbf{x}_m]$, where $\mathbf{x}_i = \left(x_{1+\sum_{j=1}^{i-1} n_j}, \dots, x_{n_i+\sum_{j=1}^{i-1} n_j}\right)$, $n = n_1 + \dots + n_m$, and $\deg_{\mathbf{x}_i}(f) \leq d_i$. We moreover assume that $d_1 \leq \dots \leq d_m$, since this can always be ensured by simply re-labelling the groups of variables. To simplify notation, for given $\mathbf{A} \in \mathbb{Q}^{n \times n}$, $\boldsymbol{\sigma} \in \mathbb{Q}^{n-1}$, and f as above, we also define:

- $d := \deg(f)$,
- $\delta_{c,\ell,k} := \deg_{\mathbf{x}_c} \left(\frac{\partial f}{\partial x_\ell} + \sum_{i=1}^k \left(\sum_{j=1}^{n-k} a_{i,(j+k)} b_{j,\ell-k,k} \right) \frac{\partial f}{\partial x_i} \right)$, for $1 \leq c \leq m$, $1 \leq k \leq n$, and $\ell > k$,
- $\tau := \text{ht}(f)$,
- $\alpha := \max_{1 \leq i,j \leq n} \text{ht}(a_{i,j})$,
- $\sigma := \max_{1 \leq i \leq n-1} \text{ht}(\sigma_i)$,
- $\gamma := n\alpha + \tau + d \log(n)$,
- Δ as the sum of the coefficients of the polynomial

$$(d_1\theta_1 + \dots + d_m\theta_m) \prod_{i=2}^n (\delta_{1,i,1}\theta_1 + \dots + \delta_{m,i,1}\theta_m) \pmod{\langle \theta_1^{n_1+1}, \dots, \theta_m^{n_m+1} \rangle},$$

- \mathbf{H} as the sum of the coefficients of the polynomial

$$(\gamma\zeta + d_1\theta_1 + \dots + d_m\theta_m) \prod_{i=2}^n (\gamma\zeta + \delta_{1,i,1}\theta_1 + \dots + \delta_{m,i,1}\theta_m) \pmod{\langle \zeta^2, \theta_1^{n_1+1}, \dots, \theta_m^{n_m+1} \rangle}.$$

Lemma 6.1.1. Suppose that $f \in \mathbb{Q}[\mathbf{x}_1, \dots, \mathbf{x}_m]$ satisfies **(S)**, $\mathbf{A} \in \mathbb{Q}^{n \times n}$ satisfies **(A1)** and **(A2)**, and that $\boldsymbol{\sigma} \in \mathbb{Q}^{n-1}$ satisfies **(B)**. Let $1 \leq k \leq n$ and $r \in \mathbb{N} \setminus \{0\}$, and suppose that $\sigma < \alpha$. Let Γ be a straight-line program of length L evaluating f . Then **CRITICALPOINTS** applied to Γ , \mathbf{A} , k , $\boldsymbol{\sigma}$ and r performs at most

$$\tilde{O}(rn^2 \Delta \mathbf{H}(L + nd + n^3))$$

bit operations. Moreover, upon success, any polynomial in the output has degree at most Δ and height in $\tilde{O}(\mathbf{H} + n\Delta)$.

Proof. Since \mathbf{A} satisfies **(A2)**, each $b_{i,j,k}$ is well-defined. Moreover, by Cramer's rule and Hadamard's inequality, we have

$$\text{ht}(b_{i,j,k}) \leq \log \left(n^{n/2} (2^\alpha)^n \right) \in \tilde{O}(n \log(n) + n\alpha) = \tilde{O}(n\alpha).$$

We begin by computing the length of the straight-line program Λ computed in **Subroutine 1**. Recall that it evaluates the polynomials \mathcal{P}_k , which consist of:

- $k - 1$ polynomials, $\sum_{i=1}^n b_{1,i,0}x_i - \sigma_1, \dots, \sum_{i=1}^n b_{k-1,i,0}x_i - \sigma_{k-1}$,
- 1 polynomial, f ,
- and $n - k$ polynomials,

$$\frac{\partial f}{\partial x_{k+1}} + \sum_{i=1}^k \left(\sum_{j=1}^{n-k} a_{i,(j+k)} b_{j,1,k} \right) \frac{\partial f}{\partial x_i}, \dots, \frac{\partial f}{\partial x_n} + \sum_{i=1}^k \left(\sum_{j=1}^{n-k} a_{i,(j+k)} b_{j,n-k,k} \right) \frac{\partial f}{\partial x_i}.$$

By Baur-Strassen [6, Theorem 1], there exists a straight-line program of length at most $3L$ evaluating f and its partial derivatives. We include in this program the n^2 entries of \mathbf{A} , the n^2 entries of \mathbf{A}^{-1} , the $(n-k)^2$ entries of B_k^{-1} , and the $n-1$ entries of $\boldsymbol{\sigma}$. Finally, to obtain the full system of **Polar**, we perform $2n(k-1)$ operations with those entries to obtain the first $k-1$ polynomials, leave f unchanged, and perform $(n-k)(k+2k(n-k))$ operations to obtain the remaining $n-k$ polynomials. Therefore, the length of Λ is bounded by

$$3L + n^2 + n^2 + (n-k)^2 + n-1 + 2n(k-1) + k(n-k)(1+2(n-k)) \in O(L + n^3).$$

We now compute the quantities appearing in the statement of [54, Theorem 1] for this system:

- $\text{len}(\Lambda) \in \tilde{O}(L + n^3)$,
- $\tilde{d} = (\tilde{d}_1, \dots, \tilde{d}_n)$, where:
 - $\tilde{d}_1 = \dots = \tilde{d}_{k-1} = (1, \dots, 1)$,
 - $\tilde{d}_k = (d_1, \dots, d_m)$,
 - $\tilde{d}_{k+1} = (\delta_{1,k+1,k}, \dots, \delta_{m,k+1,k}), \dots, \tilde{d}_n = (\delta_{1,n,k}, \dots, \delta_{m,n,k})$,
- $\mathbf{s} \in \tilde{O}((n\alpha, \dots, n\alpha, \tau, n\alpha + \tau + \log(d), \dots, n\alpha + \tau + \log(d)))$, since $\sigma < \alpha$,
- $\mathfrak{d} = d$,
- $b \in \tilde{O}(n\alpha + \tau + \log(d)) \in \tilde{O}(\gamma)$,
- $\boldsymbol{\eta} \in \tilde{O}((n\alpha, \dots, n\alpha, \tau + d \log(n), \gamma, \dots, \gamma))$.

By [54, Theorem 1], the complexity of `ZeroDimParam` is increasing in the entries of $\tilde{\mathbf{d}}$, \mathbf{s} and $\boldsymbol{\eta}$, and in this particular case, all of their entries are the largest when $k = 1$, since we assume that $d_1 \leq \dots \leq d_m$. It therefore suffices to compute the bit complexity of `CRITICALPOINTS` when $k = 1$ to obtain a bound on its complexity for all k . If $k = 1$, the above quantities yield

- $\mathcal{C}_n(\mathbf{d}) = \Delta$,
- $\mathcal{H}_n(\boldsymbol{\eta}, \mathbf{d}) \in \tilde{O}(\mathbf{H})$.

Since f satisfies **(S)**, \mathbf{A} satisfies **(A1)** and **(A2)**, and $\boldsymbol{\sigma}$ satisfies **(B)**, we can apply [19, Theorem 2.2] to conclude that the set of complex solutions to the system parametrised by Λ is zero-dimensional and regular. Therefore, we can apply [54, Theorem 1] to Λ to conclude that `ZeroDimParam` applied to Λ , \mathbf{A} , 1, $\boldsymbol{\sigma}$ and r , as done in `Subroutine 1`, performs at most

$$\begin{aligned} & \tilde{O}\left(r\left((L+n^3)\gamma + \Delta\mathbf{H}(L+n^3+nd+n^2)n(\log(\gamma)+n)\right)\right) \\ & \subseteq \tilde{O}\left(rn^2\Delta\mathbf{H}(L+nd+n^3)\right) \end{aligned}$$

bit operations, since we repeat the computation r times and since $\mathbf{H} \geq \gamma$ by construction. The statement on the degree and height of the output polynomials follows immediately from the last part of [54, Theorem 1]. \square

6.2 Bit Complexity of `ISOLATION`

We now analyse the bit complexity of `ISOLATION`. The main tool we use for this analysis is the following result by [61]:

[61, Theorem 16]: *Suppose that $p \in \mathbb{Q}[t]$ and $q \in \mathbb{Q}[t, u]$. Let $a = \deg(p)$, $b = \deg_u(q)$, $\tau = \text{ht}(p)$ and $\sigma = \text{ht}(q)$. Then, there exists an algorithm `ExtensionIsolation` which takes p and q as inputs, and returns isolation intervals for all real roots of $q(t_i, u)$, if squarefree, for all real roots t_i of p , which performs at most $\tilde{O}(ab^3 + a^2b^2(\tau + \sigma))$ bit operations. Moreover, the height of any computed endpoint is in $\tilde{O}(ab^2(\tau + \sigma))$.*

With this theorem at hand, the complexity of `ISOLATION` readily follows.

Lemma 6.2.1. *Suppose that $f \in \mathbb{Q}[x_1, \dots, x_n]$ satisfies **(S)**, has degree d and height τ , and is evaluated by a straight-line program Γ . Suppose that $\mathbb{Q} = (w, \mathbf{v}, \nu)$ is a zero-dimensional rational parametrisation with polynomials of degree δ and height η , such that $w \neq 1$, and f vanishes on every point parametrised by k . Let $\mathbf{a} \in \mathbb{Q}^n$ satisfy the requirements of `ISOLATION`, and suppose that $\text{ht}(\mathbf{a}) \leq \eta$. Then `ISOLATION` applied to Γ , \mathbf{a} and \mathbb{Q} performs at most*

$$\tilde{O}(n\delta^2d^2\tau + n\delta^2d^3\eta + n\delta^3d^3)$$

bit operations. Moreover, the polynomials in the output parametrisations \mathcal{P}^- and \mathcal{P}^+ have degree at most δ and height in $\tilde{O}(\delta d^2\tau + \delta d^3\eta + \delta^2d^3)$.

Proof. The polynomial $w(t)$ has by assumption degree δ and height η . By Mignotte's bound [25, Corollary 6.33], the polynomial $(w'(t))^d$ has height

$$d(\log(\delta - 1) + \eta) + \log \left(\left(\sqrt{d\delta + 1} \right) 2^{d(\delta-1)} \binom{d}{\lfloor d/2 \rfloor} \right) \in \tilde{O}(d(\eta + \delta)).$$

Therefore, the polynomial $(w'(t))^d f \left(\frac{1}{w'(t)} \mathbf{v}(t) + u\mathbf{a} \right)$ has degree d in u , and has height in

$$\tilde{O}(\log(d(\delta - 1) + 1) + d(\eta + \delta) + (\tau + d(\eta + \delta))) = \tilde{O}(\tau + d(\eta + \delta)).$$

Therefore, by [61, Theorem 16], `ExtensionIsolation` applied to the polynomials $w(t) \in \mathbb{Q}[t]$ and $(w'(t))^d f \left(\frac{1}{w'(t)} \mathbf{v}(t) + u\mathbf{a} \right) \in \mathbb{Q}[t, u]$, as done in `Subroutine 2`, performs at most

$$\tilde{O}(\delta d^3 + \delta^2 d^2(\eta + \tau + d(\eta + \delta))) = \tilde{O}(\delta^2 d^2 \tau + \delta^2 d^3 \eta + \delta^3 d^3)$$

bit operations. It returns at most $2d\delta$ endpoints, of height

$$\tilde{O}(\delta d^2(\eta + \tau + d(\eta + \delta))) = \tilde{O}(\delta d^2 \tau + \delta d^3 \eta + \delta^2 d^3).$$

Therefore, the rational λ of `Subroutine 2` is computed in at most $\tilde{O}(d\delta)$ bit operations, and also has height in $\tilde{O}(\delta d^2 \tau + \delta d^3 \eta + \delta^2 d^3)$. Hence, the parametrisations \mathcal{P}^- and \mathcal{P}^+ computed in `Subroutine 2` have polynomials of degree at most δ and height in

$$\tilde{O}(\delta d^2 \tau + \delta d^3 \eta + \delta^2 d^3 + \eta + \log(\delta - 1) + \eta) = \tilde{O}(\delta d^2 \tau + \delta d^3 \eta + \delta^2 d^3).$$

Computing them requires $\tilde{O}(n\delta)$ operations between rationals having at most this height, and thus costs

$$\tilde{O}(n\delta^2 d^2 \tau + n\delta^2 d^3 \eta + n\delta^3 d^3)$$

bit operations.

Therefore, the most expensive steps of `ISOLATION` are `Subroutine 2`, and hence `ISOLATION` performs at most $\tilde{O}(n\delta^2 d^2 \tau + n\delta^2 d^3 \eta + n\delta^3 d^3)$ bit operations, and outputs two rational parametrisations with polynomials of degree δ and height $\tilde{O}(\delta d^2 \tau + \delta d^3 \eta + \delta^2 d^3)$, as required. \square

6.3 Bit Complexity of `APPROXIMATION`

We now analyse the bit complexity of `APPROXIMATION`. The main result used for this section is the following:

[46, Theorem 47]: *Let \mathbb{Q} be a zero-dimensional rational parametrisation of a set Q with polynomials of degree at most δ and height η , and let κ be a positive integer. Then, there exists an algorithm `ParamApproximation` which, given \mathbb{Q} and κ , computes approximations of the real points of Q accurate to precision $2^{-\kappa}$ in $\tilde{O}(\delta^3 + n\delta^2 \eta + n\delta \kappa)$ bit operations.*

It therefore remains to determine a suitable value of κ to perform the approximation, which is the purpose of the following results:

Lemma 6.3.1. *Let $W \subset \mathbb{R}^n$ be a zero-dimensional variety defined over \mathbb{Q} , of degree δ and height η . Then any non-zero coordinate b of any point of W satisfies*

$$\frac{1}{n^{(n/2+1)}\delta^n (1 + 2^{2\eta+10\log(n+3)\delta})} \leq |b| \leq n^{(n/2+1)}\delta^n (1 + 2^{2\eta+10\log(n+3)\delta}).$$

Proof. As W contains at most δ points by definition, there exists an invertible change of variables matrix $\mathbf{M} \in \mathbb{Q}^{n \times n}$ with entries of height at most $\log(\delta)$ such that each point of $W^{\mathbf{M}^{-1}} := \{\mathbf{M}w : w \in W\}$ has distinct i -th coordinates, for any $1 \leq i \leq n$. When this is the case, the ideal $\mathbf{I}(W^{\mathbf{M}^{-1}})$ satisfies [16, Assumption 1], and we can hence apply [16, Theorem 1] to conclude that, for any $1 \leq i \leq n$, there exists a polynomial $N_i \in \mathbb{Q}[x_i]$ such that

$$\begin{aligned} \deg(N_i) &\leq \delta, & \text{ht}(N_i) &\leq \eta + 5\log(n+3)\delta, \\ N_i(c) = 0 &\iff \exists \xi \in W^{\mathbf{M}^{-1}} : \xi_i = c. \end{aligned}$$

Moreover, by the Cauchy bounds [5, Lemmata 10.2-10.3], if $N = \sum_{j=0}^{\delta} n_j x_i^j$ with $n_\delta \neq 0$, any non-zero root c of N_i satisfies

$$\frac{1}{1 + \max_{0 \leq j < \delta} \left| \frac{n_j}{n_\delta} \right|} \leq |c| \leq 1 + \max_{0 \leq j < \delta} \left| \frac{n_j}{n_\delta} \right|.$$

Combining these results for all $1 \leq i \leq n$ implies that any non-zero coordinate c of any point in $W^{\mathbf{M}^{-1}}$ satisfies

$$\frac{1}{1 + 2^{2\eta+10\log(n+3)\delta}} \leq |c| \leq 1 + 2^{2\eta+10\log(n+3)\delta}.$$

Now, by Cramer's rule and Hadamard's inequality, since the entries of \mathbf{M} have height at most $\log(\delta)$, the entries of \mathbf{M}^{-1} have height at most $\frac{1}{2}n \log(n) + n \log(\delta)$. We hence conclude that any non-zero coordinate b of any point in W satisfies

$$\frac{1}{n^{(n/2+1)}\delta^n (1 + 2^{2\eta+10\log(n+3)\delta})} \leq |b| \leq n^{(n/2+1)}\delta^n (1 + 2^{2\eta+10\log(n+3)\delta})$$

as required. □

Corollary 6.3.2. *Let $W \subset \mathbb{R}^n$ be a zero-dimensional variety defined over \mathbb{Q} , of degree δ and height η . Let $f \in \mathbb{Q}[x_1, \dots, x_n]$ be a polynomial of degree d and height τ . Then any non-zero real value b that f takes on W satisfies*

$$\frac{1}{n^{(n+3)/2}\delta^{n+1} (1 + 4^{d\eta+\tau\delta+6\log(n+4)\delta})} \leq |b| \leq (n+1)^{(n+3)/2}\delta^{n+1} (1 + 4^{d\eta+\tau\delta+6\log(n+4)\delta}).$$

In particular, if \mathbb{Q} is a zero-dimensional rational parametrisation of W , `LowerBoundValue` and `UpperBoundValue`, when applied to \mathbb{Q} and f , both return rational numbers having height in $\tilde{O}(n + d\eta + \tau\delta)$.

Proof. Let t be a new variable. To obtain bounds on the non-zero real values that f takes on W , it suffices to bound the non-zero real values of the variety $W \cap V(f - t)$. Since each point of W corresponds to a single value of t by construction, this variety is also zero-dimensional, and has the same degree δ as W . Moreover, by [40, Corollary 2.10], we have

$$\text{ht}(W \cap V(f - t)) \leq d\eta + (\tau + \log(n + 1))\delta.$$

Therefore, by applying Theorem 6.3.1 to $W \cap V(f - t)$, we conclude that any non-zero real value b that f takes on W satisfies

$$\frac{1}{n^{(n+3)/2}\delta^{n+1} (1 + 4^{d\eta + \tau\delta + 6\log(n+4)\delta})} \leq |b| \leq (n + 1)^{(n+3)/2}\delta^{n+1} (1 + 4^{d\eta + \tau\delta + 6\log(n+4)\delta}).$$

By taking logarithms, we conclude that these bounds both have heights in

$$\begin{aligned} & \tilde{O}((n + 3)\log(n + 1) + (n + 1)\log(\delta) + d\eta + \tau\delta + \log(n + 4)\delta) \\ &= \tilde{O}(n + d\eta + \tau\delta) \end{aligned}$$

as required. \square

From this corollary, a bound on the bit cost of APPROXIMATION readily follows:

Lemma 6.3.3. *Let $f \in \mathbb{Q}[x_1, \dots, x_n]$ be of degree d and height τ , and let Γ be a straight-line program evaluating f . Let $\mathbb{Q} = (w, \mathbf{v}, \nu)$ be a zero-dimensional rational parametrisation of a set Q , with polynomials of degree at most δ and height η . Suppose that f does not vanish on any point parametrised by \mathbb{Q} , and let $\gamma := \text{ht}(Q)$. Then, APPROXIMATION applied to Γ and \mathbb{Q} performs at most*

$$\tilde{O}(\delta^3 + n\delta^2\eta + n^2d\delta + nd^2\delta\gamma + \tau nd\delta^2)$$

bit operations, and returns a finite set of δ rationals having height in $\tilde{O}(nd + d^2\gamma + \tau d\delta)$.

Proof. By definition, the rational ϵ of Subroutine 3 has height τ . By Theorem 6.3.2, the rationals m and \mathcal{M} of Subroutine 3 have heights in $\tilde{O}(n\log(\delta) + d\gamma + \tau\delta + \delta\log(n))$. Therefore, d as computed in Subroutine 3 has height in

$$\begin{aligned} & \tilde{O}(\tau + (d + 1)(n + d\gamma + \tau\delta) + 2(n + d)\log(n + d)) \\ &= \tilde{O}(nd + d^2\gamma + \tau d\delta) \end{aligned}$$

and can be computed in quasi-linear time with respect to its height. Therefore, by [46, Theorem 47], the set \mathcal{A} of Subroutine 3 is computed in at most

$$\begin{aligned} & \tilde{O}(\delta^3 + n\delta^2\eta + n\delta(nd + d^2\gamma + \tau d\delta)) \\ &= \tilde{O}(\delta^3 + n\delta^2\eta + n^2d\delta + nd^2\delta\gamma + \tau nd\delta^2) \end{aligned}$$

bit operations, and contains rationals having height in $\tilde{O}(nd + d^2\gamma + \tau d\delta)$, as required. \square

6.4 Bit Complexity of **GENERICPOINTPERCONNECTEDCOMPONENT**

We now finally analyse the bit complexity of **GENERICPOINTPERCONNECTEDCOMPONENT**. Recall that we assume that $d > 1$.

Theorem 1. *Suppose that $f \in \mathbb{Q}[x_1, \dots, x_n]$ satisfies **(S)**, is of degree d and height τ , and that Γ is a straight-line program of length L evaluating f . Let $0 < \epsilon < 1$. Then **GENERICPOINTPERCONNECTEDCOMPONENT** performs at most*

$$\tilde{O}(\log(1/\epsilon)n^3\Delta\mathbf{H}(L + nd + n^3) + nd^5\Delta^3(\mathbf{H} + n\Delta))$$

bit operations, with probability of success at least $1 - \epsilon$. Upon success, it returns a finite subset of \mathbb{Q}^n containing at most $2n\Delta + 1$ elements, each of height in

$$\tilde{O}(d^5\Delta^2(\mathbf{H} + n\Delta)).$$

Proof. As the complexity of **CRITICALPOINTS** computed in **Theorem 6.1.1**, as well as the degree and height of polynomials in the output, are independent of k , it suffices to compute the bit complexity of a single iteration of the **for** loop, and multiply the result by n to obtain the complexity of **GENERICPOINTPERCONNECTEDCOMPONENT**.

By construction, each entry of the matrix \mathbf{A} chosen in **Algorithm 1** has height

$$\alpha := \log\left(3\epsilon^{-1}\left(5n^3(2d)^{2n} + \frac{n^2 - n}{2}\right)\right) \in \tilde{O}(\log(1/\epsilon) + n\log(d))$$

and each entry of the vector $\boldsymbol{\sigma}$ chosen in **Algorithm 1** has height no higher than α . We can therefore apply **Theorem 6.1.1** to conclude that the bit cost of **Algorithm 1** is at most

$$\tilde{O}(\log(1/\epsilon)n^2\Delta\mathbf{H}(L + nd + n^3)).$$

Upon success, it returns a zero-dimensional rational parametrisation $\mathbb{Q}_k = (w_k, \mathbf{v}_k, \nu_k)$ of degree at most Δ and height in $\tilde{O}(\mathbf{H} + n\Delta)$. If $w_k \equiv 1$, the loop terminates here, so suppose that $w_k \not\equiv 1$. As $\alpha \leq \gamma \leq \mathbf{H} \leq \mathbf{H} + n\Delta$, we can apply **Theorem 6.2.1** to conclude that the bit cost of **Algorithm 1** is at most

$$\begin{aligned} & \tilde{O}(nd\Delta^2d^2\tau + n\Delta^2d^3(\mathbf{H} + n\Delta) + n\Delta^3d^3) \\ &= \tilde{O}(nd^3\Delta^2(\mathbf{H} + n\Delta)), \end{aligned}$$

since $\tau \leq \gamma \leq \mathbf{H}$, and that the output parametrisations each have degree bounded by Δ and height in $\tilde{O}(d^3\Delta(\mathbf{H} + n\Delta))$. By using the arithmetic Bézout theorem [40], along with [40, Lemma 2.6], we know that the heights of the algebraic sets P^- and P^+ parametrised by \mathcal{P}^- and \mathcal{P}^+ satisfy

$$\begin{aligned} \text{ht}(P^\pm) &\leq n\gamma\Delta + (2n + 3)\Delta\log(n + 1) + (d^3\Delta(\mathbf{H} + n\Delta))\Delta + 5\Delta\log(n + 1) \\ &\in \tilde{O}(d^3\Delta^2(\mathbf{H} + n\Delta)). \end{aligned}$$

Therefore, by applying [Theorem 6.3.3](#), we conclude that the bit cost of [Algorithm 1](#) is at most

$$\begin{aligned} & \tilde{O}\left(\Delta^3 + n\Delta^2(d^3\Delta(\mathbf{H} + n\Delta)) + n^2d\Delta + nd^2\Delta(d^3\Delta^2(\mathbf{H} + n\Delta)) + \tau nd\Delta^2\right) \\ &= \tilde{O}(nd^5\Delta^3(\mathbf{H} + n\Delta)), \end{aligned}$$

and that the output rationals each have height in

$$\begin{aligned} & \tilde{O}(nd + d^2(d^3\Delta^2(\mathbf{H} + n\Delta)) + \tau d\Delta) \\ &= \tilde{O}(d^5\Delta^2(\mathbf{H} + n\Delta)) \end{aligned}$$

as required. Therefore, the total bit complexity of [GENERICPOINTPERCONNECTEDCOMPONENT](#) is at most n times the total complexity of an iteration of the `for` loop, that is, at most

$$\begin{aligned} & \tilde{O}\left(n\left(\log(1/\epsilon)n^2\Delta\mathbf{H}(L + nd + n^3) \right. \right. \\ & \quad \left. \left. + nd^3\Delta^2(\mathbf{H} + n\Delta) \right. \right. \\ & \quad \left. \left. + nd^5\Delta^3(\mathbf{H} + n\Delta)\right)\right) \\ &= \tilde{O}\left(\log(1/\epsilon)n^3\Delta\mathbf{H}(L + nd + n^3) + nd^5\Delta^3(\mathbf{H} + n\Delta)\right) \end{aligned}$$

as required. □

Using [Theorem 1](#), we obtain the following result, concerning the case where f does not have any particular multi-homogeneous structure, that is, $f \in \mathbb{Q}[\mathbf{x}_1]$. In this case, to simplify notation, we define

- $\delta_i := \deg\left(\frac{\partial f}{\partial x_i}\right)$ for $1 \leq i \leq n$,
- $\mathbf{D} := d\delta_2 \cdots \delta_n$,

and further assume that $\delta_1 \leq \cdots \leq \delta_n$, since this can always be ensured by re-labelling the variables.

Theorem 2. *In the setting of [Theorem 1](#), when $f \in \mathbb{Q}[\mathbf{x}_1]$, [GENERICPOINTPERCONNECTEDCOMPONENT](#) performs at most*

$$\tilde{O}(n^2\mathbf{D}^2(n\log(1/\epsilon) + n^2 + \tau + d)(\log(1/\epsilon)n^2(L + nd + n^3) + d^5\mathbf{D}^2))$$

bit operations, with probability of success at least $1 - \epsilon$. Upon success, it returns a finite subset of \mathbb{Q}^n containing at most $2n\mathbf{D} + 1$ elements, each of height in

$$\tilde{O}(nd^5\mathbf{D}^3(n\log(1/\epsilon) + n^2 + \tau + d)).$$

Proof. In the particular case where \mathbf{x}_1 is the only group of variables, we have:

- $\alpha \in \tilde{O}(\log(1/\epsilon) + n\log(d))$,

- $\gamma \in \tilde{O}(n \log(1/\epsilon) + n^2 + \tau + d)$,
- $\deg\left(\frac{\partial f}{\partial x_\ell} + \sum_{i=1}^k \left(\sum_{j=1}^{n-k} a_{i,(j+k)} b_{j,\ell-k,k}\right) \frac{\partial f}{\partial x_i}\right) \leq \delta_\ell$ for $\ell > k$, since we assume that $\delta_1 \leq \dots \leq \delta_\ell$.

Using these quantities in the definitions of Δ and H yield

- $\Delta = D$,
- $H \in \tilde{O}(n\gamma D)$.

Hence, substituting these values in the bit complexity result of [Theorem 1](#) gives

$$\begin{aligned} & \tilde{O}(\log(1/\epsilon)n^4\gamma D^2(L + nd + n^3) + nd^5 D^3(n\gamma D + nD)) \\ &= \tilde{O}(n^2\gamma D^2(\log(1/\epsilon)n^2(L + nd + n^3) + d^5 D^2)) \\ &= \tilde{O}(n^2 D^2(n \log(1/\epsilon) + n^2 + \tau + d)(\log(1/\epsilon)n^2(L + nd + n^3) + d^5 D^2)) \end{aligned}$$

as a bound for the total number of bit operations of [GENERICPOINTPERCONNECTEDCOMPONENT](#). The statement about the height of the output is obtained similarly. \square

Finally, if one only wishes to compute the zero-dimensional rational parametrisations of the points of S , without their rational approximation, we have the following result:

Theorem 3. *If one only wishes to obtain the zero-dimensional rational parametrisations of the points of $S := \{\mathbf{x} \in \mathbb{R}^n \mid f(\mathbf{x}) \neq 0\}$ instead of their rational approximation, this can be done in at most*

$$\tilde{O}(\log(1/\epsilon)n^3 \Delta H(L + nd + n^3) + nd^3 \Delta^2(H + n\Delta))$$

bit operations in the setting of [Theorem 1](#), or

$$\tilde{O}(n^2 D^2(n \log(1/\epsilon) + n^2 + \tau + d)(\log(1/\epsilon)n^2(L + nd + n^3) + d^3 D))$$

bit operations in the setting of [Theorem 2](#). There are $2n$ such parametrisations computed, consisting of polynomials of degree at most Δ , respectively D , and height in $\tilde{O}(d^3 \Delta(H + n\Delta))$, respectively $\tilde{O}(nd^3 D^2(n \log(1/\epsilon) + n^2 + \tau + d))$.

Proof. It suffices to apply [GENERICPOINTPERCONNECTEDCOMPONENT](#) without computing the approximations, that is, returning the \mathcal{P}_k^\pm once they have been computed. The complexity of doing so is thus the same as the one of [Theorem 1](#), without the complexity of [Algorithm 1](#). By [Theorem 1](#), this therefore gives a bit complexity of

$$\begin{aligned} & \tilde{O}\left(n(\log(1/\epsilon)n^2 \Delta H(L + nd + n^3) \right. \\ & \quad \left. + nd^3 \Delta^2(H + n\Delta))\right) \\ &= \tilde{O}(\log(1/\epsilon)n^3 \Delta H(L + nd + n^3) + nd^3 \Delta^2(H + n\Delta)) \end{aligned}$$

in the multi-homogeneous setting. In the case where f has no multi-homogeneous structure, substituting $\Delta = \mathbf{D}$ and $\mathbf{H} \in \tilde{O}(n\gamma\mathbf{D})$, as in [Theorem 2](#), yields a bit complexity of

$$\tilde{O}(n^2\mathbf{D}^2(n \log(1/\epsilon) + n^2 + \tau + d)(\log(1/\epsilon)n^2(L + nd + n^3) + d^3\mathbf{D}))$$

as required. The statements about the heights of the polynomials in the output are obtained similarly. \square

These results are precisely the bit complexity statements appearing in [Theorems 2](#) and [3](#).

We remark that the complexities obtained in [Theorem 1](#) and [Theorem 3](#) are, in the worst case scenario, essentially quartic and cubic, respectively, in the Bézout bound $d(d-1)^{n-1}$, since this quantity is a bound for \mathbf{D} .

7 Experiments

All computations were performed on a single thread of an Intel(R) Xeon(R) Gold 6244 CPU @ 3.60GHz with 1500 GB of memory. We have implemented `GENERICPOINTPERCONNECTEDCOMPONENT` in SageMath [\[57\]](#), using the library `msolve` [\[8\]](#) for the computation of zero-dimensional rational parametrisations and real root isolation. The code has been submitted as a ticket to SageMath to be made publicly available within the software in a later version ¹.

We compare these computations with our implementation of the algorithm presented in [\[18, 19, 55\]](#), which computes points per connected components of real algebraic sets. When the input polynomial of `GENERICPOINTPERCONNECTEDCOMPONENT` is $f \in \mathbb{Q}[x_1, \dots, x_n]$, the algorithm of [\[55\]](#) applied to $x_{n+1}f - 1$ for a new variable x_{n+1} returns the same information about sample points of connected components of S . We also compare our results to the Cylindrical Algebraic Decomposition [\[13\]](#), more precisely to the dedicated function `SAMPLEPOINTS`, from the `RegularChains` package present in the computer algebra system `Maple` [\[45\]](#).

Finally, we also compare our results to the numerical solver `HYPERSURFACEREGIONS` [\[11\]](#), available as a package in the `Julia` language [\[9\]](#), and to the satisfiability modulo theory (SMT) solver implemented in the `SMTLIB` package of `Maple`. These two solvers are fundamentally different to the previous ones. The latter simply answers the question “Is $S^+ := \{\mathbf{x} \in \mathbb{R}^n \mid f(\mathbf{x}) > 0\}$ empty?”, and returns a sample point if it is not; it therefore computes less information than the other algorithms. The former computes sample points as our algorithm does, but also determines whether two sample points lie in the same connected component or not; it therefore computes more information than our algorithm. However, it is numerical, and hence does not guarantee that its solutions are accurate, as shown below:

¹A functional prototype of the code can be found at <https://gillot.perso.lip6.fr/smooth.sage>.

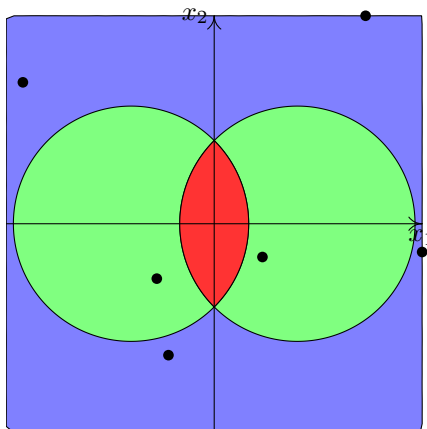


Figure 5: Illustration of the output of `HYPERSURFACEREGIONS` applied to $f = ((x_0 - 1)^2 + x_1^2 - 1 - 1/2^{16})((x_0 + 1)^2 + x_1^2 - 1 - 1/2^{16})$. Despite the success of the computation, the central component (in red) is missed (exaggerated size for visual clarity).

We apply these algorithms to three types of examples:

- generic dense polynomials of fixed degree 4 and height 6, with varying number n of variables,
- generic polynomials of fixed degree 12, with varying number n of variables, such that $n - k$ of their partial derivatives have degree 1, for some varying $0 \leq k \leq n$,
- and small deformations of singular examples arising from applications of other scientific fields, namely `Vor1` and `Vor2` from computational geometry problems [20], `P3PGen` from computer vision [21], `K1-K4` and `Kalto1-Kalto4` from real algebra theorem-proving [39], and `Sot1` and `Sot2` from problems appearing in enumerative geometry [60]. These are all polynomials with 4 to 8 variables and having degrees between 4 and 18, of varying density².

The ∞ symbol indicates that no result has been obtained after 25 days of computation, the `OOM` symbol indicates memory requirements exceeding 1500 GB, and the `Err` symbol indicates that the program returned an error.

Along with computation time, we also compare the total number of points obtained using the different algorithms. Note that this number may vary between runs for both the algorithm of [55] and `Algorithm 1`, as it depends on the randomly chosen change of variables matrix \mathbf{A} and specialisation point σ .

²The examples can be found at <https://gillot.perso.lip6.fr/Examples/list.html>.

n	REGULAR-CHAINS.MM		[55, Algorithm 1]		GENERICPOINT-PERCONNECTED-COMPONENT		HYPERSURFACE-REGIONS.JL		SMT-LIB.MM
	Time	Points	Time	Points	Time	Points	Time	Points	Time
4	11 000	20 607	14	23	3	13	84	19	1
5	∞	—	500	61	10	65	110	33	1
6	∞	—	37 000	97	120	73	230	53	40
7	∞	—	∞	—	8 800	177	1 100	117	1
8	∞	—	∞	—	1 800 000	183	7 400	113	1
9	∞	—	∞	—	∞	—	38 000	269	1
10	∞	—	∞	—	∞	—	170 000	349	430

Table 1: Comparisons (in seconds) on generic dense polynomials of fixed degree 4 and height 6 with varying number n of variables.

In Table 1, we observe that, on random examples, although `GENERICPOINTPERCONNECTEDCOMPONENT` outperforms other symbolic algorithms, it is less effective than `HYPERSURFACEREGIONS.JL` in dealing with examples of large dimensions. Of course, unlike other symbolic algorithms, `HYPERSURFACEREGIONS.JL` does not guarantee that all connected components have been sampled. Moreover, as expected, `SMTLIB.MM` easily terminates, since finding a point of S^+ (resp. S^-) is not expected to be a difficult problem when f is randomly generated.

Furthermore, we observe computationally that, in our implementation of `GENERICPOINTPERCONNECTEDCOMPONENT`, solving the polynomial system to compute the critical points (that is, Algorithm 1) is the most time-consuming step. This differs from our complexity analysis, where Algorithm 1, the solution approximation steps, were limiting. This can be explained by the fact that the height bounds on computed quantities, such as the quantity λ of `ISOLATION`, are worst-case bounds that are seldom reached in practice.

The last case that terminates for our algorithm, $n = 8$, requires to solve a polynomial system of degree 8748, and the case $n = 9$, which did not terminate, of degree 26244. This degree is known since `msolve` is based on multimodular techniques: in the allowed time frame, the system was solved modulo some prime numbers, but not enough to perform rational reconstruction, suggesting that speed-ups may be obtained through multi-threading (as these modular computations are independent).

n	REGULAR-CHAINS.MM		[55, Algorithm 1]		GENERICPOINT-PERCONNECTED-COMPONENT		HYPERSURFACE-REGIONS.JL		SMT-LIB.MM
	Time	Points	Time	Points	Time	Points	Time	Points	Time
8, 2	260	3 105	19 000	49	24	117	220	41	1
8, 3	64 000	21 343	∞	—	11 000	237	10 000	105	1
8, 4	∞	—	∞	—	∞	—	480 000	243	∞
12, 2	20 000	25 181	OOM	—	31	89	850	37	∞
12, 3	∞	—	OOM	—	20 000	241	19 000	95	∞
20, 2	∞	—	OOM	—	110	205	3 300	45	1
20, 3	∞	—	OOM	—	56 000	345	140 000	Err ⁴	1
50, 1	∞	—	OOM	—	210	261	5 900	9	1
50, 2	∞	—	OOM	—	960	325	200 000	43	2
100, 0	1	4	22 000	247	1 200	257	15 000	13	19
150, 0	Err ³	—	260 000	387	7 300	397	100 000	17	10

Table 2: Comparisons (in seconds) on generic polynomials of fixed degree 12 and height 6, with varying number n of variables, such that $n - k$ partial derivatives have degree 1. When $k = 0$, the polynomial becomes a generic dense polynomial of degree 2.

In Table 2, we have randomly generated polynomials such that most of their partial derivatives have much lower degree than expected, which is precisely the case where D is much lower than $d(d - 1)^{n-1}$, and hence the case where we expect GENERICPOINT-PERCONNECTEDCOMPONENT to perform well, according to our complexity analysis. Indeed, as can be observed in Table 2, GENERICPOINTPERCONNECTEDCOMPONENT is competitive with HYPERSURFACEREGIONS.JL in many such examples, and even outperforms SMTLIB.MM in some examples. REGULARCHAINS.MM remains however the most efficient algorithm in practice when $d = 2$.

The one example, (8, 4), that we did not manage to compute involved a polynomial system of degree 29282. The other examples had degree at most 2662.

³REGULARCHAINS returns an error due to a type list containing too many elements for Maple. We however still expect the CAD to outperform our algorithm in these degree 2 examples, both in computation time and number of sample points, were it not for this type error.

⁴This is an intentional error by HYPERSURFACEREGIONS.JL, occurring when it computes critical points that are numerically too close to being singular points.

n	REGULAR-CHAINS.MM		[55, Algorithm 1]		GENERICPOINT-PERCONNECTED-COMPONENT		HYPERSURFACE-REGIONS.JL		SMT-LIB.MM
	Time	Points	Time	Points	Time	Points	Time	Points	Time
Vor1	4	48	230	47	17	53	200	Err ⁴	1
Vor2	∞	—	830 000	87	17 000	149	220 000	23	OOM
P3PGen	∞	—	∞	—	80 000	709	8 100	234	1
K1	OOM	—	11 000	643	280	381	8 400	Err ⁴	1
K2	1	112	9 600	693	290	345	3 400	Err ⁴	1
K3	∞	—	560	355	56	201	1 600	Err ⁴	1
K4	2	224	11 000	717	340	373	6 700	Err ⁴	1
Kalto1	∞	—	∞	—	∞	—	∞	—	1 100 000
Kalto2	∞	—	∞	—	∞	—	∞	—	850 000
Kalto3	∞	—	∞	—	∞	—	∞	—	950 000
Kalto4	∞	—	∞	—	∞	—	∞	—	1 000 000
Sot1	∞	—	∞	—	200 000	125	2 000 000	47	1
Sot2	∞	—	∞	—	190 000	125	∞	—	OOM

Table 3: Comparisons (in seconds) on deformed ($1/2^{32}$ subtracted) singular examples arising from applications.

Finally, in Table 3, we can see that HYPERSURFACE-REGIONS.JL fails in most examples due to internal checks, as expected given that the input polynomials are very close to defining a singular hypersurface. Depending on the examples, however, either REGULAR-CHAINS.MM or GENERICPOINT-PERCONNECTED-COMPONENT outperform the other. We also do manage to outperform SMT-LIB.MM on some examples. The highest degree of a polynomial system appearing in an example which we managed to solve was 10296.

Acknowledgements. The authors are supported by ANR Project ANR-22-CE91-0007 “EAGLES” and the FA8655-25-1-7469 of the European Office of Aerospace Research and Development of the Air Force Office of Scientific Research (AFOSR).

References

- [1] M.-E. Alonso, E. Becker, M.-F. Roy, and T. Wörmann. “Zeros, multiplicities, and idempotents for zero-dimensional systems”. In: *Algorithms in Algebraic Geometry and Applications. Proceedings of MEGA’94 (Progress in Mathematics)*. Vol. 142. Birkhäuser Basel, 1996, pp. 1–15.
- [2] P. Aubry, F. Rouillier, and M. Safey El Din. “Real Solving for Positive Dimensional Systems”. In: *Journal of Symbolic Computation* 34.6 (2002), pp. 543–560.

- [3] B. Bank, M. Giusti, J. Heintz, and G. M. Mbakop. “Polar Varieties and Efficient Real Equation Solving: The Hypersurface Case”. In: *Journal of Complexity* 13.1 (1997), pp. 5–27.
- [4] B. Bank, M. Giusti, J. Heintz, G. Lecerf, G. Matera, and P. Solernó. “Degeneracy Loci and Polynomial Equation Solving”. In: *Foundations of Computational Mathematics* 15 (2015), pp. 159–184.
- [5] S. Basu, R. Pollack, and M.-F. Roy. *Algorithms in Real Algebraic Geometry*. Heidelberg: Springer Berlin, 2006.
- [6] W. Baur and V. Strassen. “The complexity of partial derivatives”. In: *Theoretical Computer Science* 22 (1983), pp. 317–330.
- [7] B. Bérard, S. Haddad, C. Picaronny, M. Safey El Din, and M. Sassolas. “Polynomial interrupt timed automata: Verification and expressiveness”. In: *Information and Computation* 277 (2021). Reachability Problems, p. 104580.
- [8] J. Berthomieu, C. Eder, and M. Safey El Din. “msolve: A Library for Solving Polynomial Systems”. In: *2021 International Symposium on Symbolic and Algebraic Computation*. 46th International Symposium on Symbolic and Algebraic Computation. Saint Petersburg, Russia, July 2021.
- [9] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah. “Julia: A Fresh Approach to Numerical Computing”. In: *SIAM Review* 59.1 (2017), pp. 65–98.
- [10] J. Bochnak, M. Coste, and M.-F. Roy. *Real Algebraic Geometry*. Springer Berlin, Heidelberg, 1998.
- [11] P. Breiding, B. Sturmfels, and K. Wang. “Computing arrangements of hypersurfaces”. In: *Journal of Software for Algebra and Geometry* 15 (Apr. 2025), pp. 11–27.
- [12] D. Chablat, R. Prébet, M. Safey El Din, D. H. Salunkhe, and P. Wenger. “Deciding Cuspidality of Manipulators through Computer Algebra and Algorithms in Real Algebraic Geometry”. In: *Proceedings of the 2022 International Symposium on Symbolic and Algebraic Computation*. ISSAC ’22. Villeneuve-d’Ascq, France: Association for Computing Machinery, 2022, pp. 439–448.
- [13] G. E. Collins. “Quantifier elimination for real closed fields by cylindrical algebraic decomposition”. In: *Automata Theory and Formal Languages*. Ed. by H. Brakhage. Berlin, Heidelberg: Springer Berlin Heidelberg, 1975, pp. 134–183.
- [14] D. Cox, J. Little, and D. O’Shea. “Ideals, Varieties, and Algorithms. An Introduction to Computational Algebraic Geometry and Commutative Algebra”. In: (2007).
- [15] X. Dahan, A. Kadri, and É. Schost. “Bit-size estimates for triangular sets in positive dimension”. In: *Journal of Complexity* 28.1 (2012), pp. 109–135.
- [16] X. Dahan and É. Schost. “Sharp estimates for triangular sets”. In: *Proceedings of the 2004 International Symposium on Symbolic and Algebraic Computation*. ISSAC ’04. Santander, Spain: Association for Computing Machinery, 2004, pp. 103–110.

- [17] D. Eisenbud. *Commutative Algebra With a View Toward Algebraic Geometry*. Vol. 150. Graduate Texts in Mathematics. New York: Springer New York, 1995.
- [18] J. Elliott, M. Giesbrecht, and É. Schost. “Bit complexity for computing one point in each connected component of a smooth real algebraic set”. In: *Journal of Symbolic Computation* 116 (2023), pp. 72–97.
- [19] J. Elliott, M. Giesbrecht, and É. Schost. “On the bit complexity of finding points in connected components of a smooth real hypersurface”. In: *Proceedings of the 45th International Symposium on Symbolic and Algebraic Computation*. ISSAC ’20. Kalamata, Greece: Association for Computing Machinery, 2020, pp. 170–177.
- [20] H. Everett, D. Lazard, S. Lazard, and M. Safey El Din. “The Voronoi diagram of three lines”. In: *Discrete and Computational Geometry* 42.1 (2009), pp. 94–130.
- [21] J.-C. Faugère, G. Moroz, F. Rouillier, and M. Safey El Din. “Classification of the perspective-three-point problem, discriminant variety and real solving polynomial systems of inequalities”. In: *ISSAC 2008 - 21st International Symposium on Symbolic and Algebraic Computation*. Hagenberg, Austria: ACM, July 2008, pp. 79–86.
- [22] E. Feliu and A. Sadeghimanesh. “Kac-Rice formulas and the number of solutions of parametrized systems of polynomial equations”. English. In: *Mathematics of Computation* 91.338 (2022). Publisher Copyright: © 2022 American Mathematical Society, pp. 2739–2769.
- [23] A. Ferguson. “Exact Algorithms for Polynomial Optimisation. (Algorithmes exacts pour l’optimisation polynomiale)”. PhD thesis. Sorbonne Université, Paris, France, 2022.
- [24] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences)*. First Edition. W. H. Freeman, 1979.
- [25] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge: Cambridge University Press, 2003.
- [26] P. M. Gianni and T. Mora. “Algebraic Solution of Systems of Polynomial Equations Using Groebner Bases”. In: *AAECC*. Vol. 356. Lecture Notes in Computer Science. Springer, 1989, pp. 247–257.
- [27] M. Giusti, J. Heintz, K. Hägele, J. E. Morais, L. M. Pardo, and J. L. Montaña. “Lower bounds for diophantine approximation”. In: *Journal of Pure and Applied Algebra* 117/118 (1997), pp. 277–317.
- [28] M. Giusti, J. Heintz, J. E. Morais, J. Morgenstern, and L. M. Pardo. “Straight-line programs in geometric elimination theory”. In: *Journal of Pure and Applied Algebra* 124 (1998), pp. 101–146.

- [29] M. Giusti, J. Heintz, J. E. Morais, and L. M. Pardo. “When polynomial equation systems can be “solved” fast?” In: *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*. Ed. by G. Cohen, M. Giusti, and T. Mora. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, pp. 205–231.
- [30] M. Giusti, G. Lecerf, and B. Salvy. “A Gröbner Free Alternative for Polynomial System Solving”. In: *Journal of Complexity* 17 (2001), pp. 154–211.
- [31] A. Goharshady, S. Hitarth, F. Mohammadi, and H. Motwani. “Algebro-geometric Algorithms for Template-based Synthesis of Polynomial Programs”. In: *Proceedings of the ACM on Programming Languages* 7 (2023), pp. 727–758.
- [32] D. Grigoriev and N. Vorobjov. “Counting connected components of a semialgebraic set in subexponential time”. In: *Computational Complexity* 2 (1992), pp. 133–186.
- [33] D. Grigoriev and N. Vorobjov. “Solving systems of polynomial inequalities in subexponential time”. In: *Journal of Symbolic Computation* 5.1/2 (1988), pp. 37–64.
- [34] J. Heintz. “Definability and fast quantifier elimination in algebraically closed fields”. In: *Theoretical Computer Science* 24.3 (1983), pp. 239–277.
- [35] J. Heintz, M.-F. Roy, and P. Solernó. “Description of the connected components of a semialgebraic set in single exponential time”. In: *Discrete & Computational Geometry* 11 (1994), pp. 121–140.
- [36] A. Ibrahim and B. Salvy. “Positivity Certificates for Linear Recurrences”. In: *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)* (2024), pp. 982–994.
- [37] W. R. Inc. *Mathematica, Version 14.3*. Champaign, IL, 2025.
- [38] G. Jeronimo and J. Sabia. “Computing multihomogeneous resultants using straight-line programs”. In: *Journal of Symbolic Computation* 42.1 (2007). Effective Methods in Algebraic Geometry (MEGA 2005), pp. 218–235.
- [39] E. Kaltofen, Z. Yang, and L. Zhi. “A proof of the monotone column permanent (MCP) conjecture for dimension 4 via sums-of-squares of rational functions”. In: *Proceedings of the 2009 Conference on Symbolic Numeric Computation. SNC '09*. Kyoto, Japan: Association for Computing Machinery, 2009, pp. 65–70.
- [40] T. Krick, L. M. Pardo, and M. Sombra. “Sharp estimates for the arithmetic Nullstellensatz”. In: *Duke Mathematical Journal* 109.3 (2001), pp. 521–598.
- [41] L. Kronecker. “Grundzüge einer arithmetischen Theorie der algebraischen Größen”. In: *Journal für die reine und angewandte Mathematik* 92 (1882), pp. 1–122.
- [42] H. P. Le and M. Safey El Din. “Faster One Block Quantifier Elimination for Regular Polynomial Systems of Equations”. In: *Proceedings of the 2021 International Symposium on Symbolic and Algebraic Computation. ISSAC '21*. ACM, July 2021.
- [43] H. Le and M. Safey El Din. “Solving parametric systems of polynomial equations over the reals through Hermite matrices”. In: *Journal of Symbolic Computation* 112 (2022), pp. 25–61.

- [44] F. Macaulay. *The Algebraic Theory of Modular Systems*. Cambridge University Press, 1916.
- [45] Maplesoft, a division of Waterloo Maple Inc.. *Maple*. Version 2023. Waterloo, Ontario, 2023.
- [46] S. Melczer and B. Salvy. “Effective coefficient asymptotics of multivariate rational functions via semi-numerical algorithms for polynomial systems”. In: *Journal of Symbolic Computation* 103 (2021), pp. 234–279.
- [47] J. Milnor. “On the Betti Numbers of Real Varieties”. In: *Proceedings of the American Mathematical Society* 15.2 (1964), pp. 275–280.
- [48] D. Mumford. *Algebraic Geometry I. Complex Projective Varieties*. Classic in Mathematics. Heidelberg: Springer Berlin, 1995.
- [49] O. Oleinik and I. Petrovski. “On the topology of real algebraic varieties”. In: *Izv. Akad. Nauk SSSR* 13 (1949), pp. 289–302.
- [50] P. Philippon. “Sur des hauteurs alternatives III”. In: *Journal de Mathématiques Pures et Appliquées* 74 (1995), pp. 345–365.
- [51] A. Poteaux and É. Schost. “On the complexity of computing with zero-dimensional triangular sets”. In: *Journal of Symbolic Computation* 50 (2013), pp. 110–138.
- [52] F. Rouillier. “Solving Zero-Dimensional Systems Through the Rational Univariate Representation”. In: *Applicable Algebra in Engineering, Communication and Computing* 9.5 (1999), pp. 433–461.
- [53] M. Safey El Din. “Testing Sign Conditions on a Multivariate Polynomial and Applications”. In: *Mathematics in Computer Science* 1 (2007), pp. 177–207.
- [54] M. Safey El Din and É. Schost. “Bit complexity for multi-homogeneous polynomial system solving— Application to polynomial minimization”. In: *Journal of Symbolic Computation* 87 (2018), pp. 176–206.
- [55] M. Safey El Din and É. Schost. “Polar varieties and computation of one point in each connected component of a smooth real algebraic set”. In: *ISSAC 2003 - 28th International Symposium on Symbolic and Algebraic Computation*. ISSAC '03. Philadelphia, PA, USA: Association for Computing Machinery, 2003, pp. 224–231.
- [56] M. Safey El Din and É. Schost. “A Nearly Optimal Algorithm for Deciding Connectivity Queries in Smooth and Bounded Real Algebraic Sets”. In: *Journal of the ACM* 63.6 (Jan. 2017).
- [57] The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 9.5)*. 2022.
- [58] J. T. Schwartz. “Fast Probabilistic Algorithms for Verification of Polynomial Identities”. In: *J. ACM* 27.4 (Oct. 1980), pp. 701–717.
- [59] I. R. Shafarevich. *Basic algebraic geometry; 3rd ed.* Berlin: Springer, 2013.
- [60] F. Sottile. “Real Schubert Calculus: Polynomial Systems and a Conjecture of Shapiro and Shapiro”. In: *Experimental Mathematics* 9 (1999), pp. 161–182.

- [61] A. Strzebonski and E. Tsigaridas. “Univariate real root isolation in an extension field and applications”. In: *Journal of Symbolic Computation* 92 (2019), pp. 31–51.
- [62] A. Tarski. *A decision method for elementary algebra and geometry*. 1951.
- [63] R. Thom. “Sur l’homologie des variétés algébriques réelles”. In: *Differential and Combinatorial Topology*. Princeton Mathematical Series Princeton Mathematical Series (1965), pp. 255–265.