An Improved Complexity Bound for Computing the Topology of a Real Algebraic Space Curve

Jin-San Cheng, Kai Jin, Marc Pouget, Junyi Wen, Bingwei Zhang

An Improved Complexity Bound for Computing the Topology of a Real Algebraic Space Curve

Jin-San Cheng[♭,♯,1], Kai Jin[♮], Marc Pouget[◇], Junyi Wen[♭,♯], Bingwei Zhang[♭,♯]
♭ KLMM, Institute of Systems Science, Academy of Mathematics and Systems Science, CAS
♯School of Mathematical Sciences, University of Chinese Academy of Sciences
♮School of Mathematics and Statistics, Hubei University of Science and Technology, Xianning 437100, China
◇ Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
jcheng@amss.ac.cn, jinkaijl@163.com, marc.pouget@inria.fr, wenjunyi15@mails.ucas.ac.cn, zhangbingwei18@mails.ucas.ac.cn

Abstract

We propose a new algorithm to compute the topology of a real algebraic space curve. The novelties of this algorithm are a new technique to achieve the lifting step which recovers points of the space curve in each plane fiber from several projections and a weaken notion of generic position. As opposed to previous work, our *sweep generic position* does not require that x-critical points have different x-coordinates. The complexity of achieving this sweep generic position is thus no longer a bottleneck in term of complexity. The bit complexity of our algorithm is $\widetilde{\mathcal{O}}(d^{18} + d^{17}\tau)$ where d and τ bound the degree and the bitsize of the integer coefficients of the defining polynomials of the curve and polylogarithmic factors are ignored. To the best of our knowledge, this improves upon the best currently known results at least by a factor of d^2.

Keywords: Real algebraic space curve, Topology, Bit complexity.

1. Introduction

Algebraic curves are widely used in computer aided geometric design and geometric modeling. For example, implicit surface-surface intersection computation is an important topic in computer aided geometric design, which is strongly related to algebraic space curves. Algebraic curves are also basic research objects in classical algebraic geometry. One basic task in the study of an algebraic curve is to compute its topology. Computing the topologies of an algebraic plane curve and an algebraic space curve are fundamental steps to compute the topology of an algebraic surface [10, 16]. Furthermore, some control problems need implicitly the topology information of algebraic (space) curves [28].

Previous work. There have been many papers studying the computation of the topology of an algebraic plane curve, see for instance [1, 3, 5, 8, 9, 13, 18, 19, 22, 23, 26, 31], but only a few studying the topology of an algebraic space curve [2, 12, 14, 20, 21, 24]. Almost all the existing work [2, 12, 14, 20, 24] computing the topology of an algebraic space curve require the curve to be in a so-called generic position. Although the definitions of generic positions vary in the literature, they all include the condition that the x-critical points have different x-coordinates. Checking whether an algebraic space curve is in generic position is not a trivial task, and finding a shearing of the coordinate system so that the sheared curve is in generic position is a complexity bottleneck [14, 25]. One can use the CAD method to compute the topology of algebraic space curve without computing generic position, but the CAD method requires many real solving of triangular sets in the lifting step, which is a bottleneck of the method, as explained by Lazard [27].

The complexity for computing the topology of an algebraic plane curve is well studied, the record bound is $\tilde{\mathcal{O}}(d^6 + d^5\tau)$ for an input polynomial of degree d and bitsize τ [15, 26]. Only few results are known for the complexity of computing the topology of an algebraic space curve given by polynomials of degree d and bitsize τ. Diatta et. al [16] present a bit complexity of $\tilde{\mathcal{O}}(d^{21}\tau)$ under the assumption that the input space curve is in generic position. Cheng et. al [12] propose a method without a generic position hypothesis with complexity $\tilde{\mathcal{O}}(d^{37}\tau)$. Jin and Cheng [25] give a bit complexity of $\tilde{\mathcal{O}}(d^{20} + d^{19}\tau)$ via the computation of a strong generic position [11].

[1]Corresponding author

Contributions. Let the space curve $\mathcal{C}$ be defined by $\{(x, y, z) \in \mathbb{R}^3, f(x, y, z) = g(x, y, z) = 0\}$ with f and g integer polynomials of total degree at most d. Our algorithm follows a classical projection/lifting method followed by a sweep plane connection. We denote by $h_0(x, y)$ the square-free part of the resultant of f and g with respect to z and by $r(x)$ the resultant of $h_0(x, y)$ and $\frac{\partial h_0(x,y)}{\partial y}$ with respect to y. The curve $\{h_0(x, y) = 0\}$ is thus the projection of $\mathcal{C}$ on the xy-plane and $r(x)$ encodes its x-critical points.

For the lifting step, we compute the $\mathcal{O}(d^2)$ space points of $\mathcal{C}$ on the fibers $\{x = \alpha\}$, where α is a real root of $r(x)$. Our new lifting algorithm uses the following subroutine (Section 3.3.1): given n points with two coordinates but providing only the first coordinates and n linear combinations of the coordinates of these points, one can recover the second coordinates of these n points. To apply this subroutine in a fiber $x = \alpha$, we consider the linear coordinate transformations $\phi_s : (\alpha, y, z) \to (\alpha, y + sz, z)$ for d^2 different values of s. The d^2 linear combinations of the y and z coordinates, $y - sz$, are then given by solving the triangular systems $\{r(x), h_s(x, y)\}$, where h_s is the resultant of the polynomials $f \circ \phi_s$ and $g \circ \phi_s$ with respect to the z-variable (Section 3.3.2).

To ease our sweep-plane connection algorithm in the x-direction, we require that the curve $\mathcal{C}$ has (a) no asymptote in the z-direction, (b) a finite number of points on any x-fiber plane $\{x = \alpha\}$ for $\alpha \in \mathbb{R}$. We show how to shear the coordinate system to achieve this *sweep generic position* (Section 3.1). It is worth noting that we do not require that the x-fibers contain only one x-critical point as in classical generic positions. The complexity of achieving this sweep generic position is thus no longer a bottleneck in term of complexity. The connections between two fibers are recovered using two different plane projections of $\mathcal{C}$ to solve possible ambiguities when two space components project to the same plane component.

Our method does not need the curve to be reduced, that is, the ideal generated by f and g needs not to be radical, our only assumption is that f and g are coprime. This is important for computing the topology of a surface since its polar curve is, in general, not reduced [16].

Based on the state-of-the-art complexity result in [15] for isolating bivariate triangular systems and computing the topology of algebraic plane curves, we analyze the bit complexity of our algorithm for computing the topology of an algebraic space curve. The bit complexity of our algorithm is $\tilde{\mathcal{O}}(d^{18} + d^{17}\tau)$, where d and τ are the degree bound and the bit size bound of the coefficients of the defining polynomials of the algebraic space curve. To the best of our knowledge, this improves upon the former results by at least a factor of d^2.

2. Notation and preliminaries

Let $\mathbb{R}$ and $\mathbb{C}$ be the fields of real and complex numbers, and let $\mathbb{Z}$ be the ring of integers. For a polynomial $P(x)$ in $R[x]$ with R a ring, the leading coefficient of P with respect to x is denoted $\mathrm{Lc}_x(P)$. Let $h(x, y) \in \mathbb{Z}[x, y]$, we denote the algebraic plane curve defined by $\{h = 0\}$ as $\mathcal{C}_h$. Let p be a point on $\mathcal{C}_h$, we call p an **x-critical point** if $h(p) = \partial_y h(p) = 0$, and a **singular point** if $h(p) = \partial_x h(p) = \partial_x h(p) = 0$, where $\partial_x h$ and $\partial_y h$ are the partial derivatives with respect to x and y.

We always use $\mathcal{C}$ to denote an algebraic space curve defined by two coprime polynomials $f(x, y, z)$ and $g(x, y, z)$ in $\mathbb{Z}[x, y, z]$, that is $\mathcal{C} = \{(x, y, z) \in \mathbb{R}^3 | f(x, y, z) = g(x, y, z) = 0\}$ with $\gcd(f, g) = 1$. We denote by d the maximum of the total degrees of f and g. We call **x-fiber** a plane of equation $x = \alpha$ for $\alpha \in \mathbb{R}$.

The curve $\mathcal{C}$ is called **reduced** when the ideal generated by f and g is radical. Our algorithm does not require that $\mathcal{C}$ is reduced and does not compute the radical. The singularities of a variety are geometric features so they are naturally defined from the ideal of the variety. Let the ideal of the curve $\mathcal{C}$ be generated by the polynomials $(f_i)_{i=1,\dots,n}$ and let J be the Jacobian matrix of the f_i, that is, its rows are the gradients of the f_i. A point P of $\mathcal{C}$ is called **regular** if J has rank 2 at P. A point of $\mathcal{C}$ which is not regular is called a **singular point or singularity**. A point P of $\mathcal{C}$ is called **x-critical** if it is either singular or it is regular and the tangent line of $\mathcal{C}$ at P is in an x-fiber. Note that the tangent line is the kernel of the Jacobian matrix. The x-coordinates of the x-critical points give all the x-fiber planes one may have to consider for a sweep plane algorithm. On the other hand, when projecting the space curve $\mathcal{C}$ to the plane curve $\pi(\mathcal{C})$, an x-critical point of $\mathcal{C}$ may no longer be an x-critical point of $\pi(\mathcal{C})$, where the projection map is

$$\pi : \begin{array}{ccc} \mathbb{R}^3 & \longrightarrow & \mathbb{R}^2 \\ (x, y, z) & \longrightarrow & (x, y). \end{array}$$

To better understand this fact, the geometric characterization of a singularity in terms of intersection of local branches is useful. Geometrically, a regular point of a space curve is a point where the curve has a well defined tangent line given by the kernel of the Jacobian matrix J. For a singular point, there are at least 2 (maybe complex) curve branches passing through the point. An x-critical point of $\mathcal{C}$ is called **cylindrical** if it is not an x-critical point of the plane curve $\mathcal{C}_{h_0}$, where $h_0 = \mathrm{Squarefree}(\mathrm{Res}_z(f, g))$. Note that $\pi(\mathcal{C}) \subset \mathcal{C}_{h_0}$ and the equality holds when the

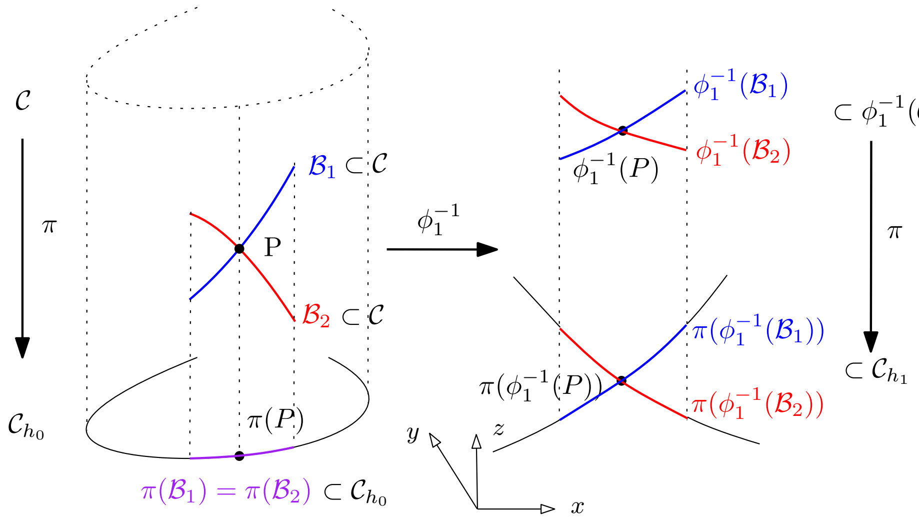

Figure 1: P is a cylindrical singular point of $\mathcal{C}$, its image, $\phi_s^{-1}(P)$, on the sheared curve projects to $\pi(\phi_s^{-1}(P))$ which is a singular point of $\mathcal{C}_{h_1}$.

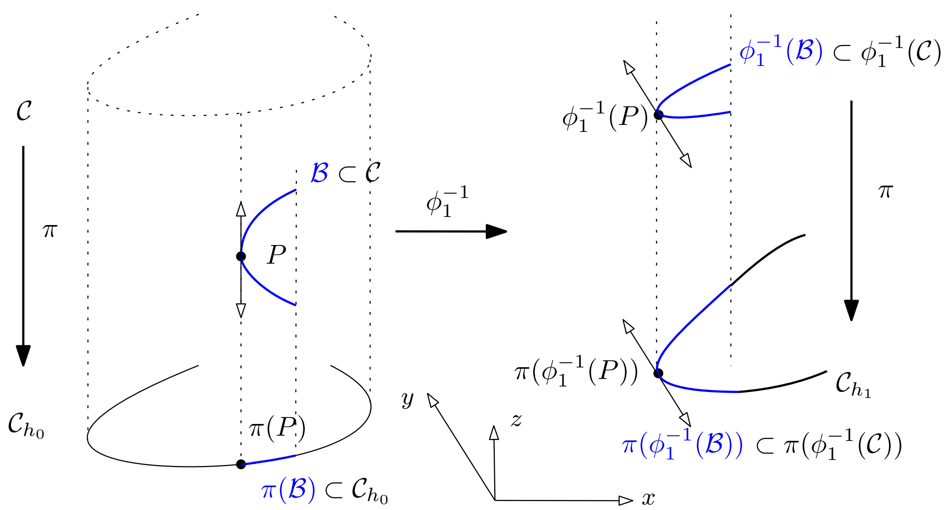

Figure 2: P is a regular cylindrical x-critical point of $\mathcal{C}$, its image, $\phi_s^{-1}(P)$, on the sheared curve projects to $\pi(\phi_s^{-1}(P))$ which is a regular x-critical point of $\mathcal{C}_{h_1}$.

leading coefficients of f and g with respect to z do not vanish simultaneously and one considers the complex points projecting to real points. A cylindrical singular point P of $\mathcal{C}$ occurs when all branches that intersect at P have the same projection and there is no other branch of $\mathcal{C}$ projecting to another branch of $\mathcal{C}_{h_0}$ passing through $\pi(P)$, see Figure 1. A cylindrical regular x-critical point of $\mathcal{C}$ occurs when the branch of $\mathcal{C}$ passing through P has a tangent line in the z-direction, and there is no other branch of $\mathcal{C}$ projecting to another branch of $\mathcal{C}_{h_0}$ passing through $\pi(P)$, see Figure 2. Lemma 3.3 shows that all x-critical points of $\mathcal{C}$ are witnessed either as x-critical points of the projection $\mathcal{C}_{h_0}$ or as x-critical points of $\mathcal{C}_{h_1}$, which is the projection of a shearing of $\mathcal{C}$. Our algorithm uses these two plane curves to identify all the x-critical points of $\mathcal{C}$. We define the shearing function:

$$\phi_s : \begin{array}{ccc} \mathbb{R}^3 & \longrightarrow & \mathbb{R}^3 \\ (x, y, z) & \longrightarrow & (x, y + sz, z) \end{array} \tag{1}$$

The sheared curve of $\mathcal{C}$ by the shear ϕ_s is defined by $\phi_s^{-1}(\mathcal{C}) = \{(x, y, z) \,|\, f \circ \phi_s(x, y, z) = g \circ \phi_s(x, y, z) = 0\} = \{(x, y - sz, z) \,|\, (x, y, z) \in \mathcal{C}\} = \phi_{-s}(\mathcal{C})$.

Algorithm 1: SGP: Sweep generic position

Input : $f, g \in \mathbb{Z}[x, y, z]$ coprime defining a curve $\mathcal{C}$.

Output: $\tilde{f}, \tilde{g} \in \mathbb{Z}[x, y, z]$ coprime defining a curve isotopic to $\mathcal{C}$ that has (a) no asymptote in the z-direction, (b) a finite number of points on any x-fiber plane $\{x = \alpha\}$ for $\alpha \in \mathbb{R}$.

1 Let $f = \sum_{0 \le i+j+k \le d} c_{i,j,k} x^i y^j z^k$, evaluate the bivariate polynomial $\sum_{0 \le i+j \le d} c_{i,j,d-i-j} u^i v^j$ on the integer grid $\{0, \dots, d\} \times \{0, \dots, d\}$, choose (α, β) that does not vanish it.
2 $\hat{f}(x, y, z) = f(x + \alpha z, y + \beta z, z)$, $\hat{g}(x, y, z) = g(x + \alpha z, y + \beta z, z)$.
3 Compute $h(x, y) = \mathrm{Res}_z(\hat{f}(x, y, z), \hat{g}(x, y, z)) = \sum_{i=0,\dots,2d^2} c_i(x) y^i$.
4 **if** $\gcd(c_0(x), \dots, c_{2d^2}(x))$ *is a constant* **then**
5 $\quad \gamma = 0$
6 **else**
7 $\quad$ Let $h(x, y) = \sum_{0 \le i+j \le d'} c_{i,j} x^i y^j$ with d' the total degree of h. Evaluate the univariate polynomial $\sum_{0 \le i \le d'} c_{i,d'-i} s^i$ at integer values starting from 0 until an integer $\gamma \le d'$ does not vanish it.
8 **return** $\tilde{f}(x, y, z) = \hat{f}(x + \gamma y, y, z)$, $\tilde{g}(x, y, z) = \hat{g}(x + \gamma y, y, z)$.

3. Algorithm

In this section, we detail our deterministic algorithm for computing the topology of the space curve $\mathcal{C}$ given by the coprime polynomials f and g in $\mathbb{Z}[x, y, z]$. Our algorithm follows a classical projection/lifting method followed by a sweep plane connection:

1. Shear the coordinate system such that the space curve is in a sweep generic position.
2. Projection: Project the space curve $\mathcal{C}$ onto the xy-plane and compute the topology of the plane curve $\mathcal{C}_{h_0}$, where $h_0(x, y) = \mathrm{Squarefree}(\mathrm{Res}_z(f, g))$.
3. Lifting: Lift some plane points of $\mathcal{C}_{h_0}$ to obtain the corresponding space points of $\mathcal{C}$.
4. Connection: Connect the space points by line segments to recover the topology of $\mathcal{C}$.

The topology of the curve $\mathcal{C}$ is then implicitly encoded by an embedded graph isotopic to $\mathcal{C}$ in $\mathbb{R}^3$ whose vertices are the computed space points in the x-fibers, and edges are straight line segments connecting the vertices. We explain each step in detail in the following sections.

3.1. Sweep generic position

To ease our sweep-plane algorithm in the x-direction, we require that the curve $\mathcal{C}$ has (a) no asymptote in the z-direction, (b) a finite number of points on any x-fiber plane $\{x = \alpha\}$ for $\alpha \in \mathbb{R}$. We show how to achieve these requirements by applying shearings of the coordinate system as detailed in Algorithm 1. We simply denote the algorithm as $(\tilde{f}, \tilde{g}) = \mathbf{SGP}(f, g)$. Since such shearings do not change the topology of the curve, we then compute the topology of the sheared curve. Note that shearings preserve the fact that the polynomials remain coprime. The proof of correctness of Algorithm 1 follows from the following lemmas.

Lemma 3.1. *The polynomials $\tilde{f}$ and $\tilde{g}$ output by Algorithm 1 are such that (a) the leading coefficient of $\tilde{f}$ with respect to z is a non-zero constant and (b) the resultant of $\tilde{f}$ and $\tilde{g}$ with respect to z has no factor containing only the variable x.*

Proof. Writing $f = \sum_{0 \le i+j+k \le d} c_{i,j,k} x^i y^j z^k = \sum_{0 \le i+j \le d} c_{i,j,d-i-j} x^i y^j z^{d-i-j} + \sum_{0 \le i+j+k < d} c_{i,j,k} x^i y^j z^k$ yields that the leading coefficient of $f(x + uz, y + vz, z)$ with respect to z is $L(u, v) = \sum_{0 \le i+j \le d} c_{i,j,d-i-j} u^i v^j$, which is not the null polynomial since there exists one $c_{i,j,d-i-j} \neq 0$. If for all $\alpha \in \{0, \dots, d\}$, the univariate polynomial $L(\alpha, v)$ is the null polynomial, then the polynomial $\Pi_{\alpha=0}^{\alpha=d}(u - \alpha)$, which is of degree $d + 1$, divides $L(u, v)$. This is in contradiction with the fact that L has degree at most d in u. Hence there exists $\alpha \in \{0, \dots, d\}$ such that the univariate polynomial $L(\alpha, v)$ is not the null polynomial, it is of degree at most d in v, so that there exists $\beta \in \{0, \dots, d\}$ that does not vanish it. Hence, the choice of (α, β) ensures that $\mathrm{Lc}_z(\hat{f})$ is a constant and since the second shear of Line 8 does not modify this leading, $\mathrm{Lc}_z(\tilde{f})$ is also a constant.

For point (b), if $\gamma = 0$ then $\tilde{f} = \hat{f}$ and $\tilde{g} = \hat{g}$ thus h is the resultant of $\tilde{f}$ and $\tilde{g}$ and it has no factor containing only the variable x according to the condition of Line 4.

If $\gamma \neq 0$ then Line 7 ensures that $\mathrm{Lc}_y(h(x+\gamma y, y))$ is a non-zero constant, which implies that $h(x+\gamma y, y) = \mathrm{Res}_z(\tilde{f}(x,y,z), \tilde{g}(x,y,z))(x+\gamma y, y)$ has no factor containing only the variable x. By the specialization property of the resultant, $\mathrm{Res}_z(\tilde{f}(x,y,z), \tilde{g}(x,y,z))(x,y) = \mathrm{Res}_z(\hat{f}(x,y,z), \hat{g}(x,y,z))(x+\gamma y, y)$, thus $\mathrm{Res}_z(\tilde{f}(x,y,z), \tilde{g}(x,y,z))(x,y)$ has no factor containing only the variable x, which is condition (b). □

The next lemma shows that the properties of the output of Algorithm 1 established in Lemma 3.1 are sufficient conditions for the correctness of the algorithm.

Lemma 3.2. *If the leading coefficient of f with respect to z is a non-zero constant and the resultant of f and g with respect to z has no factor containing only the variable x, then $\mathcal{C}$ has (a) no asymptote in the z direction and (b) a finite number of points on any x-fiber.*

Proof. An asymptote in the z-direction is a solution to the bivariate system $\{\mathrm{Lc}_z(f), \mathrm{Lc}_z(g)\}$ where $\mathrm{Lc}_z(T)$ is the leading coefficient of $T \in \mathbb{Z}[x,y,z]$ with respect to z. The assumption that $\mathrm{Lc}_z(f)$ a non-zero scalar is thus a sufficient condition to avoid such asymptotes.

For point (b), by contradiction, assume that there exists a fiber $x = \alpha$ which is not finite, that is the system $\{f(\alpha,y,z), g(\alpha,y,z)\}$ is not 0-dimensional. In other words $f(\alpha,y,z)$ and $g(\alpha,y,z)$ have a non-constant gcd $G(y,z)$ in $\mathbb{C}[y,z]$. Let f_1 and g_1 in $\mathbb{C}[y,z]$ be defined such that $f(\alpha,y,z) = G(y,z)f_1(y,z)$ and $g(\alpha,y,z) = G(y,z)g_1(y,z)$.

If the degree of G with respect to z is 0, then its degree with respect to y is non-zero. Let β be a root of this $G(y)$, then $f(\alpha,\beta,z) = G(\beta)f_1(\beta,z) = 0$ so that (α,β) is a common solution to all the coefficients of f with respect to z and in particular its leading term. This is in contradiction with the hypothesis that this leading coefficient is a constant. One thus has that the degree of G with respect to z is at least 1. The polynomials $f(\alpha,y,z)$ and $g(\alpha,y,z)$ thus have a common factor depending on z which implies that their resultant with respect to z vanishes. On the other hand, since α does not vanish the leading coefficient of f with respect to z (which is a constant), the specialization property of the resultant yields that for a constant c:

$$\mathrm{Res}_z(f(x,y,z), g(x,y,z))(\alpha, y) = c\,\mathrm{Res}_z(f(\alpha,y,z), g(\alpha,y,z))(y) = 0.$$

The bivariate polynomial $\mathrm{Res}_z(f(x,y,z), g(x,y,z)) \in \mathbb{Z}[x,y]$ identically vanishes at α, thus, $m(x)$, the minimal polynomial of α divides all its coefficients with respect to y. The polynomial $m(x)$ is thus a factor of the resultant $\mathrm{Res}_z(f(x,y,z), g(x,y,z))$. This contradicts the hypothesis and concludes the proof. □

3.2. Bivariate triangular system solving and plane curve topology

As subroutines of our algorithm, we solve bivariate triangular systems and compute the topology of plane curves. There are many algorithms for computing the topology of plane curves, however, most of them shear the coordinate system which is not desirable for our algorithm. To avoid such a change of coordinates, we use the recent results of [15] that achieve the best complexity for computing the topology of a plane curve in its original coordinate system. Also in [15], an algorithm to solve triangular bivariate systems is given with a detailed complexity analysis taking into account the degrees and bitsizes of both polynomials of the system. This latter complexity is critical for using amortization in the analysis of our algorithm.

3.3. Lifting: computing the space key points on $\mathcal{C}$

In this step, we compute space points of $\mathcal{C}$ on x-fibers $\{x = \alpha\}$, from the plane projections of $\mathcal{C}$ and shearings of $\mathcal{C}$, where α is a real roots of $r^*(x)$ defined in (3). We call these space points of $\mathcal{C}$ the **space key points of $\mathcal{C}$**. We will use d^2 plane curves $\mathcal{C}_{h_m}$ defined, for the integers $(s_m)_{0 \le m \le d^2}$, by the polynomials

$$h_m(x,y) = \mathrm{Squarefree}(\mathrm{Res}_z(f \circ \phi_{s_m}, g \circ \phi_{s_m})) = \mathrm{Squarefree}(\mathrm{Res}_z(f(x, y+s_m z, z), g(x, y+s_m z, z))). \quad (2)$$

In the following, we assume $s_m = m$ for $0 \le m \le d^2$ but keep the variable s_m to emphasize that it is a shear parameter. With lcm standing for the least common multiple, we define

$$r_t(x) = \mathrm{Res}_y(h_t, \partial_y h_t), \text{ for } t = 0 \text{ or } 1, \quad r^*(x) = \mathrm{lcm}(r_0(x), r_1(x)). \quad (3)$$

Lemma 3.3 shows that all x-critical points of $\mathcal{C}$ are witnessed either as x-critical points of the projection $\mathcal{C}_{h_0}$ or as x-critical points of $\mathcal{C}_{h_1}$.

Lemma 3.3. *Let P be an x-critical point of $\mathcal{C}$, then $\pi(P)$ is an x-critical point of $\mathcal{C}_{h_0}$ or $\pi(\phi_1^{-1}(P))$ is an x-critical point of $\mathcal{C}_{h_1}$, where $h_1 = \mathrm{Squarefree}(\mathrm{Res}_z(f \circ \phi_1, g \circ \phi_1))$.*

Proof. Assume $\pi(P)$ is not an x-critical point of $\mathcal{C}_{h_0}$. The point P is thus a cylindrical x-critical point of $\mathcal{C}$.

In the case where P is a cylindrical singular point of $\mathcal{C}$, any two distinct branches passing through P are mapped on the sheared curve $\phi_1^{-1}(\mathcal{C})$ on two branches those projections by π only intersect at $\pi(\phi_1^{-1}(P))$, see Figure 1. This implies that $\pi(\phi_1^{-1}(P))$ is a singular point of $\mathcal{C}_{h_1}$.

In the other case, P is a cylindrical regular x-critical point, the tangent line to $\mathcal{C}$ at $P = (P_x, P_y, P_z)$ has equation $\{x = P_x, y = P_y\}$, so that the sheared curve $\phi_1^{-1}(\mathcal{C})$ has equation $\{x = P_x, y + z = P_y\}$ that projects by π to the line $\{x = P_x\}$ on the plane, see Figure 2. This line is thus the tangent line to a branch of $\mathcal{C}_{h_1}$ at $\pi(\phi_1^{-1}(P))$. If there is no other branch of $\phi_1^{-1}(\mathcal{C})$ that projects to a branch of $\mathcal{C}_{h_1}$ passing through $\pi(\phi_1^{-1}(P))$, this point is a regular x-critical point of $\mathcal{C}_{h_1}$. If there are other branches of $\phi_1^{-1}(\mathcal{C})$ that projects to branches of $\mathcal{C}_{h_1}$ passing through $\pi(\phi_1^{-1}(P))$, this point is a singular x-critical point of $\mathcal{C}_{h_1}$.

In both cases, $\pi(\phi_1^{-1}(P))$ is an x-critical point of $\mathcal{C}_{h_1}$, which concludes the proof. □

This lemma thus yields the following corollary.

Corollary 3.4. *The x-coordinates of all the x-critical points of $\mathcal{C}$ are zeros of r^*.*

On the other hand, some zeros of r^* do not witness x-critical points of $\mathcal{C}$ but only x-critical points of $\mathcal{C}_{h_0}$ or $\mathcal{C}_{h_1}$. For instance, when two distinct branches of $\mathcal{C}$ intersect in projection and thus generate a singular point in the projection, this point is sometimes called an apparent singularity [14]. All the x-fibers given by zeros of r^* together with intermediate fibers are enough to recover the topology of $\mathcal{C}$ from the topologies of the two plane curves $\mathcal{C}_{h_0}$ and $\mathcal{C}_{h_1}$ as explained in Section 3.4.

3.3.1. Recovering points from multiple projections

Our lifting step is based on the following combinatorial observation and its corollary. We use in this section the function

$$\begin{array}{rccc} \varphi_s : & \mathbb{R}^2 & \longrightarrow & \mathbb{R}^2 \\ & (x, y) & \longrightarrow & (x + sy, y) \end{array}$$

Lemma 3.5. *Let $P = \{(\alpha_1, \beta_{1,1}), \ldots, (\alpha_1, \beta_{1,n_1}), (\alpha_2, \beta_{2,1}), \ldots, (\alpha_2, \beta_{2,n_2}), \ldots, (\alpha_l, \beta_{l,1}), \ldots, (\alpha_l, \beta_{l,n_l})\} \subset \mathbb{C}^2$ with cardinality $\sum_{i=1}^n n_i = n$. For $s \in \mathbb{C} \setminus 0$, define $\varphi_s(P) = \{p_x + sp_y | (p_x, p_y) \in P\}$ and $T_{s,j} = \{\frac{a-\alpha_j}{s} | a \in \varphi_s(P)\}$. For any $m > n - \min_i\{n_i\}$ nonzero distinct $s_1, \ldots, s_m \in \mathbb{C}$, one has*

$$\bigcap_{i=1}^{m} T_{s_i,j} = \{\beta_{j,1}, \ldots, \beta_{j,n_j}\}, \forall j \in \{1, \ldots, l\}.$$

Proof. It is enough to prove the case $j = 1$, that is, $\bigcap_{i=1}^m T_{s_i,1} = \{\beta_{1,1}, \ldots, \beta_{1,n_1}\}$, since the other cases can be proved in a similar way.

Denote $Y = \{\beta_{1,1}, \cdots, \beta_{1,n_1}\}$. One has $\forall t \in \{1, \ldots, n_1\}, \forall m \in \{1, \ldots, m\}, \alpha_1 + s_i\beta_{1,t} \in \varphi_{s_i}(P)$, and $\frac{(\alpha_1 + s_i\beta_{1,t}) - \alpha_1}{s_1} = \beta_{1,t} \in T_{s_i,1}$, thus $Y \subset \bigcap_{i=1}^m T_{s_i,1}$.

Next, we prove $\bigcap_{i=1}^m T_{s_i,1} \subset Y$. Assume that $\bigcap_{i=1}^m T_{s_i,1} \nsubseteq Y$, then there exists $\beta^* \in \bigcap_{i=1}^m T_{s_i,1}$ and $\beta^* \notin Y$. Therefore we have $\forall i \in \{1, \ldots, m\}, \exists t_i \in \{1, \ldots, l\}, k_i \in \{1, \ldots, n_i\}$ s.t. $\frac{(\alpha_{t_i} + s_i\beta_{t_i,k_i}) - \alpha_1}{s_i} = \beta^*$, i.e.,

$$\alpha_{t_i} + s_i\beta_{t_i,k_i} = \alpha_1 + s_i\beta^*. \tag{4}$$

We claim that $\forall i \in \{1, \ldots, m\}, t_i \neq 1$. Otherwise, if $t_i = 1$, we can get $\alpha_1 + s_i\beta_{1,k_1} = \alpha_1 + s_i\beta^*$ from (4), i.e., $\beta^* = \beta_{1,k_1} \in Y$. It is a contradiction with $\beta^* \notin Y$. Hence, we have that $i \in \{1, \ldots, m\}$ has m choices and (t_i, k_i) has only $n - n_1$ choices. By assumption $m > n - n_1$, thus, there exist two different values $s_{i_1} \neq s_{i_2}$, $1 \leq i_1 \neq i_2 \leq m$ and the same values t_{i^*}, k_{i^*}, $1 < t_{i^*} \leq m, 1 \leq k_{i^*} \leq n_{t_{i^*}}$ s.t.

$$\alpha_{t_{i^*}} + s_{i_1}\beta_{t_{i^*},k_{i^*}} = \alpha_1 + s_{i_1}\beta^*. \tag{5}$$
$$\alpha_{t_{i^*}} + s_{i_2}\beta_{t_{i^*},k_{i^*}} = \alpha_1 + s_{i_2}\beta^*. \tag{6}$$

Subtracting Equation (6) from Equation (5), we have $(s_{i_1} - s_{i_2})\beta_{t_{i^*},k_{i^*}} = (s_{i_1} - s_{i_2})\beta^*$, i.e., $\beta^* = \beta_{t_{i^*},k_{i^*}}$. Substituting $\beta^* = \beta_{t_{i^*},k_{i^*}}$ into Equation (5), we get $\alpha_{t_{i^*}} = \alpha_1$. This results contradicts the condition $1 < t_{i^*}$. Therefore, we have $\bigcap_{i=1}^m T_{s_i,1} \subset Y$. □

Algorithm 2: Space key points

Input : The plane key points on the fiber $x = x_i$ of $\mathcal{C}_{h_m}$: $(x_i, y^m_{i,j}), 1 \le j \le l^m_i$, $0 \le m \le d^2$.
Output: The space key points on the fiber $x = x_i$ of $\mathcal{C}$: $P_{i,j,k}(x_i, y_{i,j}, z_{i,j,k})$, where $y_{i,j} = y^0_{i,j}$, $1 \le j \le l^0_i = l_i$, $1 \le k \le l_{i,j}$.

Compute $t_{j,m} = y^0_{i,j}/s_m$ for $1 \le j \le l^0_i, 1 \le m \le d^2$.
Compute $R_m = \{y^m_{i,j}/s_m, 1 \le j \le l^m_i\}$ for $1 \le m \le d^2$.
Compute $T_{s_m,j} := \{t_{j,m} - R_m\} = \{t_{j,m} - a \,|\, \forall a \in R_m\}$ for $1 \le j \le l^0_i, 1 \le m \le d^2$.
Compute $\bigcap_{1 \le m \le d^2} T_{s_m,j} = [z_{i,j,1}, \ldots, z_{i,j,k}, \ldots, z_{i,j,l_{i,j}}]$ for $1 \le j \le l^0_i$.
return *The point set* $\{P_{i,j,k}(x_i, y_{i,j}, z_{i,j,k}), 1 \le j \le l^0_i, 1 \le k \le l_{i,j}\}$.

Corollary 3.6. *Assume that the point set P has at most n distinct points in $\mathbb{C}^2$ and that only their first coordinates are known. If for $m\,(\ge n)$ given distinct nonzero $s_1, \cdots, s_m \in \mathbb{C}$, one knows the sets $\varphi_{s_i}(P) = \{p_x + s_i p_y \,|\, (p_x, p_y) \in P\}$ for $i = 1 \ldots m$, then one can recover the second coordinates of all the points in P.*

Proof. Using the notation of Lemma 3.5 for the point set P, it is enough to note that $m \ge n > n-1 \ge n - \min_i n_i$. The second coordinates are recovered via the construction of the sets T_{s_i}, j. □

We give an example to illustrate the theorem and its corollary.

Example 3.7. *Assume that $P = \{(1,1), (1,2), (2,2)\}$ but we know only their first coordinates $\{1,2\}$. We notice that the total number of points in P is 3, so we set $s_1 = 1, s_2 = 2, s_3 = 3$ and compute*

$$\begin{aligned}
\varphi_{s_1}(P) = \{p_x + s_1 p_y \,|\, (p_x, p_y) \in P\} &= \{2,3,4\}, \\
\varphi_{s_2}(P) = \{p_x + s_2 p_y \,|\, (p_x, p_y) \in P\} &= \{3,5,6\}, \\
\varphi_{s_3}(P) = \{p_x + s_3 p_y \,|\, (p_x, p_y) \in P\} &= \{4,7,8\}.
\end{aligned}$$

Next we begin to recover the second coordinates. We compute

$$T_{s_1,1} = \{1,2,3\}, T_{s_2,1} = \{1,2,2.5\}, T_{s_3,1} = \{1,2,7/3\} \text{ and } T_{s_1,1} \cap T_{s_2,1} \cap T_{s_3,1} = \{1,2\}.$$

Hence, we get the two points $(1,1), (1,2) \in P$ with first coordinate 1. Similarly, we compute

$$T_{s_1,2} = \{0,1,2\}, T_{s_2,2} = \{0.5,1.5,2\}, T_{s_3,2} = \{2/3,5/3,2\} \text{ and } T_{s_1,2} \cap T_{s_2,2} \cap T_{s_3,2} = \{2\}.$$

The point with first coordinate 2 of P is thus $(2,2) \in P$. Hence, we recover the second coordinates of the points in P.

3.3.2. Space key points: Algorithm 2

Based on the above results, we show how to recover the space key points with plane key points on the curves $\mathcal{C}_{h_m}$. The first step is to solve the triangular systems $\{r^*(x), h_m(x,y)\}$ for $0 \le m \le d^2$. Let $\{x_1, \ldots, x_l\}$ denote the real roots of $r^*(x)$ such that $x_i < x_j$ for $1 \le i < j \le l$. We call plane key points for a fiber $x = x_i$, all the solutions of the above triangular systems in this fiber, we denote them by $(x_i, y^m_{i,j}), 1 \le j \le l^m_i$, $0 \le m \le d^2$, in addition we simplify the notation $y^0_{i,j}$ as $y_{i,j}$. Let $\{P_{i,j,k}(x_i, y_{i,j}, z_{i,j,k}), 1 \le j \le l^0_i, 1 \le k \le l_{i,j}\}$ be the space key points of $\mathcal{C}$. By definitions of h_m by Equation (2) and ϕ_{s_m} by Equation (1), one has $\mathcal{C}_{h_m} = \pi \circ \phi_{s_m}(\mathcal{C})$, so the values $y^m_{i,j} = y^0_{i,j} - s_m z_{i,j,k}$, thus Corollary 3.6 enables to recover the values $z_{i,j,k}$ as detailed in Algorithm 2.

In practical computation, all points in Algorithm 2 are represented by intervals, in order to avoid spurious additional z points due to over-estimation in interval operations, we need to analyze the width of these intervals. Let i, j be fixed. A sufficient condition for Algorithm 2 to be correct when working with intervals is to compute intervals for the values in each $T_{s_m,j}$ such that (c1): two intervals do not intersect if and only if they isolate different values in $\cup_{1 \le m \le d^2} T_{s_m,j}$. Let δ be a lower bound on the minimum distance between distinct elements of $\cup_{1 \le m \le d^2} T_{s_m,j}$. If the widths of the isolating intervals of the values in the $T_{s_m,j}$ are smaller than $\delta/2$ then condition (c1) is satisfied. Such a lower bound δ can be computed as follows. Let $\mathrm{Sep}(h_m)$ be the separation bound of the solutions of $h_m(x_i, y)$. For a set of real numbers E, define the separation of E, denoted $\mathrm{Sep}(E)$, as the minimum distance between distinct elements of E. The elements of $T_{s_m,j}$ are the solutions of $h_m(x_i, y)$ shifted by the value $y^0_{i,j}$ and divided by s_m, thus $\mathrm{Sep}(T_{s_m,j}) = \mathrm{Sep}(h_m)/s_m \ge \mathrm{Sep}(h_m)/d^2$. The separation of the union $\cup_{1 \le m \le d^2} T_{s_m,j}$ is thus larger than $1/d^2$ times

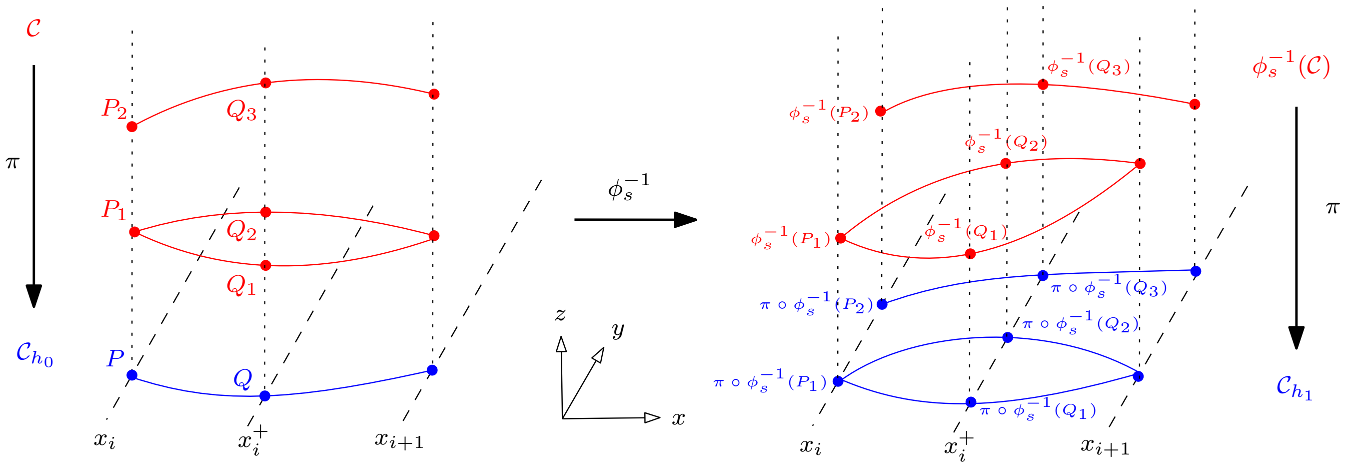

Figure 3: Connection between the x-critical fiber $x = x_i$ and the intermediate fiber $x = x_{i+1}$.

the minimum distance between two distinct values, one being a root of h_{m_1} and the other a root of h_{m_2} (m_1 may be equal to m_2). On the other hand, the minimum distance between two distinct values, one being a root of h_{m_1} and the other a root of h_{m_2} is at least $\mathrm{Sep}(h_{m_1}h_{m_2})$. One can thus define $\delta = 1/d^2 \min_{1\le m_1 \le m_2 \le d^2} \mathrm{Sep}(h_{m_1}h_{m_2})$ as a lower bound on $\mathrm{Sep}(\cup_{1\le m\le d^2} T_{s_m,j})$.

It remains to express the condition that the isolating intervals of the values in the $T_{s_m,j}$ are smaller than $\delta/2$ in terms of widths of the intervals of the input of the algorithm which are isolating the solutions $y^m_{i,j'}$ of $h_m(x_i, y)$ for $0 \le m \le d^2$. Assume that all the solutions $y^m_{i,j'}$ are given by the intervals $[y^m_{i,j'}]$ of widths smaller than $\delta/4$. By construction $T_{s_m,j} = \{(y^0_{i,j} - y^m_{i,j'})/s_m, 1 \le j' \le l^m_i\}$, so that the computed intervals are of widths $w(([y^0_{i,j}] - [y^m_{i,j'}])/s_m]) \le w([y^0_{i,j}]) + w([y^m_{i,j'}]) \le \delta/2$.

According to the analysis above, we thus obtain the following result for the interval version of Algorithm 2.

Lemma 3.8. *Let $\delta = 1/d^2 \min_{1\le m_1\le m_2\le d^2} \mathrm{Sep}(h_{m_1}h_{m_2})$, note that δ has the same asymptotic bound as $\mathrm{Sep}(h_m)$ and [15, Proposition 25] yields $\mathrm{Sep}(h_m) = 2^{-\tilde{O}(d^8+d^7\tau)}$, also a non-asymptotic bound may be computed following the proof of [15, Proposition 25]. Assume the input isolating intervals of the values $y^m_{i,j}$ have widths smaller than $\delta/4$, then Algorithm 2 is correct, that is it returns exactly disjoint intervals for the different values $z_{i,j,k}$.*

3.4. Connections between space points of $\mathcal{C}$

In this subsection, we compute the connections between the space key points of $\mathcal{C}$ on the fibers $x = x_i$ and $x = x_{i+1}$ for $1 \le i \le l-1$, where x_i are the roots of r^* defined in (3). By the definition of r^* and Corollary 3.4, between these fibers, there are no x-critical points of $\mathcal{C}$ nor $\mathcal{C}_{h_0}$ nor $\mathcal{C}_{h_1}$. The connections between two x-fibers can thus be done by straight line segments. These connections can be recovered from the ones in the topology of $\mathcal{C}_{h_0}$ except when several space points project to the same plane point. To separate branches of $\mathcal{C}$ that project to the same branch of $\mathcal{C}_{h_0}$, we add intermediate fibers defined by the $x_i^+ \in \mathbb{Q}$:

$$x_0^+ < x_1 < x_1^+ < \cdots < x_i < x_i^+ < x_{i+1} < \cdots < x_{l-1}^+ < x_l < x_l^+ \tag{7}$$

In practice, x_i^+ is chosen as the mid-point between the isolating bounds of the intervals isolating the x_i. So connecting the points between the fibers $x = x_i$ and $x = x_{i+1}$ is decomposed into connecting the points between the fibers $x = x_i$ and $x = x_i^+$, and connecting the points between the fibers $x = x_i^+$ and $x = x_{i+1}$. By solving the bivariate systems $\{f(x_i^+, y, z), g(x_i^+, y, z)\}$, we obtain the space points of $\mathcal{C}$ on the intermediate fibers. Together with the space key points already computed, one has all space points in all fibers.

In order to recover the topology of $\mathcal{C}$ we use the topology of the two plane curves C_{h_0} and C_{h_1}. We first refine the topologies of $\mathcal{C}_{h_0}$ and $\mathcal{C}_{h_1}$ by adding all the fibers x_i and x_i^+. Figure 3 illustrates the connections between the points of $\mathcal{C}$ on the fibers $x = x_i$ and $x = x_i^+$ (the connection between the points on the fibers $x = x_i^+$ and $x = x_{i+1}$ is similar). For a plane curve segment $\widetilde{PQ}$ of $\mathcal{C}_{h_0}$ between two fibers, let P be its endpoint on $x = x_i$ and Q be its endpoint on $x = x_i^+$. We define the point sets $\pi^{-1}(P) \cap \mathcal{C} = \{P_1, \ldots, P_u\}$ and $\pi^{-1}(Q) \cap \mathcal{C} = \{Q_1, \ldots, Q_v\}$. Since the space curve $\mathcal{C}$ has no x-critical points in the fiber $x = x_i^+$, the number of space curve segments of $\mathcal{C}$ over $\widetilde{PQ}$ is exactly the number of points in $\pi^{-1}(Q) \cap \mathcal{C}$ and all these segments end at different Q_j. Our aim is to determine

Algorithm 3: Space curve topology

Input: $f, g \in \mathbb{Z}[x, y, z]$ coprime defining a curve $\mathcal{C}$.
Output: The topology of $\mathcal{C}$.

0. $(f, g) = \mathbf{SGP}(f, g)$ (Algorithm 1).
1. Compute $h_m(x, y) = \text{Squarefree}(\text{Res}_z(f(x, y + s_m z, z), g(x, y + s_m z, z)))$, for $0 \leq s_m := m \leq d^2$. Compute $r_t(x) = \text{Res}_y(h_t, \partial_y h_t)$, $t \in \{0, 1\}$ and the topology of $\mathcal{C}_{h_0}$ and $\mathcal{C}_{h_1}$.
2. Compute $r^* = \text{lcm}(r_0(x), r_1(x))$.
   Add the points on the fibers corresponding to the real zeros of $\frac{r^*}{r_0}$ into the topology of $\mathcal{C}_{h_0}$.
   Add the points on the fibers corresponding to the real zeros of $\frac{r^*}{r_1}$ into the topology of $\mathcal{C}_{h_1}$.
3. Solve the triangular polynomial systems $\{r^*(x), h_m(x, y)\}$, for $1 \leq m \leq d^2$.
4. Compute the space key points of $\mathcal{C}$ on the fiber $x = x_i \in \mathbb{V}_{\mathbb{R}}(r^*)$ with Algorithm 2, for $1 \leq i \leq l$.
5. Compute x_i^+, $0 \leq i \leq l$, satisfying Equation (7).
6. Add the points on the fibers of $x = x_i^+$, $0 \leq i \leq l$, into the topology of $\mathcal{C}_{h_0}$ and $\mathcal{C}_{h_1}$.
7. Solve the bivariate systems $\{f(x_i^+, y, z), g(x_i^+, y, z)\}$, for $0 \leq i \leq l$.
8. Connect the points between the fibers (Section 3.4).

which Q_j's connects to each P_i. Since P_i may be an x-critical point of $\mathcal{C}$ it may connect 0, 1 or more Q_j's. This information is recovered using the topology of $\mathcal{C}_{h_0}$ and $\mathcal{C}_{h_1}$ as follows.

Lemma 3.9. *Let notations be as above. The space points P_i and Q_j are connected in $\mathcal{C}$ if and only if the plane points $P = \pi(P_i)$ and $Q = \pi(Q_j)$ are connected in $\mathcal{C}_{h_0}$ and $\pi(\phi_{s_1}^{-1}(P_i))$ and $\pi(\phi_{s_1}^{-1}(Q_j))$ are connected in $\mathcal{C}_{h_1}$.*

Proof. "$\Rightarrow$" If P_i connects Q_j on $\mathcal{C}$, the connections are preserved by shearing and projection on $\mathcal{C}_{h_0}$ and $\mathcal{C}_{h_1}$, thus P connects Q and $\pi(\phi_{s_1}^{-1}(P_i))$ connects $\pi(\phi_{s_1}^{-1}(Q_j))$.

"$\Leftarrow$" Assume that P connects Q and $\pi(\phi_{s_1}^{-1}(P_i))$ connects $\pi(\phi_{s_1}^{-1}(Q_j))$. The first condition implies that there exists at least one space curve segment of $\mathcal{C}$ whose projection in $\mathcal{C}_{h_0}$ is $\widetilde{PQ}$. Since x_i^+ is not root of r^*, there is no x-critical point of $\mathcal{C}$ in this fiber, thus the points Q_j are not x-critical points of $\mathcal{C}$. Thus each Q_j connects one and only one point in $\pi^{-1}(P)$. If P_i does not connect Q_j, then there must exist one space point $P_{i'} \in \mathcal{C}$ on the fiber $x = x_i$ which connects Q_j such that $\pi(P_{i'}) = P$ and $P_{i'} \neq P_i$. Since $P_{i'}$ connects Q_j, $\pi(\phi_{s_1}^{-1}(P_{i'}))$ connects $\pi(\phi_{s_1}^{-1}(Q_j))$ in $\mathcal{C}_{h_1}$. In addition, since $P_{i'}$ and P_i are distinct points in $\pi^{-1}(P)$ and $s_1 = 1 \neq 0$, their projections on $\mathcal{C}_{h_1}$ are distinct $\pi(\phi_{s_1}^{-1}(P_{i'})) \neq \pi(\phi_{s_1}^{-1}(P_i))$. By assumption, $\pi(\phi_{s_1}^{-1}(P_i))$ also connects $\pi(\phi_{s_1}^{-1}(Q_j))$. We conclude that $\pi(\phi_{s_1}^{-1}(Q_j))$ is an x-critical point of $\mathcal{C}_{h_1}$, which is a contradiction since the x_i^+ fiber does not contain x-critical point of $\mathcal{C}_{h_1}$. We must then have that P_i connects Q_j which concludes the proof. □

3.5. Main algorithm

We summarize in Algorithm 3 the different steps described in this section for the computation of the topology of the space curve $\mathcal{C}$.

4. Complexity

In this section, we analyze the bit complexity of our algorithms using the notation $\tilde{\mathcal{O}}(\cdot)$ to indicate that we omit poly-logarithmic factors. We first recall complexity results for computations with polynomials. In particular, our analysis uses the recent results of [15] for solving bivariate triangular systems and computing the topology of an algebraic plane curve without shearings.

4.1. Basic results for complexity

The bitsize of an integer n is the number of bits needed to represent it, that is $\lfloor \log n \rfloor + 1$ (log refers to the logarithm in base 2). For a rational q, its bitsize, $\mathcal{L}(q)$, is the maximum bitsize of its numerator and denominator. A multivariate polynomial is called of magnitude (d, τ) if its total degree is bounded by d, and the bitsize of its coefficients is bounded by τ.

Lemma 4.1 ([33])**.** *Let F and G be univariate polynomials of magnitudes (d, τ).*

- *Computing their gcd or the square-free of F have bit complexity $\tilde{\mathcal{O}}(d^2\tau)$, and the gcd or the square-free of F are of magnitude $(d, \mathcal{O}(d+\tau))$.*
- *Given a polynomial $H \in \mathbb{Z}[x]$ that divides F, computing $\frac{F}{H}$ has bit complexity $\tilde{\mathcal{O}}(d(d+\tau))$.*
- *Computing their lcm has bit complexity $\tilde{\mathcal{O}}(d^2\tau)$, and the lcm is of magnitude $(\mathcal{O}(d),\, \mathcal{O}(d+\tau))$.*

Lemma 4.2 ([4, 30, 34])**.** *Let F be a polynomial in $\mathbb{Z}[x]$ of magnitude (d,τ), the separation bound of F, that is the mininum distance between two complex roots, satisfies*

$$\mathrm{Sep}(F) \geq d^{-\frac{d+2}{2}}(d+1)^{\frac{1-d}{2}}2^{\tau(1-d)} = 2^{-\tilde{\mathcal{O}}(d\tau)}.$$

Lemma 4.3 ([29, Theorem 5])**.** *For a polynomial F of magnitude (d,τ), one can compute isolating intervals for all its real roots of width less than 2^{-L} using $\tilde{\mathcal{O}}(d^3 + d^2\tau + dL)$ bit operations. Isolating intervals can be computed such that the sum of the bitsizes is $\tilde{\mathcal{O}}(d\tau)$.*

Lemma 4.4 ([33, Chap. 8])**.** *For two m-variate polynomials of magnitudes (d,τ), the magnitude of their product polynomial is $(\mathcal{O}(d),\, \mathcal{O}(\tau + \log d))$ and it is computed in $\mathcal{O}(d^m\tau)$ bit operations.*

A recursive Horner scheme yields the following results for multivariate polynomial evaluation.

Lemma 4.5 ([7, Lemma 6])**.** *Let $F \in \mathbb{Z}[x_1,\ldots,x_m]$ be of magnitude (d,τ), and $q_1,\ldots,q_m$ be rational numbers each of bitsize σ, then evaluating $F(q_1,\ldots,q_m)$ has a bit complexity of $\tilde{\mathcal{O}}(d^m(\sigma+\tau))$, and the bitsize of $F(q_1,\ldots,q_m)$ is $\mathcal{O}(d\sigma+\tau)$.*

Corollary 4.6 ([6, Lemma 5])**.** *Let $F(x,y) \in \mathbb{Z}[x,y]$ be of magnitude (d,τ), the bit complexity of computing the square-free of F is $\tilde{\mathcal{O}}(d^5 + d^4\tau)$.*

Lemma 4.7 ([17])**.** *Let F and G be polynomials in $\mathbb{Z}[y_1,\ldots,y_k][x]$ with $\deg_x(F) = p \geq q = \deg_x(G)$, $\deg_{y_i}(F) \leq p$ and $\deg_{y_i}(G) \leq q$, F has bitsize τ larger than σ, the bitsize of G. The resultant $\mathrm{Res}_x(F,G)$ can be computed in bit complexity $\tilde{\mathcal{O}}(q(p+q)^{k+1}p^k\tau)$ and it has magnitude $(2pq, \tilde{\mathcal{O}}(p\sigma + q\tau))$.*

Theorem 4.8 ([26],[6] Theorem 43)**.** *Let $F(x,y)$, $G(x,y)$ in $\mathbb{Z}[x,y]$ be of magnitude (d,τ), solving the bivariate system $\{F(x,y) = G(x,y) = 0\}$ has bit complexity $\tilde{\mathcal{O}}(d^5(d+\tau))$.*

We use the following result for the complexity of computing the topology of a plane curve in its original coordinate system, that is identifying the x-fibers of its x-critical points without shearing.

Lemma 4.9 ([15])**.** *Let $F \in \mathbb{Z}[x,y]$ be a square-free polynomial of magnitude (d,τ). Computing the topology of the plane curve $\mathcal{C}(F) = \{(x,y) \in \mathbb{R}^2 \mid F(x,y) = 0\}$ without shearing has bit complexity $\tilde{\mathcal{O}}(d^6 + d^5\tau)$.*

The following result is used in the analysis of the lifting part of our algorithm.

Lemma 4.10 (Teissier [32])**.** *Let $F \in \mathbb{Z}[x,y]$, for an x-critical point $p = (\alpha,\beta)$ of the plane curve $\mathcal{C}_F$, it holds that*

$$\mathrm{mult}(\beta, F(\alpha,y)) = \mathrm{mult}(p, \{F, \partial_y F\}) - \mathrm{mult}(p, \{\partial_x F, \partial_y F\}) + 1,$$

where $\mathrm{mult}(\beta, F(\alpha,y))$ denotes the multiplicity of β as a root of $F(\alpha,y) \in \mathbb{R}[y]$, ∂_x and ∂_y are the partial derivatives and $\mathrm{mult}(p,\{G,H\})$ is the multiplicity of p in the ideal generated by the bivariate polynomials G and H.

4.2. Analysis of Algorithm 3

The proof of our main result, Theorem 4.11, is decomposed in several lemmas.

Theorem 4.11. *Let f and g be coprime polynomials in $\mathbb{Z}[x,y,z]$ of magnitude (d,τ), Algorithm 3 computes the topology of the algebraic space curve $\{(x,y,z) \in \mathbb{R}^3 \mid f(x,y,z) = g(x,y,z) = 0\}$ in $\tilde{\mathcal{O}}(d^{18} + d^{17}\tau)$ bit operations.*

Lemma 4.12. *The bit complexity of Step 0 of Algorithm 3, that is Algorithm 1, is $\tilde{\mathcal{O}}(d^7(d+\tau))$ and the output polynomials have bitsizes $\tilde{\mathcal{O}}(d+\tau)$.*

Proof. In Line 1 of Algorithm 1, a bivariate polynomial of magnitude (d, τ) is evaluated at $(d+1)^2$ points with a complexity of $\mathcal{O}(d^4(d+\tau))$ using Lemma 4.5. For Line 2, first consider the computation of $f(x+\alpha z, y, z) = \sum_{i=0,\ldots,d} c_i(y,z)(x+\alpha z)^i$. The computation of $(x+\alpha z)^i$ can be done with d bivariate multiplications of polynomials of magnitudes $(d, \log d)$, the output bitsize is $\tilde{\mathcal{O}}(d)$ and it is computed in $\tilde{\mathcal{O}}(d^4)$ according to Lemma 4.4. Then the d products of trivariate polynomials $c_i(y,z)(x+\alpha z)^i$ is computed in $\tilde{\mathcal{O}}(d^4(d+\tau))$. The second shearing has the same complexity and finally, the bitsize of $\hat{f}$ and $\hat{g}$ is $\tilde{\mathcal{O}}(d+\tau)$. In Line 3, computing the resultant of $\hat{f}$ and $\hat{g}$ with respect to z has complexity $\tilde{\mathcal{O}}(d(d+d)^3 d^2(d+\tau)) = \tilde{\mathcal{O}}(d^6(d+\tau))$ according to Lemma 4.7. The magnitude of the resultant $h(x,y)$ is $(\tilde{\mathcal{O}}(d^2), \tilde{\mathcal{O}}(d(d+\tau)))$. In Line 4, the computation of $\mathcal{O}(d^2)$ gcds of univariate polynomials of magnitudes $(\mathcal{O}(d^2), \tilde{\mathcal{O}}(d(d+\tau)))$ costs $\tilde{\mathcal{O}}(d^2(d^2)^2 d(d+\tau)) = \tilde{\mathcal{O}}(d^7(d+\tau))$. In Line 7, the $\tilde{\mathcal{O}}(d^2)$ evaluations of the univariate polynomial of magnitude $(\tilde{\mathcal{O}}(d^2), \mathcal{O}(d(d+\tau)))$ costs $\mathcal{O}(d^2 d^2 d(d+\tau)) = \mathcal{O}(d^5\tau)$. Finally, the shearing of Line 8 has the same complexity as the one in Line 2, that is $\tilde{\mathcal{O}}(d^4(d+\tau))$. ☐

Lemma 4.13. *The bit complexity of Steps 1 to 3 of Algorithm 3 is* $\tilde{\mathcal{O}}(d^{13}(d+\tau))$.

Proof. In Step 1, for $0 \le m \le d^2$, the bitsize of $s_m = m$ is $\mathcal{O}(\log d)$, thus the magnitudes of the sheared polynomials $f(x, y+s_m z, z)$ and $g(x, y+s_m z, z)$ are $(d, \mathcal{O}(d\log d+\tau)) = (d, \tilde{\mathcal{O}}(d+\tau))$ and they can be computed in $\tilde{\mathcal{O}}(d^4(d+\tau))$ as in Line 2 of Algorithm 1. By Lemma 4.7, the complexity of computing their resultant is:

$$\tilde{\mathcal{O}}(d(d+d)^3 \cdot d^2(d+\tau)) = \tilde{\mathcal{O}}(d^7 + d^6\tau),$$

and this resultant is of magnitude $(\mathcal{O}(d^2), \tilde{\mathcal{O}}(d^2+d\tau))$. According to Corollary 4.6, the square-free computation for h_m has bit complexity

$$\tilde{\mathcal{O}}((d^2)^5 + (d^2)^4 \cdot (d^2+d\tau)) = \tilde{\mathcal{O}}(d^{10} + d^9\tau).$$

Using again Lemma 4.7, the complexity of computing the resultant r_t is

$$\tilde{\mathcal{O}}(d^2(d^2+d^2)^2 d^2(d^2+d\tau)) = \tilde{\mathcal{O}}(d^{10} + d^9\tau).$$

Since m ranges from 0 to d^2 and t ranges from 0 to 1, computing all $h_m(x,y)$ and $r_t(x)$ has bit complexity

$$(d^2+1)(\tilde{\mathcal{O}}(d^7+d^6\tau) + \tilde{\mathcal{O}}(d^{10}+d^9\tau)) + 2\,\tilde{\mathcal{O}}(d^{10}+d^9\tau) = \tilde{\mathcal{O}}(d^{12}+d^{11}\tau).$$

The last operation of Step 1 is the computation of the topology of $\mathcal{C}_{h_0}$ (and $\mathcal{C}_{h_1}$), according to Lemma 4.9, the bit complexity is $\tilde{\mathcal{O}}((d^2)^6 + (d^2)^5(d^2+d\tau)) = \tilde{\mathcal{O}}(d^{12}+d^{11}\tau)$. Hence, the total complexity of Step 1 is $\tilde{\mathcal{O}}(d^{12}+d^{11}\tau)$.

In Step 2, the polynomials $r_0(x)$ and $r_1(x)$ are of magnitudes $(d^4, \tilde{\mathcal{O}}(d^4+d^3\tau))$. By Lemma 4.1, computing the lcm r^* of r_0 and r_1 has a bit complexity of $\tilde{\mathcal{O}}((d^4)^2 \cdot (d^4+d^3\tau)) = \tilde{\mathcal{O}}(d^{11}(d+\tau))$, and the division of r^* by r_0 costs $\tilde{\mathcal{O}}((d^4)\cdot(d^4+d^3\tau)) = \tilde{\mathcal{O}}(d^7(d+\tau)$. Adding the fibers of $\frac{r_0^*}{r_0}$ in the topology of $\mathcal{C}_{h_0}$ means solving the triangular system $\{\frac{r_0^*}{r_0}, h_0\}$. The polynomial $\frac{r_0^*}{r_0}$ is of magnitude $(\mathcal{O}(d^4), \tilde{\mathcal{O}}(d^4+d^3\tau))$ and h_0 is of magnitude $(d^2, \tilde{\mathcal{O}}(d^2+d\tau))$. According to [15, Proposition 27(a.1)], this triangular system solving is in $\tilde{\mathcal{O}}(d^{12}+d^{11}\tau)$. Adding the points on the fibers corresponding to the real zeros of $\frac{r^*}{r_1}$ into the topology of $\mathcal{C}_{h_1}$ has the same complexity.

Finally, in Step 3, each of the d^2 triangular systems solving $\{r^*(x), h_m(x,y)\}$ costs $\tilde{\mathcal{O}}(d^{12}+d^{11}\tau)$, thus they can all be solved in $\tilde{\mathcal{O}}(d^{14}+d^{13}\tau)$. ☐

Lemma 4.14. *The bit complexity of Step 4 of Algorithm 3, that is Algorithm 2, is* $\tilde{\mathcal{O}}(d^{17}(d+\tau))$.

Proof. First note that with fast arithmetic operations, addition, multiplication or division of rational numbers has a bit complexity that is softly linear in the bitsize of the input. The bit complexity of comparing two rationals is upper bounded by twice the smallest bitsize of the two rationals. The complexity of computing the intersection of two ordered sets is linear in the number of elements, the bit complexity is thus linear in the sum of the bitsizes of all elements in the sets. To apply Algorithm 2, one has to construct the $\mathcal{O}(d^4 \times d^2 \times d^2)$ sets $T^i_{s_m,j}$, each with $\mathcal{O}(d^2)$ elements. All operations performed by the algorithm, in particular the intersection computations, are softly linear in the sum of the bitsizes of all the elements of all the sets $T^i_{s_m,j}$. It is thus enough to compute this sum.

According to Lemma 3.8, for Algorithm 2 to be correct with intervals, it is sufficient to refine the input y coordinates up to width $1/4d^2 \min_{1\le m_1\le m_2\le d^2} \mathrm{Sep}(h_{m_1}h_{m_2}) = 2^{-\tilde{\mathcal{O}}(d^8+d^7\tau)}$. There are d^{10} elements in all the T sets, thus the total bit size is $\tilde{\mathcal{O}}(d^{18}+d^{17}\tau)$.

We now analyze the cost of refining all the y-intervals of one triangular system $\{r^*(x), h_m(x,y)\}$ to a bitsize $L = \tilde{\mathcal{O}}(d^8+d^7\tau)$. According to [15, Proposition 27(b)], the bit complexity of this refinement is bounded by $\tilde{\mathcal{O}}(L(N\mu+$

$n^2 \sum \mu_x)$), with $N = \mathcal{O}(d^4)$ is the degree of r^*, $n = \mathcal{O}(d^2)$ is the degree of h_m, for x in the solution set $V(r^*)$ of r^*, $\mu_x = \max_{y|(x,y)\in V=V(r^*,h_m)} \text{mult}(y, h_m(x,.))$ and $\mu = \max_{x\in V(r^*)} \mu_x \leq n = \mathcal{O}(d^2)$. Teissier's Lemma 4.10 yields $\text{mult}(y, h_m(x,.)) \leq \text{mult}((x,y), \{h_m, \partial_y h_m\}) + 1$. So that $\mu_x \leq \max_{y|(x,y)\in V=V(r^*,h_m)}(\text{mult}((x,y), \{h_m, \partial_y h_m\}) + 1)$ and

$$\begin{aligned} \sum_{x\in V(r^*)} \mu_x &\leq \#V(r^*/r_m) + \sum_{x\in V(r^*)\cap V(r_m)} \max_{y|(x,y)\in V} (\text{mult}((x,y), \{h_m, \partial_y h_m\}) + 1) \\ &\leq N + \sum_{x\in V(r^*)\cap V(r_m)} \max_{y|(x,y)\in V} \text{mult}((x,y), \{h_m, \partial_y h_m\}) \end{aligned}$$

with $r_m(x) = \text{Res}_y(h_m, \partial_y h_m)$. One the other hand, Bézout bound for the system $\{h_m, \partial_y h_m\}$ yields

$$\sum_{x\in V(r_m)} \sum_{y|(x,y)\in V(h_m,\partial_y h_m)} \text{mult}((x,y), \{h_m, \partial_y h_m\}) \leq n^2 = \mathcal{O}(d^4)$$

Thus $\sum_{x\in V(r^*)\cap V(r_m)} \max_{y|(x,y)\in V} \text{mult}((x,y), \{h_m, \partial_y h_m\})$ is also bounded by $\mathcal{O}(d^4)$ and $\sum_{x\in V(r^*)} \mu_x = \mathcal{O}(d^4)$.

So, the bit complexity of refining all the y-intervals of one triangular system $\{r^*(x), h_m(x,y)\}$ is bounded by $\tilde{\mathcal{O}}(L(N\mu + n^2 \sum \mu_x)) = \tilde{\mathcal{O}}((d^8 + d^7\tau)(d^4 d^2 + (d^2)^2 d^4)) = \tilde{\mathcal{O}}(d^{15}(d+\tau))$. Since we have $d^2 + 1$ curves h_m, the bit complexity of Step 4 is $\tilde{\mathcal{O}}(d^{17}(d+\tau))$. ☐

Lemma 4.15. *The bit complexity of Steps 5 to 8 of Algorithm 3 is* $\tilde{\mathcal{O}}(d^{13}(d+\tau))$.

Proof. In Step 5, the roots of $r^*(x)$ have already been isolated in Step 3. The rationals x_i^+ computed as the mid-points between the isolating intervals of the x_i have a total bitsize equivalent to the total bitsize of the isolating intervals of the x_i, thus

$$\sum_{i=0}^{l} \mathcal{L}(x_i^+) = \tilde{\mathcal{O}}(d^8 + d^7\tau)$$

according to Lemma 4.3.

In Step 6, $h_0(x,y)$ and $h_1(x,y)$ are of magnitudes $(\mathcal{O}(d^2), \tilde{\mathcal{O}}(d^2 + d\tau))$ and one has to isolate the roots of $h_0(x_i^+, y)$ and $h_1(x_i^+, y)$. The univariate polynomials $h_0(x_i^+, y)$ and $h_1(x_i^+, y)$ are of magnitude $(\mathcal{O}(d^2), \tilde{\mathcal{O}}(d^2\mathcal{L}(x_i^+) + d\tau)$ and they are computed in $\tilde{\mathcal{O}}(d^4(\mathcal{L}(x_i^+) + d^2 + d\tau))$ according to Lemma 4.5. According to Lemma 4.3, the bit complexity of the real root isolations for all $i = 0, \ldots, l = \tilde{\mathcal{O}}(d^4)$, is

$$\tilde{\mathcal{O}}(\sum_{i=0}^{l} d^6 + d^4(d^2\mathcal{L}(x_i^+) + d\tau)) = \tilde{\mathcal{O}}(d^{10} + d^9\tau) + \mathcal{O}(d^6) \sum_{i=0}^{l} \mathcal{L}(x_i^+) = \mathcal{O}(d^6) \cdot \tilde{\mathcal{O}}(d^8 + d^7\tau) = \tilde{\mathcal{O}}(d^{13}(d+\tau)).$$

The complexity of adding the fibers $x = x_i^+$ into the topology of $\mathcal{C}_{h_0}$ and $\mathcal{C}_{h_1}$ is thus $\tilde{\mathcal{O}}(d^{13}(d+\tau))$.

In Step 7, writing $f(x,y,z) = \sum_{j,k} c_{j,k}(x) y^j z^k$, the evaluation at x_i^+ is done via $\mathcal{O}(d^2)$ evaluations of univariate polynomials of degree at most d. According to Lemma 4.5, this costs $\mathcal{O}(d^3\mathcal{L}(x_i^+) + \tau)$ and $f(x_i^+, y, z)$ and $g(x_i^+, y, z)$ are of magnitudes $(d, d\mathcal{L}(x_i^+) + \tau)$. By Theorem 4.8, solving one of these systems has complexity $\tilde{\mathcal{O}}(d^5(d\mathcal{L}(x_i^+) + \tau))$, so that the total complexity of solving all these bivariate systems is:

$$\sum_{i=0}^{l} \tilde{\mathcal{O}}(d^5(d\mathcal{L}(x_i^+) + \tau)) = \tilde{\mathcal{O}}(d^9\tau) + \tilde{\mathcal{O}}(d^6) \sum_{i=0}^{l} \mathcal{L}(x_i^+) = \tilde{\mathcal{O}}(d^{14} + d^{13}\tau).$$

Finally, in Step 8, the complexity of the connection, following Lemma 3.9, is bounded by the size of the graphs encoding the topologies of $\mathcal{C}_{h_0}$ and $\mathcal{C}_{h_1}$, it is thus in $\mathcal{O}(d^6)$. ☐

Acknowledgement

The work is partially supported by NSFC Grants 11471327 and 12001177.

References

[1] Lionel Alberti, Bernard Mourrain, and Julien Wintz. Topology and arrangement computation of semi-algebraic planar curves. *Computer Aided Geometric Design*, 25(8):631–651, 2008.

[2] Juan Gerardo Alcazar and J. Rafael Sendra. Computation of the topology of real algebraic space curves. *Journal of Symbolic Computation*, 39(6):719–744, 2005.

[3] Dennis S Arnon and Scott McCallum. A polynomial-time algorithm for the topological type of a real algebraic curve. *Journal of Symbolic Computation*, 5(1):213–236, 1988.

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

[5] Eric Berberich, Pavel Emeliyanenko, Alexander Kobel, and Michael Sagraloff. Exact symbolic-numeric computation of planar algebraic curves. *Theoretical Computer Science*, 491:1–32, 2013.

[6] Yacine Bouzidi, Sylvain Lazard, Guillaume Moroz, Marc Pouget, Fabrice Rouillier, and Michael Sagraloff. Solving bivariate systems using rational univariate representations. *Journal of Complexity*, 37:34–75, 2016.

[7] Yacine Bouzidi, Sylvain Lazard, Marc Pouget, and Fabrice Rouillier. Separating linear forms and rational univariate representations of bivariate systems. *Journal of Symbolic Computation*, 68:84 – 119, 2015.

[8] Michael Burr, Sung Woo Choi, Benjamin Galehouse, and Chee K Yap. Complete subdivision algorithms, ii: Isotopic meshing of singular algebraic curves. In *Proceedings of the twenty-first international symposium on Symbolic and algebraic computation*, pages 87–94. ACM, 2008.

[9] Changbo Chen and Wenyuan Wu. A continuation method for visualizing planar real algebraic curves withsingularities. *International Workshop on Computer Algebra in Scientific Computing*, 2018.

[10] Jin-San Cheng, Xiao-Shan Gao, and Ming Li. Determining the topology of real algebraic surfaces. In *Mathematics of Surfaces XI*, pages 121–146. Springer, 2005.

[11] Jin-San Cheng and Kai Jin. Finding a deterministic generic position for an algebraic space curve. In Vladimir Gerdt, Wolfram Koepf, Werner Seiler, and Evgenii V. Vorozhtsov, editors, *Computer Algebra in Scientific Computing*, volume 8660 of *Lecture Notes in Computer Science*, pages 74–84. Springer International Publishing, 2014.

[12] Jin-San Cheng, Kai Jin, and Daniel Lazard. Certified rational parametric approximation of real algebraic space curves with local generic position method. *Journal of Symbolic Computation*, 58:18–40, 2013.

[13] Jin-San Cheng, Sylvain Lazard, Luis Peñaranda, Marc Pouget, Fabrice Rouillier, and Elias Tsigaridas. On the topology of real algebraic plane curves. *Mathematics in Computer Science*, 4(1), 2010.

[14] Diatta Niang Daouda, Bernard Mourrain, and Olivier Ruatta. On the computation of the topology of a non-reduced implicit space curve. In *Proceedings of the twenty-first international symposium on Symbolic and algebraic computation*, pages 47–54. ACM, 2008.

[15] Daouda Niang Diatta, Sény Diatta, Fabrice Rouillier, Marie-Françoise Roy, and Michael Sagraloff. Bounds for polynomials on algebraic numbers and application to curve topology, 2021.

[16] Daouda Niang Diatta, Bernard Mourrain, and Olivier Ruatta. On the isotopic meshing of an algebraic implicit surface. *Journal of Symbolic Computation*, 47(8):903–925, 2012.

[17] Dimitrios I Diochnos, Ioannis Z Emiris, and Elias P Tsigaridas. On the asymptotic and practical complexity of solving bivariate systems over the reals. *Journal of Symbolic Computation*, 44(7):818–835, 2009.

[18] Arno Eigenwillig and Michael Kerber. Exact and efficient 2d-arrangements of arbitrary algebraic curves. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 122–131. Society for Industrial and Applied Mathematics, 2008.

[19] Arno Eigenwillig, Michael Kerber, and Nicola Wolpert. Fast and exact geometric analysis of real algebraic plane curves. In *Proceedings of the 2007 international symposium on Symbolic and algebraic computation*, pages 151–158. ACM, 2007.

[20] M'hammed El Kahoui. Topology of real algebraic space curves. *Journal of Symbolic Computation*, 43(4):235–258, 2008.

[21] Gregory Gatellier, Abder Labrouzy, Bernard Mourrain, and Jean-Pierre Técourt. Computing the topology of three-dimensional algebraic curves. In *Computational methods for algebraic spline surfaces*, pages 27–43. Springer, 2005.

[22] Laureano Gonzalez-Vega and Ioana Necula. Efficient topology determination of implicitly defined algebraic plane curves. *Computer aided geometric design*, 19(9):719–743, 2002.

[23] Hoon Hong. An efficient method for analyzing the topology of plane real algebraic curves. *Mathematics and Computers in Simulation*, 42(4):571–582, 1996.

[24] Kai Jin and Jin-San Cheng. Isotopic epsilon-meshing of real algebraic space curves. In *Proceedings of the 2014 Symposium on Symbolic-Numeric Computation*, SNC '14, pages 118–127, New York, NY, USA, 2014. ACM.

[25] Kai Jin and Jin-San Cheng. On the complexity of computing the topology of real algebraic space curves. *Journal of Systems Science and Complexity*, 34:809–826, 2021.

[26] Alexander Kobel and Michael Sagraloff. On the complexity of computing with planar algebraic curves. *Journal of Complexity*, 31(2):206–236, 2015.

[27] Daniel Lazard. Cad and topology of semi-algebraic sets. *Mathematics in Computer Science*, 4(1):93–112, 2010.

[28] Qianqian Li, Yimin Sun, and Xueli Xu. Constructive criterion algorithm of the global controllability for a class of planar polynomial systems. *Control Theory and Applications Iet*, 9(5):811–816, 2014.

[29] Kurt Mehlhorn, Michael Sagraloff, and Pengming Wang. From approximate factorization to root isolation with application to cylindrical algebraic decomposition. *Journal of Symbolic Computation*, 66:34–69, 2015.

[30] Maurice Mignotte. *Mathematics for Computer Algebra*. Springer-Verlag, 1992.

[31] Raimund Seidel and Nicola Wolpert. On the exact computation of the topology of real algebraic curves. In *Proceedings of the twenty-first annual symposium on Computational geometry*, pages 107–115. ACM, 2005.

[32] Bernard Teissier. Cycles évanescents, sections planes et conditions de whitney. *Astérisque*, 7(8):285–362, 1973.

[33] Joachim Von Zur Gathen and Jürgen Gerhard. *Modern computer algebra*. Cambridge university press, 2013.

[34] Chee Keng Yap. *Fundamental Problems in Algorithmic Algebra*. Cambridge university press, 2000.