HAL open science

On the Geometry and the Topology of Parametric Curves

Christina Katsamaki, Fabrice Rouillier, Elias Tsigaridas, Zafeirakis Zafeirakopoulos

▶ **To cite this version:**

Christina Katsamaki, Fabrice Rouillier, Elias Tsigaridas, Zafeirakis Zafeirakopoulos. On the Geometry and the Topology of Parametric Curves. ISSAC 2020 - International Symposium on Symbolic and Algebraic Computation, Jul 2020, Kalamata / Virtual, Greece. hal-02573423v2

HAL Id: hal-02573423

https://hal.science/hal-02573423v2

Submitted on 18 Jun 2020

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.

On the Geometry and the Topology of Parametric Curves

CHRISTINA KATSAMAKI *, christina.katsamaki@inria.fr
FABRICE ROUILLIER *, Fabrice.Rouillier@inria.fr
ELIAS TSIGARIDAS, elias.tsigaridas@inria.fr
* Inria Paris, IMJ-PRG, Sorbonne Université and Paris Université, France
ZAFEIRAKIS ZAFEIRAKOPOULOS, zafeirakopoulos@gtu.edu.tr
Institute of Information Technologies, Gebze Technical University, Turkey

We consider the problem of computing the topology and describing the geometry of a parametric curve in $\mathbb{R}^n$. We present an algorithm, PTOPO, that constructs an abstract graph that is isotopic to the curve in the embedding space. Our method exploits the benefits of the parametric representation and does not resort to implicitization.

Most importantly, we perform all computations in the parameter space and not in the implicit space. When the parametrization involves polynomials of degree at most d and maximum bitsize of coefficients τ, then the worst case bit complexity of PTOPO is $\widetilde{O}_B(nd^6 + nd^5\tau + d^4(n^2 + n\tau) + d^3(n^2\tau + n^3) + n^3d^2\tau)$. This bound matches the current record bound $\widetilde{O}_B(d^6 + d^5\tau)$ for the problem of computing the topology of a planar algebraic curve given in implicit form. For planar and space curves, if $N = \max\{d, \tau\}$, the complexity of PTOPO becomes $\widetilde{O}_B(N^6)$, which improves the state-of-the-art result, due to Alcázar and Díaz-Toca [CAGD'10], by a factor of N^{10}. However, visualizing the curve on top of the abstract graph construction, increases the bound to $\widetilde{O}_B(N^7)$. We have implemented PTOPO in MAPLE for the case of planar curves. Our experiments illustrate its practical nature.

Additional Key Words and Phrases: Parametric curve, topology, bit complexity, polynomial systems

1 INTRODUCTION

Parametric curves constitute a classical and important topic in computational algebra and geometry [38] that constantly receives attention, e.g., [11, 13, 35, 39]. The interest in efficient algorithms for computing with parametric curves has been motivated, among others, by the omnipresence of parametric representations in computer modeling and computer aided geometric design, e.g., [16].

We focus on computing the topology of a real parametric curve, that is, the computation of an abstract graph that is isotopic [7, p. 184] to the curve in the embedding space. We design a complete algorithm, PTOPO, that applies directly to parametric curves of any dimension. We consider different characteristics of the parametrization, like properness and normality, before computing the singularities and other interesting points on the curve. These points are necessary for representing the geometry of the curve, as well as for producing a certified visualization of planar and space curves.

Previous work. A common strategy when dealing with parametric curves is implicitization. There has been great research effort, e.g., [11, 36] and references therein, in designing algorithms to compute the implicit equations describing the curve. However, it is also important to manipulate parametric curves directly, without converting them to implicit form.

The study of the topology of a real parametric curve is a topic that has not received much attention in the literature, in contrast to its implicit counterpart [14, 22]. It requires special treatment, since for instance it is not always easy to choose a parameter interval such that when we plot the curve over it, we include all the important topological features [3]. Moreover, while visualizing the curve using symbolic computational tools, the problem of missing points and branches may arise [32, 37]. Alcázar and Díaz-Toca [3] study the topology of real parametric curves without implicitizing. They work directly with the parametrization and address both planar and space real rational curves. Our algorithm to compute the topology is to be juxtaposed to their work; we refer

to the next paragraph for more details. We also refer to [12] and [2] for other approaches based on computations by values and subdivision, respectively.

To compute the topology of a curve it is essential to detect its singularities. This is an important and well studied problem [3, 22, 33] of independent interest. Apart from classical approaches [17, 42] that work in the implicit representation, we can also compute the singularities using directly the parametrization. For instance, necessary and sufficient conditions to identify cusps and inflection points are expressed in the form of determinants, e.g., [23, 26].

On computing the singularities of a parametric curve, a line of work related to our approach, does so by means of a univariate resultant [1, 19, 29, 31, 33]. Notably in [33] the authors work on rational parametric curves in affine n-space; they use generalized resultants to find the parameters of the singular points. Moreover, they characterize the singularities and compute their multiplicities.

Cox [13] uses the syzygies of the ideal generated by the polynomials that give the parameterization to compute the singularities and their structure. There are state-of-the-art approaches that exploit this idea and relate the problem of computing the singularities with the notion of the μ-basis of the parametrization, e.g., [21] and references therein. Another method is used in [6], where they compute and characterize the singularities using factorization of resultants. In [5] they use the projection from the rational normal curve to the curve and its relation with the secant varieties to the normal curve.

Overview of our contributions. We introduce PTOPO, a complete, exact, and efficient algorithm (Alg. 3) for computing the geometric properties and the topology of parametric curves in $\mathbb{R}^n$. Unlike other algorithms, e.g. [3], it makes no assumptions on the input curves, such as the absence of axis-parallel asymptotes, and it does not perform any projections and liftings when $n \geq 2$. For this, it is applicable to any dimension. Nevertheless, it does not handle knots for space curves.

If the (proper) parametrization of the curve consists of polynomials of degree d and bitsize τ, then PTOPO outputs a graph isotopic [7, p.184] to the curve in the embedding space, by performing

$$\widetilde{O}_B(nd^6 + nd^5\tau + d^4(n^2 + n\tau) + d^3(n^2\tau + n^3) + n^3d^2\tau)$$

bit operations in the worst case (Thm. 5.5). We also provide a Las Vegas variant with expected complexity

$$\widetilde{O}_B(d^6 + d^5(n + \tau) + d^4(n^2 + n\tau) + d^3(n^2\tau + n^3) + n^3d^2\tau).$$

If $n = O(1)$, the bounds become $\widetilde{O}_B(N^6)$, where $N = \max\{d, \tau\}$. The vertices of the output graph correspond to special points on the curve, in whose neighborhood the topology is not trivial, given by their parameter values. Each edge of the graph is associated with two parameter values and corresponds to a unique smooth parametric arc. For an embedding isotopic to the curve, we map every edge of the abstract graph to the corresponding parametric arc.

For planar and space curves, our bound improves the previously known one due to Alcázar and Díaz-Toca [3] by a factor of $\widetilde{O}_B(N^{10})$. The latter algorithm [3] performs computations in the implicit space. On the contrary, PTOPO is a fundamentally different approach since we work exclusively in the parameter space; we do not use a sweep-line algorithm to construct the isotopic graph. We handle only the parameters that give important points on the curve and thus we avoid performing operations such as univariate root isolation in an extension field or evaluation of a polynomial at an algebraic number.

Computing singular points is an essential part of PTOPO (Lem. 4.7). We chose not to exploit recent methods, e.g., [6], for this task, but to employ older techniques, e.g., [3, 31, 33], that rely on a bivariate polynomial system, Eq. (2). We take advantage of this system's symmetry and of nearly optimal algorithms for bivariate system solving and for computations with real algebraic numbers [8, 14, 15, 28]. In particular, we introduce an algorithm for isolating the roots of over-determined bivariate polynomial systems by exploiting the Rational Univariate

Representation (RUR) [8–10] that has worst case and expected bit complexity that matches the ones for square systems (Thm. 4.6). These are definitive steps for obtaining the complexity bounds of Thm. 5.4 and Thm. 5.5.

Moreover, our bound matches the current state-of-the-art complexity bound, $\widetilde{O}_B(d^6 + d^5\tau)$ or $\widetilde{O}_B(N^6)$, for computing the topology of implicit plane curves [14, 22]. However, if we want to visualize the graph in 2D or 3D, then we have to compute a characteristic box (Lem. 5.1) that contains all the the topological features of the curve and the intersections of the curve with its boundary. In this case, the complexity of PTOPO becomes $\widetilde{O}_B(N^7)$ (Thm. 5.4).

A preprocessing step of PTOPO consists in finding a proper reparametrization of the curve (if it is not proper). We present explicit bit complexity bounds (Lem. 3.2) for the algorithm of Pérez [30] to compute a proper parametrization. Another preprocessing step is to ensure that there are no singularities at infinity; Lem. 3.3 handles this task and provides explicit complexity estimates.

Last but not least, we provide a certified implementation[1] of PTOPO in MAPLE. So far, the implementation handles the topology computation and visualization of planar curves.

Organization of the paper. The next section presents our notation and some useful results needed for our proofs. In Sect. 3 we give the basic background on rational curves in affine n-space. We characterize the parametrization by means of injectivity and surjectivity and describe a reparametrization algorithm. In Sect. 4 we present the algorithm to compute the singular, extreme points, and isolated points on the curve. In Sect. 5 we describe our main algorithm, PTOPO, that constructs a graph isotopic to the curve in the embedding space and its complexity. Finally, in Sect. 6 we give examples and experimental results.

2 NOTATION AND ALGEBRAIC TOOLS

For a polynomial $f \in \mathbb{Z}[x]$, its infinity norm is equal to the maximum absolute value of its coefficients. We denote by $\mathcal{L}(f)$ the logarithm of its infinity norm. We also call the latter the bitsize of the polynomial. A univariate polynomial is of size (d, τ) when its degree is at most d and has bitsize τ. The bitsize of a rational function is the maximum of the bitsizes of the numerator and the denominator. We represent an algebraic number $\alpha \in \mathbb{C}$ by the *isolating interval representation*. When $\alpha \in \mathbb{R}$ (resp. $\mathbb{C}$), it includes a square-free polynomial which vanishes at α and a (rational) interval (resp. Cartesian products of intervals) containing α and no other root of this polynomial. We denote by O, resp. O_B, the arithmetic, resp. bit, complexity and we use $\widetilde{O}$, resp. $\widetilde{O}_B$, to ignore (poly-)logarithmic factors. We denote by $\mathtt{res}_x(f, g)$ the resultant of the polynomials f, g with respect to x. For $t \in \mathbb{C}$, we denote by $\bar{t}$ its complex conjugate. We use $[n]$ to signify the set $\{1, \ldots, n\}$.

We now present some useful results, needed for our analysis.

LEMMA 2.1. *Let $A = \sum_{i=0}^{m} a_i X^i, B = \sum_{i=0}^{n} b_i X^i \in \mathbb{Z}[X]$ of degrees m and n and of bitsizes τ and σ respectively. Let $\alpha_1, \ldots, \alpha_m$ be the complex roots of A, counting multiplicities. Then, for any $\kappa = 1, \ldots, m$ it holds that*

$$2^{-m\sigma - n\tau - (m+n)\log(m+n)} < |B(\alpha_\kappa)| < 2^{m\sigma + n\tau + (m+n)\log(m+n)}.$$

PROOF. Following [40], let $R = \mathtt{res}_X(A(X), Y - B(X)) \in \mathbb{Z}[Y]$. Using the Poisson's formula for the resultant we can write $R(Y) = a_m^n \prod_{\kappa=1}^{m}(Y - B(\alpha_\kappa))$. The maximum bitsize of the coefficients of $R(Y)$ is at most $m\sigma + n\tau + (m+n)\log(m+n)$. We observe that the roots of $R(Y)$ are $B(\alpha_\kappa)$ for $\kappa = 1, \ldots, m$. Therefore, using Cauchy's bound we deduce that

$$2^{-m\sigma - n\tau - (m+n)\log(m+n)} < |B(\alpha_\kappa)| < 2^{m\sigma + n\tau + (m+n)\log(m+n)}.$$

□

Lemmata 2.2, 2.3 restate known results on the gcd computation of various univariate and bivariate polynomials.

[1] https://webusers.imj-prg.fr/~christina.katsamaki/ptopo/

Lemma 2.2. *Let $f_1(X), \dots, f_n(X) \in \mathbb{Z}[X]$ of sizes (δ, L). We can compute their* gcd *in worst case complexity $\widetilde{O}_B(n(\delta^3 + \delta^2 L))$, or with a Monte Carlo algorithm in $\widetilde{O}_B(\delta^2 + \delta L)$, or with a Las Vegas algorithm in $\widetilde{O}_B(n(\delta^2 + \delta L))$.*

Proof. These are known results [41]; we repeat the arguments adapted to our notation.

Worst case: We compute g by performing n consecutive gcd computations, that is $\gcd(f_1, \gcd(f_2, \gcd(\cdots, \gcd(f_{n-1}, f_n)))$. Since each gcd computation costs $\widetilde{O}_B(\delta^3 + \delta^2 L)$ [10, Lem.4], the result for this case follows.

Monte Carlo: We perform one gcd computation by allowing randomization. If we choose integers $a_3, \dots, a_n$ independently at random from the set $\{1, \dots, Kd\}$, where $K = O(1)$, we get that $\gcd(f_1, \dots, f_n) = \gcd(f_1, f_2 + a_3 f_3 + \cdots + a_n f_n)$ in $\mathbb{Z}[x]$, with probability $\geq 1/2$ [41, Thm. 6.46]. We compute $g^* = \gcd(f_1, f_2 + a_3 f_3 + \cdots + a_n f_n)$. Notice that the polynomial $f_2 + a_3 f_3 + \cdots + a_n f_n$ is asymptotically of size (δ, L). So, it takes $\widetilde{O}_B(\delta^2 + \delta L)$ to find g^*, using the probabilistic algorithm in [34].

Las Vegas: We can reduce the probability of failure in the Monte Carlo variant of the gcd computation to zero, by performing n exact divisions. In particular, we check if g^* divides $h_3, \dots, h_n$. Using [41, Ex.10.21], the bit complexity of these operations is in total $\widetilde{O}_B(n(\delta^2 + \delta\tau))$. □

Lemma 2.3. *Let $f_1(X, Y), \dots, f_n(X, Y) \in \mathbb{Z}[X, Y]$ of bidegrees (δ, δ) and $\mathcal{L}(f_i) = L$. We can compute their* gcd *in worst case complexity $\widetilde{O}_B(n(\delta^5 + \delta^4 L))$, or with a Monte Carlo algorithm in $\widetilde{O}_B(\delta^3 + \delta^2 L)$, or with a Las Vegas algorithm in $\widetilde{O}_B(n(\delta^3 + \delta^2 L))$.*

Proof. The straightforward approach is to perform n consecutive gcd computations, that is $\gcd(f_1, \gcd(f_2, \gcd(\cdots, \gcd(f_{n-1}, f_n)))$. To accelerate the practical complexity we should sort f_i in increasing order with respect to their degree. Each gcd computation costs $\widetilde{O}_B(\delta^5 + \delta^4 L)$ [8, Lem. 5], so the total worst case cost is $\widetilde{O}_B(n\delta^5 + n\delta^4 L)$.

Alternatively, we consider the operation $\gcd(f_1, \sum_{k=2}^{n} a_k f_k)$, where a_k are random integers, following [41, Thm. 6.46]. The expected cost of this gcd is $\widetilde{O}_B(\delta^3 + \delta^2 L)$. To see this, notice that we can perform a bivariate gcd in expected time $\widetilde{O}(\delta^2)$ [41, Cor. 11.12], over a finite field with enough elements, and the bitsize of the result is $\widetilde{O}(\delta + L)$ [25].

Then, for a Las Vegas algorithm, using exact division, we test if the resulting polynomial divides all f_i, for $2 \leq i \leq n$. This costs $\widetilde{O}_B(n(\delta^3 + \delta^2 L))$, by adapting [41, Ex.10.21] to the bivariate case. □

3 RATIONAL CURVES

Following closely [3], we introduce basic notions for rational curves. Let $\widetilde{C}$ be an algebraic curve over $\mathbb{C}^n$, parametrized by the map

$$\begin{aligned} \phi : \quad & \mathbb{C} \dashrightarrow \widetilde{C} \\ & t \mapsto \big(\phi_1(t), \dots, \phi_n(t)\big) = \Big(\frac{p_1(t)}{q_1(t)}, \dots, \frac{p_n(t)}{q_n(t)}\Big), \end{aligned} \tag{1}$$

where $p_i, q_i \in \mathbb{Z}[t]$ are of size (d, τ) for $i \in [n]$, and $\widetilde{C}$ is the Zariski closure of $\mathrm{Im}(\phi)$. We call $\phi(t)$ a *parametrization* of $\widetilde{C}$.

We study the real trace of $\tilde{C}$, that is $C := \tilde{C} \cap \mathbb{R}^n$. A parametrization ϕ is chatacterized by means of *properness* (Sec. 3.1) and *normality* (Sec. 3.2). To ensure these properties, one can reparametrize the curve, i.e., apply a rational change of parameter to the given parametrization. We refer to [39, Ch. 6] for more details on reparameterization.

Without loss of generality, we assume that no component of the parametrization ϕ is constant; otherwise we could embed $\tilde{C}$ in a lower dimensional space. We consider that ϕ is in *reduced form*, i.e., $\gcd(p_i(t), q_i(t)) = 1$, for

all $i \in [n]$. The point at infinity, $\mathbf{p}_\infty$, is the point on C we obtain for $t \to \pm\infty$ (if it exists). For a parametrization ϕ, we consider the following system of bivariate polynomials:

$$h_i(s,t) = \frac{p_i(s)q_i(t) - q_i(s)p_i(t)}{s-t}, \quad \text{for } i \in [n]. \tag{2}$$

Remark 1. *The h_i's are polynomials since (s,s) is a root of the numerator for every s. Also, $h_i(t,t) = \phi_i'(t)q_i^2(t)$ for $i \in [n]$ [20, Lem. 1.7].*

3.1 Proper parametrization

A parametrization is proper if $\phi(t)$ is injective for almost all points on $\widetilde{C}$. In other words, almost every point on $\widetilde{C}$ is the image of exactly one parameter value (real or complex). For other equivalent definitions of properness we refer to [39, Ch. 4], [33]. The following condition [3, Thm. 1] leads to an algorithm for checking properness: a parametrization is proper if and only if $\deg(\gcd(h_1(s,t),\dots,h_n(s,t))) = 0$. By applying Lem. 2.3 we get the following:

Lemma 3.1. *There is an algorithm that checks if a parametrization ϕ is proper in worst-case bit complexity $\widetilde{O}_B(n(d^5 + d^4\tau))$ and in expected bit complexity $\widetilde{O}_B(n(d^3 + d^2\tau))$.*

Proof. The construction of all h_i costs $O_B(nd^2\tau)$. We need to check if $\deg(\gcd(h_1(s,t),\dots,h_n(s,t))) = 0$ [3, Thm. 1]. For the gcd computation, we employ Lem. 2.3 and the result follows. □

If ϕ is a not a proper parametrization, then there always exists a parametrization $\psi \in \mathbb{Z}(t)^n$ and $R(t) \in \mathbb{Z}(t)$ such that $\psi(R(t)) = \phi(t)$ and ψ is proper [39, Thm. 7.6]. There are various algorithms for obtaining a proper parametrization, e.g., [18, 19, 30, 35, 39]. We consider the algorithm in [30] for its simplicity; its pseudo-code is in Alg. 1.

Lemma 3.2. *Consider a non-proper parametrization of a curve C, consisting of univariate polynomials of size (d,τ). Alg. 1 computes a proper parametrization of C, involving polynomials of degree at most d and bitsize $O(d^2 + d\tau)$, in $\widetilde{O}_B(n(d^5 + d^4\tau))$, in the worst case.*

Proof. The algorithm first computes the bivariate polynomials $H_1,\dots H_n$. They have bi-degree at most (d,d) and bitsize at most $2\tau + 1$. Then, we compute their gcd, that we denote by H, in $\widetilde{O}_B(n(d^5 + d^4\tau))$ (Lem. 2.3). By [25] and [4, Prop. 10.12] we have that $\mathcal{L}(H) = O(d + \tau)$, which is also the case for $C_j(s)$.

If the degree of H is one, then the parametrization is already proper and we have nothing to do. Otherwise, we consider H as a univariate polynomial in s and we find two of its coefficients that are relatively prime, using exact division. The complexity of this operation is $m^2 \cdot \widetilde{O}_B(d^2 + d\tau) = \widetilde{O}_B(d^4 + d^3\tau)$ [41, Ex. 10.21].

Subsequently, we perform n resultant computations to get $L_1,\dots L_n$. From these we obtain the rational functions of the new parametrization. We focus on the computation of L_1. The same arguments hold for all L_i. The bi-degree of $L_1(s,x)$ is (d,d) [4, Prop. 8.49] and $\mathcal{L}(L_1) = O(d^2 + d\tau)$ [4, Prop. 8.50]; the latter dictates the bitsize of the new parametrization.

To compute L_1, we consider F_1 and G as univariate polynomials in t and we apply a fast algorithm for computing the univariate resultant based on subresultants [24]; it performs $\widetilde{O}(d)$ operations. Each operation consists of multiplying bivariate polynomials of bi-degree (d,d) and bitsize $O(d^2 + d\tau)$; so it costs $\widetilde{O}_B(d^4 + d^3\tau)$. We compute the resultant in $\widetilde{O}_B(d^5 + d^4\tau)$. We multiply the latter bound by n to conclude the proof. □

Algorithm 1: `Make_Proper`(ϕ)

Input: A parametrization $\phi \in \mathbb{Z}(t)^n$ as in Eq. (1)
Output: A proper parametrization $\psi = (\psi_1, \ldots, \psi_n) \in \mathbb{Z}(t)^n$

for $i \in [n]$ **do** $H_i(s,t) \leftarrow p_i(s)q_i(t) - p_i(t)q_i(s) \in \mathbb{Z}[s,t]$;
$H \leftarrow \gcd(H_1, \ldots, H_n) = C_m(t)s^m + \cdots + C_0(t) \in (\mathbb{Z}[t])[s]$
if $m = 1$ **then** RETURN $\phi(t)$;
Find $k, l \in [m]$ such that:
$\deg(\gcd(C_k(t), C_l(t))) = 0$ and $\frac{C_k(t)}{C_l(t)} \notin \mathbb{Q}$
$R(t) \leftarrow \frac{C_k(t)}{C_l(t)}$
$r \leftarrow \deg(R) = \max\{\deg(C_k), \deg(C_l)\}$
$G \leftarrow s\,C_l(t) - C_k(t)$
for $i \in [n]$ **do**
  $F_i \leftarrow xq_i(t) - p_i(t)$
  $L_i(s,x) \leftarrow \mathtt{res}_t(F_i(t,x), G(t,s)) = (\tilde{q}_i(s)x - \tilde{p}_i(s))^r$
end
RETURN $\psi(t) = \left(\frac{\tilde{p}_1(t)}{\tilde{q}_1(t)}, \ldots, \frac{\tilde{p}_n(t)}{\tilde{q}_n(t)}\right)$

3.2 Normal parametrization

Normality of the parametrization concerns the surjectivity of the map ϕ. The parametrization $\phi(t)$ is $\mathbb{R}$-normal if for all points $\mathbf{p}$ on C there exists $t_0 \in \mathbb{R}$ such that $\phi(t_0) = \mathbf{p}$. When the parametrization is not $\mathbb{R}$-normal, the points that are not in the image of ϕ for $t \in \mathbb{R}$ are $\mathbf{p}_\infty$ (if it exists) and the isolated points that we obtain for complex values of t [32, Prop. 4.2]. An $\mathbb{R}$-normal reparametrization does not always exist. We refer to [39, Sect. 7.3] for further details.

However, if $\mathbf{p}_\infty$ exists, then we reparametrize the curve to avoid possible singularities at infinity. The point $\mathbf{p}_\infty$ exists if $\deg(p_i) \leq \deg(q_i)$, for all $i \in [n]$.

Lemma 3.3. *If $\mathbf{p}_\infty$ exists, then we can reparametrize the curve with a linear function to ensure that $\mathbf{p}_\infty$ is not a singular point, using a Las Vegas algorithm in expected time $\widetilde{O}_B(n(d^2 + d\tau))$. The new parametrization involves polynomials of size $(d, \widetilde{O}(d + \tau))$.*

Proof. The point at infinity depends on the parametrization. So, for this proof, let us denote the point at infinity of ϕ by $\mathbf{p}_\infty^\phi$.

The reparametrization consists in choosing $t_0 \in \mathbb{R}$ and applying the map $r : t \mapsto \frac{t_0\, t+1}{t-t_0}$ to ϕ, to obtain a new parametrization, $\psi = \phi \circ r$. The point at infinity of the new parametrization is $\mathbf{p}_\infty^\psi = \phi(t_0)$. We need to ensure that $\mathbf{p}_\infty^\psi = \phi(t_0)$ is not singular. There are $O(d^2)$ singular points, so we choose t_0 uniformly at random from the set $\{1, \ldots, Kd^2\}$ where $K = O(1)$. Then, with probability $\geq 1/2$, $\phi(t_0)$ is not singular and $\mathbf{p}_\infty^\psi$ is also not singular. The bound on the possible values of t_0 implies that the bitsize of t_0 is $O(\lg(d))$.

We compute the new parametrization, ψ, in $\widetilde{O}_B(n(d^2 + d\tau))$ using multipoint evaluation and interpolation, by exploiting the fact that the polynomials in ψ have degrees at most d and bitsize $\widetilde{O}(d + \tau)$.

For a Las Vegas algorithm we need to check if $\phi(t_0)$ is a cusp or a multiple point. For the former, we evaluate ϕ' at t_0 (see Rem. 1). This costs $\widetilde{O}_B(nd\tau)$ [9, Lem. 3]. For the latter, we check if $\deg(\gcd(\phi_1(t_0)q_1(t) -$

$p_1(t), \ldots, \phi_1(t_0)q_1(t) - p_1(t))) = 0$ in $\widetilde{O}_B(n(d^2 + d\tau))$ (Lem. 2.2). If $\phi'(t_0)$ is not the zero vector and the degree of the gcd is zero, then $\phi(t_0)$ is not singular. □

Remark 2. *Since the reparametrizing function in the previous lemma is linear, it does not affect properness [39, Thm. 6.3].*

4 SPECIAL POINTS ON THE CURVE

We consider a parametrization ϕ of C as in Eq. (1), such that ϕ is proper and there are no singularities at infinity. We highlight the necessity of these assumptions when needed. We detect the *parameters* that generate the *special points* of C, namely the singular, the isolated, and the extreme points. We identify the values of the parameter for which ϕ is not defined, namely the poles (see Def. 1); in presence of poles, C consists of multiple components.

Definition 1. *The parameters for which $\phi(t)$ is not defined are the poles of ϕ. The sets of poles over the complex and the reals are:*

$$\mathsf{T}_P^{\mathbb{C}} = \{t \in \mathbb{C} : \prod_{i \in [n]} q_i(t) = 0\} \text{ and } \mathsf{T}_P^{\mathbb{R}} = \mathsf{T}_P^{\mathbb{C}} \cap \mathbb{R}.$$

We consider the solution set S of system (2) over $\mathbb{C}^2$:

$$S = \{(t, s) \in \mathbb{C}^2 : h_i(t, s) = 0 \text{ for all } i \in [n]\}.$$

Remark 3. *Notice that when ϕ is in reduced form, if $(s, t) \in S$ and $(s, t) \in (\mathbb{C} \setminus \mathsf{T}_P^{\mathbb{C}}) \times \mathbb{C}$, then also $t \notin \mathsf{T}_P^{\mathbb{C}}$ [33, (in the proof of) Lem. 9].*

Next, we present some well known results [33, 39] that we adapt in our notation.

Singular points. Quoting [26], "Algebraically, singular points are points on the curve, in whose neighborhood the curve cannot be represented as an one-to-one and C^∞ bijective map with an open interval on the real line". Geometrically, singularities correspond to shape features that are known as cusps and self-intersections of smooth branches. *Cusps* are points on the curve where the tangent vector is the zero vector. This is a necessary and sufficient condition when the parametrization is proper [26]. Self-intersections are *multiple* points, i.e., points on C with more than one preimages.

Lemma 4.1. *The set of parameters corresponding to real cusps is*

$$\mathsf{T}_C = \left\{t \in \mathbb{R} \setminus \mathsf{T}_P^{\mathbb{R}} : (t, t) \in S\right\}.$$

The set of parameters corresponding to real multiple points is

$$\mathsf{T}_M = \{t \in \mathbb{R} \setminus \mathsf{T}_P^{\mathbb{R}} : \exists s \neq t, s \in \mathbb{R} \text{ such that } (t, s) \in S\}.$$

Proof. The description of T_C is an immediate consequence of Rem. 1. It states that $h_i(t, t) = \phi_i'(t)q_i^2(t)$, for $i \in [n]$.

Now let $\mathbf{p} = \phi(t)$ be a multiple point on C. Then, there is $s \in \mathbb{R} \setminus \mathsf{T}_P^{\mathbb{R}}$ with $\phi(t) = \phi(s) \Rightarrow h_i(t, s) = 0$ for all $i \in [n]$ and so $t \in \mathsf{T}_M$. Conversely, let $t \in \mathsf{T}_M$ and $s \neq t$, $s \in \mathbb{R}$ such that $h_i(t, s) = 0$ for all $i \in [n]$. From [33, (in the proof of) Lem. 9], when ϕ is in reduced form, if $(t, s) \in S$ and $(t, s) \in (\mathbb{R} \setminus \mathsf{T}_P^{\mathbb{R}}) \times \mathbb{R}$, then also $s \notin \mathsf{T}_P^{\mathbb{R}}$. So, $h_i(t, s) = 0 \Leftrightarrow \frac{p_i(t)}{q_i(t)} = \frac{p_i(s)}{q_i(s)}$ for all $i \in [n]$, and thus $\mathbf{p} = \phi(t) = \phi(s)$ is real multiple point. □

Notice that T_C and T_M are not necessarily disjoint, for at the same point we may have both cusps and smooth branches that intersect.

Isolated points. An isolated point on a real curve can only occur for complex values of the parameter. The point at infinity is not isolated because it is the limit of a sequence of real points.

Lemma 4.2. *The set of parameters generating isolated points of C is*

$$\mathsf{T}_I = \{t \in \mathbb{C} \setminus (\mathbb{R} \cup \mathsf{T}_P^{\mathbb{C}}) : (t, \bar{t}) \in S \text{ and } \nexists s \in \mathbb{R} \text{ s.t. } (t, s) \in S\}.$$

Proof. Let $\mathbf{p} = \phi(t) \in \mathbb{R}^n$ be an isolated point, where $t \in \mathbb{C} \setminus (\mathbb{R} \cup \mathsf{T}_P^{\mathbb{C}})$. Notice that $\mathbf{p}$ is also a multiple point, since it holds that $\phi_i(t) = \overline{\phi_i(t)} = \phi_i(\bar{t})$ for $i \in [n]$. Thus, $h_i(t, \bar{t}) = 0$ for all $i \in [n]$ and $(t, \bar{t}) \in S$. Moreover, since $\mathbf{p}$ is isolated, there are no real branches through $\mathbf{p}$ and there does not exist $s \in \mathbb{R}$ such that $\phi(t) = \phi(s) \Rightarrow h_i(t, s) = 0$, for all $i \in [n]$. So, $t \in \mathsf{T}_I$.

Conversely, let $(t, \bar{t}) \in S$ with $t \in \mathbb{C} \setminus \mathbb{R} \cup \mathsf{T}_P^{\mathbb{C}}$. Since ϕ is in reduced form, we have that $\bar{t} \notin P^{\mathbb{C}}$[33, (in the proof of) Lem. 9], therefore $h_i(t, \bar{t}) = 0$, for all $i \in [n]$, implies that $\phi(t) = \phi(\bar{t}) = \overline{\phi(t)} \in \mathbb{R}^n$. Since there does not exist $s \in \mathbb{R}$ with $\phi(t) = \phi(s)$, $\mathbf{p}$ is an isolated point on C. □

Extreme points. Consider a vector $\vec{\delta}$ and a point on C whose tangent vector is parallel to $\vec{\delta}$. If the point is not singular, then it is an extreme point of C with respect to $\vec{\delta}$. We compute the extreme points with respect to the direction of each coordinate axis. Rem. 1 leads to the following lemma:

Lemma 4.3. *The set of parameters generating extreme points is*

$$\mathsf{T}_E = \left\{t \in \mathbb{R} \setminus \mathsf{T}_P^{\mathbb{R}} : \prod_{i \in [n]} h_i(t, t) = 0 \text{ and } t \notin \mathsf{T}_C \cup \mathsf{T}_M\right\}.$$

4.1 Computation and Complexity

From Lemmata 4.1, 4.2, and 4.3, it follows that given a proper parametrization ϕ without singular points at infinity, we can easily find the poles and the set of parameters generating cusps, mutliple, extreme, and isolated points. We do so, by solving an over-determined bivariate polynomial system and univariate polynomial equations. Then, we classify the parameters that appear in the solutions, by exploiting the fact the system is symmetric. For sake of completeness, we describe the procedure in Alg. 2.

To compute the RUR of an overdetermined bivariate system (Thm. 4.6), we employ Lem. 4.4 and Prop. 4.5, which adapt the techniques used in [8] to our setting.

Lemma 4.4. *Let $f, g, h_1, \ldots, h_n \in \mathbb{Z}[X, Y]$ with degrees bounded by δ and bitsize of coefficients bounded by L. Computing a common separating element in the form $X + \alpha Y, \alpha \in \mathbb{Z}$ for the n+1 systems of bivariate polynomial equations $\{f = g = 0\}$, $\{f = h_i = 0\}$, $i = 1 \ldots n$ needs $\widetilde{O}_B(n(\delta^6 + \delta^5 L))$ bit operations in the worst case, and $\widetilde{O}_B(n(\delta^5 + \delta^4 L))$ in the expected case with a Las Vegas Algorithm. Moreover, the bitsize of α does not exceed* $\log(2n\delta^4)$.

Proof. A straightforward strategy consists in running simultaneously Algorithm 5 (worst case) or Algorithm 5' (Las Vegas) from [8] on all the systems. The only modifications needed are that the values of α to be considered are less than $2n\delta^4$ (twice a bound on the total number of solutions of all the systems) and that the exit test is valid if and only if it is valid for all the systems. □

Proposition 4.5. *Let $f, g \in \mathbb{Z}[X, Y]$ with degrees bounded by δ and coefficients' bitsizes bounded by L. We can compute a rational parameterization $\{h(T), X = \frac{h_X(T)}{h_1(T)}, Y = \frac{h_Y(T)}{h_1(T)}\}$ of f, g with $h, h_1, h_X, h_Y \in \mathbb{Z}[T]$ with degrees less than δ^2 and coefficients' bitsizes in $\widetilde{O}(\delta(L + \delta))$, in $\widetilde{O}_B(\delta^5(L + \delta))$ bit operations in the worst case and $\widetilde{O}_B(\delta^4(L + \delta))$ expected bit operations with a Las Vegas Algorithm.*

Algorithm 2: `Special_Points`(ϕ)

Input: Proper parametrization $\phi \in \mathbb{Z}(t)^n$ without singularity at infinity, as in Eq. (1)
Output: Real poles and parameters that give real cusps, multiple, isolated and extreme points.

```
/* The subroutines SOLVE_R and SOLVE_C return the solution set of a univariate
   polynomial or a system of polynomials over the real and complex numbers resp.   */
Compute polynomials h_1(s,t), ..., h_n(s,t)
T_P^R ← ⋃_{i∈n} SOLVE_R(q_i(t) = 0)
T_P^C ← ⋃_{i∈[n]} SOLVE_C(q_i(t) = 0)
S ← SOLVE_C(h_1(s,t) = 0, ..., h_n(s,t) = 0)
T_C, T_M, T_I, W ← ∅
for (s,t) ∈ S do
    if s = t and s ∈ ℝ \ T_P^R then
        T_C ← T_C ∪ {t}
    end
    else if s ≠ t then
        if s ∈ ℝ \ T_P^R then
            if t ∈ ℝ then
                T_M ← T_M ∪ {t}
            end
            else
                W ← W ∪ {t}
            end
        end
        else if s = t̄ and s ∉ T_P^C then
            T_I ← T_I ∪ {t}
        end
    end
end
T_I ← T_I \ W
/* Extreme points                                                                   */
T_E ← ⋃_{i∈n} SOLVE_R(h_i(t,t) = 0)
T_E ← T_E \ (T_E ∩ (T_C ∪ T_M))
```

Proof. Algorithms 6 and 6' from [8] compute a RUR decomposition of $f = g = 0$ in $\widetilde{O}_B(\delta^5(L+\delta))$ bit operations in the worst case and $\widetilde{O}_B(\delta^4(L+\delta))$ expected bit operations with a Las Vegas Algorithm respectively. They provide $s \leqslant \delta$ parameterizations in the form $\{h_i(T), \frac{h_{i,X}(T)}{h_{i,1}(T)}, \frac{h_{i,Y}(T)}{h_{i,1}(T)}\}$, where $i = 1..s$, with the following properties:

- $\prod_{i=1}^{s} h_i$ is a polynomial of degree at most δ^2 with coefficients of bitsize $\widetilde{O}(\delta L + \delta^2)$.
- The degrees of $h_{i,1}(T), h_{X,1}(T)$ and $h_{Y,1}(T)$ are less then the degree of h_i.
- The coefficients' bitsizes of $h_{i,1}(T), h_{X,1}(T)$ and $h_{Y,1}(T)$ are in $\widetilde{O}_B(\delta L + \delta^2)$.

Also, $\prod_{i=1}^{s} h_i, \frac{\sum_{n=1}^{n} h_{j,X} \prod_{i\neq j} h_i}{\sum_{n=1}^{n} h_{j,1} \prod_{i\neq j} h_i}, \frac{\sum_{n=1}^{n} h_{j,Y} \prod_{i\neq j} h_i}{\sum_{n=1}^{n} h_{j,1} \prod_{i\neq j} h_i}$ is a rational parameterization of the system $\{f = g = 0\}$, defined by polynomials of degree less than δ^2 with coefficients of bitsizes $\widetilde{O}(\delta(L+\delta))$ and can be computed from the RUR decomposition performing $O(s)$ multiplications of polynomials of degree at most δ^2 with coefficients of bitsize $\widetilde{O}(\delta(L+\delta))$, which requires $\widetilde{O}_B(\delta^4(L+\delta))$ bit operations. □

Theorem 4.6. *There exists an algorithm that computes the RUR and the isolating boxes of the roots of the system $\{h_1(s,t) = \cdots = h_n(s,t) = 0\}$ with worst-case bit complexity $\widetilde{O}_B(n(d^6 + d^5\tau))$. There is also a Las Vegas variant with expected complexity $\widetilde{O}_B(d^6 + nd^5 + d^5\tau + nd^4\tau)$.*

Proof. Assume that we know a common separating linear element $\ell(s,t) = \ell_0 + \ell_1 s + \ell_2 t$ that separates the roots of the n-1 systems of bivariate polynomial equations $\{h_1 = h_2 = 0\}$, $\{h_1 = h_i = 0\}$, for $3 \leq i \leq n$. We can compute ℓ with $\widetilde{O}_B(n(d^6 + d^5\tau))$ bit operations in the worst case and with $\widetilde{O}_B(n(d^5 + d^4\tau))$ expected bit operations with a Las Vegas algorithm (Lem. 4.4).

We denote by $\{r(T), \frac{r_s(T)}{r_I(T)}, \frac{r_t(T)}{r_I(T)}\}$ a RUR for $\{h_1 = h_2 = 0\}$ with respect to ℓ. In addition, for $i = 3 \ldots n$, let $\{r_i(T), \frac{r_{i,s}(T)}{r_{i,I}(T)}, \frac{r_{i,t}(T)}{r_{i,I}(T)}\}$ be the RUR of $\{h_1 = h_i = 0\}$, also with respect to ℓ. We can compute all these representations with $\widetilde{O}_B(n(d^6 + d^5\tau))$ bit operations in the worst case, and with $\widetilde{O}_B(n(d^5 + d^4\tau))$ in expected case with a Las Vegas algorithm (Lem. 4.5).

Then, for the system $\{h_1 = h_2 = \ldots = h_n = 0\}$ we can define a rational parameterization $\{\chi(T), \frac{r_s(T)}{r_I(T)}, \frac{r_t(T)}{r_I(T)}\}$,

$$\begin{aligned}\chi(T) = \gcd(\quad & r(T), r_3(T), \ldots, r_n(T),\\ & r_s(T)r_{3,I}(T) - r_{3,s}(T)r_I(T), r_t(T)r_{3,I}(T) - r_{3,t}(T)r_I(T),\\ & \vdots\\ & r_s(T)r_{n,I}(T) - r_{n,s}(T)r_I(T), r_t(T)r_{n,I}(T) - r_{n,t}(T)r_I(T)).\end{aligned}$$

where

So, to compute such a parameterization we still need to compute the gcd of $3n-5$ univariate polynomials of degrees at most d^2 and coefficients of bitsizes in $\widetilde{O}(d\tau)$ which needs $\widetilde{O}_B(n(d^6 + d^4\tau))$ bit operations in the worst case. Isolating the roots of such a parameterization requires $\widetilde{O}_B(d^6 + d^5\tau)$ according to Alg. 7 from [8]. □

Remark 4 (RUR and isolating interval representation). *If we use Thm.4.6 to solve the over-determined bivariate system of the h_i polynomials of Eq. (2), then we obtain in the output a RUR for the roots, which is as follows: There is a polynomial $\chi(T) \in \mathbb{Z}[T]$ of size $(O(d^2), \widetilde{O}(d^2 + d\tau))$ and a mapping*

$$\begin{aligned} V(\chi) &\to V(h_1, \ldots, h_n)\\ T &\mapsto \left(\frac{r_s(T)}{r_I(T)}, \frac{r_t(T)}{r_I(T)}\right), \end{aligned} \tag{3}$$

that defines an one-to-one correspondence between the roots of χ and those of the system. The polynomials r_s, r_t, and r_I are in $\mathbb{Z}[T]$ and have also size $(O(d^2), \widetilde{O}(d^2 + d\tau))$.

Taking into account the cost to compute this parametrization of the solutions (Thm.4.6), we can also compute at no extra cost the resultant of $\{h_1, h_2\}$ with respect to s or t. Notice that both resultants are the same polynomial, since the system is symmetric. Let $R_s(t) = \mathtt{res}_s(h_1, h_2)$. It is of size $(O(d^2), O(d^2 + d\tau))$ [4, Prop. 8.46].

Under the same bit complexity, we can sufficiently refine the isolating boxes of the solutions of the bivariate system (computed in Thm.4.6), so that every root $(\frac{r_s(\xi)}{r_I(\xi)}, \frac{r_t(\xi)}{r_I(\xi)})$, where $\chi(\xi) = 0$, has a representation as a pair of algebraic numbers in isolating interval representation:

$$((R_s, I_{1,\xi} \times I_{2,\xi}), (R_s, J_{1,\xi} \times J_{2,\xi})). \tag{4}$$

Both coordinates are roots of the same polynomial. Moreover, $I_{2,\xi}, J_{2,\xi}$ are empty sets when the corresponding algebraic number is real. Therefore, we can immediately distinguish between real and complex parameters. At the same time,

we associate to each isolating box of a root of R_s the algebraic numbers $\rho = (\chi, I_\rho \times J_\rho)$ for whom it holds that $\frac{r_s(\rho)}{r_I(\rho)}$ projects inside this isolating box. We can interchange between the two of representations in constant time, and this will simplify our computations in the sequel.

Lemma 4.7. *Let C be a curve with a proper parametrization $\phi(t)$ as in (1), that has no singularities at infinity. We compute the real poles of ϕ and the parameters corresponding to singular, extreme, and isolated points of C in worst-case bit complexity*

$$\widetilde{O}_B(nd^6 + nd^5\tau + d^4(n^2 + n\tau) + d^3(n^2\tau + n^3) + n^3d^2\tau),$$

and using a Las Vegas algorithm in expected bit complexity

$$\widetilde{O}_B(d^6 + d^5(n + \tau) + d^4(n^2 + n\tau) + d^3(n^2\tau + n^3) + n^3d^2\tau).$$

Proof. The proof is an immediate consequence of the following:

• *We compute all $h_i \in \mathbb{Z}[s,t]$ in $\widetilde{O}_B(nd^2\tau)$*: To construct each h_i we perform d^2 multiplications of numbers of bitsize τ; the cost for this is $\widetilde{O}_B(d^2\tau)$. The bi-degree of each is at most (d,d) and $\mathcal{L}(h_i) \leq 2\tau + 1 = O(\tau)$.

• *The real poles of ϕ are computed in $\widetilde{O}_B(n^2(d^4 + d^3\tau))$:* To find the poles of ϕ, we isolate the real roots of each polynomial $q_i(t)$, for $i \in [n]$. This costs $\widetilde{O}_B(n(d^3 + d^2\tau))$ [28]. Then we sort the roots in $\widetilde{O}_B(n\, d\, n(d^3 + d^2\tau)) = \widetilde{O}_B(n^2(d^4 + d^3\tau))$.

• *The parameters corresponding to cusps, multiple and isolated points of C are computed in $\widetilde{O}_B(n(d^6 + d^5\tau))$:* We solve the bivariate system (2) in $\widetilde{O}_B(n(d^6 + d^5\tau))$ or in expected time $\widetilde{O}_B(d^6 + nd^5 + d^5\tau + nd^4\tau)$ (Thm. 4.6). Then we have a parametrization of the solutions of the bivariate system (2) of the form (3) and in the same time of the form (4) (see Rem. 4). Some solutions $(s,t) \in S$ may not correspond to points on the curve, since s,t can be poles of ϕ. Notice that from Rem. 3, s and t are either both poles or none of them is a pole. We compute $g_s = \gcd(R_s, Q)$, where $Q(t) = \prod_{i\in[n]} q_i(t)$, and the gcd-free part of R_s with respect to Q. This is done in $\widetilde{O}_B(\max\{n,d\}\,(nd^3\tau + nd^2\tau^2))$ [10, Lem. 5].
Every root of R_s^* is an algebraic number of the form $(R_s, I_{1,\xi} \times I_{2,\xi})$, for some ξ that is root of χ. We can easily determine if it corresponds to a cusp, a multiple or an isolated point; when real (i.e., $I_{2,\xi} = \emptyset$) it corresponds to a cusp of C if and only if $((R_s, I_{1,\xi}), (R_s, I_{1,\xi}))$ is in S. Otherwise, it corresponds to a multiple point. When it is complex (i.e., $I_{2,\xi} \neq \emptyset$), it corresponds to an isolated pont of C if and only if $((R_s, I_{1,\xi} \times I_{2,\xi}), (R_s, I_{1,\xi} \times (-I_{2,\xi}))) \in S$ and there is no root in S of the form $((R_s, I_{1,\xi} \times I_{2,\xi}), (R_s, J_{1,\xi'}))$.

• *The parameters corresponding to extreme points of C are computed in $\widetilde{O}_B(d^4n\tau + d^3(n^2\tau + n^3) + d^2n^3\tau)$:* For all $i \in [n]$, $h_i(t,t)$ is a univariate polynomial of size $(O(d), O(\tau))$. Then, $H(t) = \prod_{i\in[n]} h_i(t,t)$ is of size $(O(nd), \widetilde{O}(n\tau))$. The parameters that correspond to extreme points are among the roots of $H(t)$. To make sure that poles and parameters that give singular points are excluded, we compute $\gcd(H, Q \cdot R_s)$, where $Q(t) = \prod_{i\in[n]} q_i(t)$, and the gcd-free part of H with respect to $Q \cdot R_s$, say H^*. Since $Q \cdot R_s$ is a polynomial of size $(d^2 + nd, (d+n)\tau)$, the computation of the gcd and the gcd-free part costs $\widetilde{O}_B(n(d^4\tau + nd^3\tau + n^2d^2\tau))$ [10, Lem. 5]. Then, $H = \gcd(H, Q \cdot R_s)H^*$, and the real roots of H^* give the parameters that correspond to extreme points. We isolate the real roots of H^* in $\widetilde{O}_B(n^3(d^3 + d^2\tau))$, since it is a polynomial of size $(O(nd), \widetilde{O}(n(d+\tau)))$. □

5 PTOPO: TOPOLOGY AND COMPLEXITY

We present PTOPO, an algorithm to construct an abstract graph G that is isotopic [7, p.184] to C when we embed it in $\mathbb{R}^n$. We emphasize that, currently, we do not treat/compute knots in the case of space curves. The embedding consists of a graph whose vertices are points on the curve given by their parameter values. The edges are smooth parametric arcs that we can continuously deform to branches of C without any topological changes. We need to specify a bounding box in $\mathbb{R}^n$ inside which the constructed graph results in an isotopic embedding to C. We

comment at the end of the section on the case where an arbitrary box is provided at the input. We determine a bounding box in $\mathbb{R}^n$, which we call *characteristic*, that captures all the topological information of C:

Definition 2. *A characteristic box of C is a box enclosing a subset of $\mathbb{R}^n$ that intersects all components of C and contains all its singular, extreme, and isolated points.*

Let $\mathcal{B}_C$ be a characteristic box of C. If C is bounded, then $C \subset \mathcal{B}_C$. If C is unbounded, then the branches of C that extend to infinity intersect the boundary of $\mathcal{B}_C$. A branch of the curve extends to infinity if for $t \to t_0$, it holds $||\phi(t)|| > M$, for any $M > 0$, where $t_0 \in \mathbb{R} \cup \{\infty\}$. Lem. 5.1 computes a characteristic box using the degree and bitsize of the polynomials in the parametrization (1).

Lemma 5.1. *Let C be a curve with a parametrization as in (1). For $b = 15d^2(\tau + \log d) = O(d^2\tau)$, $\mathcal{B}_C = [-2^b, 2^b]^n$ is a characteristic box of C.*

Proof. We estimate the maximum and minimum values of ϕ_i, $i \in [n]$, when we evaluate it at the parameter values that correspond to special points and also at each pole that is not a root of q_i.

Let t_0 be a parameter that corresponds to a cusp or an extreme point with respect to the i-th direction. Then, it is a root of $\phi_i'(t)$. Let $N(t) = p_i'(t)q_i(t) - p_i(t)q_i'(t)$ the numerator of $\phi_i'(t)$. Then $N(t_0) = 0$. The degree of $N(t)$ is $\leq 2d - 1$ and $\mathcal{L}(N) \leq 2^{2\tau + \log d + 1}$. From Lem. 2.1 we conclude that $|p_i(t_0)| \leq 2^{4d\tau + d\log(d) + (3d-1)\log(3d-1) + d - \tau}$. Analogously, it holds that $|q_i(t_0)| \geq 2^{-4d\tau - d\log(d) - (3d-1)\log(3d-1) - d + \tau}$. Therefore,

$$|\phi_i(t_0)| \leq 2^{2(4d\tau + d\log(d) + (3d-1)\log(3d-1) + d - \tau)}.$$

Now, let (t_1, t_2) be two parameters corresponding to a multiple point of C, i.e., (t_1, t_2) is a root of the bivariate system in Eq. (2). Take any $j, k \in [n]$ with $j \neq k$ and let $R(t) = \mathtt{res}_s(h_j, h_k)$. It holds that $R(t_1) = 0$. The degree of R is $\leq 2d^2$ and $\mathcal{L}(R) \leq 2d(\tau + \log(d) + \log(d+1) + 1)$ [4, Prop. 8.29]. Applying Lem. 2.1, we deduce that

$$|\phi_i(t_1)| \leq 2^{4d^2(\tau + \log(d) + \log(d+1) + 1) + 4d^2\tau + (2d^2 + d)\log(2d^2 + d)}.$$

Let t_3 be a pole of ϕ with $q_j(t_3) = 0$, for some $j \neq i$. If $\phi_i(t_3)$ is defined, applying Lem. 2.1 gives

$$|\phi_i(t_3)| \leq 2^{4d\tau + 4d\log 2d}.$$

To conclude, we take the maximum of the three bounds. However, to simplify notation, we slightly overestimate the latter bound. □

The vertices of the embedded graph must include the singular and the isolated points of C. Additionally, to rigorously visualize the geometry of C, we consider as vertices the extreme points of C, with respect to all coordinate directions, as well as the intersections of C with the boundary of the bounding box. We label the vertices of G using the corresponding parameter values generating these points and we connect them accordingly. Alg. 3 presents the pseudo-code of PTOPO and here we give some more details on the various steps. We construct G as follows:

First, we compute the poles and the sets $\mathsf{T}_C, \mathsf{T}_M, \mathsf{T}_E$, and T_I of parameters corresponding to "special points". Then, we compute the characteristic box of C, say $\mathcal{B}_C$. We compute the set T_B of parameters corresponding to the intersections of C with the boundary of $\mathcal{B}_C$ (if any). Lem. 5.2 describes this procedure and its complexity.

Lemma 5.2. *Let $\mathcal{B} = [l_1, r_1] \times \cdots \times [l_n, r_n]$ in $\mathbb{R}^n$ and $\mathcal{L}(l_i) = \mathcal{L}(r_i) = \sigma$, for $i \in [n]$. We can find the parameters that give the intersection points of ϕ with the boundary of $\mathcal{B}$ in $\widetilde{O}_B(n^2d^3 + n^2d^2(\tau + \sigma))$.*

Proof. For each $i \in [n]$ the polynomials $q_i(t)l_i - p_i(t) = 0$ and $q_i(t)r_i - p_i(t) = 0$ are of size $(O(d), O(\tau + \sigma))$. So, we compute isolating intervals for all their real solutions in $\widetilde{O}_B(d^2(\tau + \sigma))$ [27]. For any root t_0 of each of these polynomials, since ϕ is in reduced form (by assumption), we have $t_0 \notin \mathsf{T}_P^{\mathbb{R}}$. We check if $\phi_j(t_0) \in [l_j, r_j]$, $j \in [n] \setminus i$. This requires 3 sign evaluations of univariate polynomials of size $(d, \tau + \sigma)$ at all roots of a polynomial

of size $(d, \tau+\sigma)$. The bit complexity of performing these operations for all the roots is $\widetilde{O}_B(d^3 + d^2(\tau+\sigma))$ [40, Prop. 6]. Since we repeat this procedure $n-1$ times for every $i \in [n]$, the total cost is $\widetilde{O}_B(n^2d^3 + n^2d^2(\tau+\sigma))$. □

We partition $\mathsf{T}_C \cup \mathsf{T}_M \cup \mathsf{T}_E \cup \mathsf{T}_I \cup \mathsf{T}_B$ into groups of parameters that correspond to the same point on C. For each group, we add a vertex to G if and only if the corresponding point is inside the bounding box $\mathcal{B}$; for the characteristic box it is inside by construction.

Lemma 5.3. *The graph G has $\kappa = O(d^2+nd)$ vertices, which can be computed using $O(d^2+nd)$ arithmetic operations.*

Proof. Since $\mathsf{T}_B \cap \mathsf{T}_M = \emptyset$ and $\mathsf{T}_E \cap \mathsf{T}_M = \emptyset$, to each parameter in T_B and T_E corresponds a unique point on C. So for every $t \in \mathsf{T}_B \cup \mathsf{T}_E$ we add a vertex to G, labeled by the respective parameter. Next, we group the parameters in $\mathsf{T}_C \cup \mathsf{T}_M \cup \mathsf{T}_I$ that give the same point on C and we add for each group a vertex at G labeled by the corresponding parameter values.

Grouping of the parameters is done as follows: For every $t \in \mathsf{T}_C \cup \mathsf{T}_M$ we add a vertex to G labeled by the set $\{s \in \mathbb{R} : (s,t) \in S\} \cup \{t\}$ and for every $t \in \mathsf{T}_I$ we add a vertex to G labeled by the set $\{s \in \mathbb{C} : (s,t) \in S\} \cup \{t\}$. Notice that we took into account Rem.3. We compute these sets simply by reading the elements of S.

It holds that $\mathsf{T}_B = O(nd)$, $\mathsf{T}_E = O(nd)$ and $|S| = O(d^2)$. Since for each vertex, we can find the parameters that give the same point in constant time, the result follows. □

We denote by $v_1, \ldots, v_\kappa$ the vertices (with distinct labels) of G and by $\lambda(v_1), \ldots, \lambda(v_\kappa)$ their label sets (i.e., the parameters that correspond to each vertex). Let T be the *sorted* list of parameters in $\mathsf{T}_C \cup \mathsf{T}_M \cup \mathsf{T}_E \cup \mathsf{T}_B$[2]. If for two consecutive elements $t_1 < t_2$ in T, there exists a pole $s \in \mathsf{T}_P^{\mathbb{R}}$ such that $t_1 < s < t_2$, then we split T into two lists: T_1 containing the elements $\leq t_1$ and T_2 containing the elements $\geq t_2$. We continue recursively for T_1 and T_2, until there are no poles between any two elements of the resulting list. This procedure partitions T into $\mathsf{T}_1, \ldots, \mathsf{T}_\ell$.

To add edges to G, we consider each T_i with more than one element, where $i \in [\ell]$, independently. For any consecutive elements $t_1 < t_2$ in T_i, with $t_1 \in \lambda(v_{i,1})$ and $t_2 \in \lambda(v_{i,2})$, we add the edge $\{v_{i,1}, v_{i,2}\}$[3]. If $\mathbf{p}_\infty$ exists, we add an edge to the graph connecting the vertices corresponding to the last element of T_ℓ and the first element of the T_1.

Theorem 5.4 (PTOPO inside the characteristic box). *Consider a proper parametrization ϕ of curve C involving polynomials of degree d and bitsize τ, as (1). Alg. 3 outputs a graph G that, if embedded in $\mathbb{R}^n$, is isotopic to C, within the characteristic box $\mathcal{B}_C$. It has worst case complexity*

$$\widetilde{O}_B(d^6(n+\tau) + nd^5\tau + n^2d^4\tau + d^3(n^2\tau + n^3) + n^3d^2\tau),$$

while its expected complexity is

$$\widetilde{O}_B(d^6\tau + nd^5\tau + n^2d^4\tau + d^3(n^2\tau + n^3) + n^3d^2\tau).$$

If $n = O(1)$, then bounds become $\widetilde{O}_B(N^7)$, where $N = \max\{d, \tau\}$.

Proof. We count on the fact that ϕ is continuous in $\mathbb{R} \setminus \mathsf{T}_P^{\mathbb{R}}$. Thus, for each real interval $[s,t]$ with $[s,t] \cap \mathsf{T}_P^{\mathbb{R}} = \emptyset$, there is a parametric arc connecting the points $\phi(s)$ and $\phi(t)$. Since for any (sorted) list T_i, for $i \in [\ell]$, the interval defined by the minimum and maximum value of its elements has empty intersection with $\mathsf{T}_P^{\mathbb{R}}$, then for any $s, t \in \mathsf{T}_i$ there exists a parametric arc connecting $\phi(s)$ and $\phi(t)$ and it is entirely contained in $\mathcal{B}_C$. If $\mathbf{p}_\infty$ exists, then $\mathbf{p}_\infty$ is inside $\mathcal{B}_C$. Let $t_{1,1}, t_{\ell,k_\ell}$ be the first element of the first list and the last element of the last list. There is a parametric arc connecting $\phi(t_{1,1})$ with $\mathbf{p}_\infty$ and $\mathbf{p}_\infty$ with $\phi(t_{\ell,k_\ell})$. So we add the edge $\{t_{1,1}, t_{\ell,\kappa_\ell}\}$ to G. Then, every

[2]Notice that we exclude the parameters of the isolated points.

[3]To avoid multiple edges, we make the convention that we add an edge between $v_{i,j}$, $j = 1, 2$, and an (artificial) intermediate point corresponding to a parameter in (t_1, t_2).

Algorithm 3: PTOPO(ϕ) (Inside the characteristic box)

Input: A proper parametrization $\phi \in \mathbb{Z}(t)^n$ without singular points at infinity.
Output: Abstract graph G

1 Compute real poles $\mathsf{T}_P^{\mathbb{R}}$.
2 Compute parameters of 'special points' $\mathsf{T}_C, \mathsf{T}_M, \mathsf{T}_E, \mathsf{T}_I$.
/* Characteristic box */
3 $b \leftarrow 15d^2(\tau + \log d)$, $\mathcal{B}_C \leftarrow [-2^b, 2^b]^n$
4 $\mathsf{T}_B \leftarrow$ parameters that give to intersections of C with $\mathcal{B}_C$
5 Construct the set of vertices of G using Lem.5.3
6 Sort the list of all the parameters $\mathsf{T} = [\mathsf{T}_C, \mathsf{T}_M, \mathsf{T}_E, \mathsf{T}_B]$.
7 Let $T_1, \ldots, T_\ell$ the sublists of T when split at parameters in $\mathsf{T}_P^{\mathbb{R}}$
8 **for** *every list* $\mathsf{T}_i = [t_{i,1}, \ldots, t_{i,k_i}]$ **do**
9 **for** $j = 1, \ldots, k_i - 1$ **do**
10 Add the edge $\{t_{i,j}, t_{i,j+1}\}$ to the graph
11 **end**
12 **end**
13 **if** $\mathbf{p}_\infty$ *exists* **then**
14 Add the edge $\{t_{1,1}, t_{\ell,k_\ell}\}$ to the graph
15 **end**

edge of G is embedded to a unique smooth parametric arc and the embedding of G can be trivially continuously deformed to C.

For the complexity analysis, we know from Lem.4.7 that steps 1-2 can be performed in wost-case bit complexity

$$\widetilde{O}_B(nd^6 + nd^5\tau + d^4(n^2 + n\tau) + d^3(n^2\tau + n^3) + n^3d^2\tau),$$

and in expected bit complexity

$$\widetilde{O}_B(d^6 + d^5(n + \tau) + d^4(n^2 + n\tau) + d^3(n^2\tau + n^3) + n^3d^2\tau),$$

using a Las Vegas algorithm. From Lemmata 5.1, 5.2, and 5.3 steps 4-5 cost $\widetilde{O}_B(n^2(d^3\tau))$.

To perform steps 6-7 we must sort all the parameters in $\mathsf{T} \cup \mathsf{T}_P^{\mathbb{R}}$, i.e., we sort $O(d^2 + nd)$ algebraic numbers: The parameters that correspond to cusps and extreme points can be expressed as roots of $\prod_{i\in[n]} h_i(t, t)$, which is of size $(nd, n\tau)$. The poles are roots of $\prod_{i\in[n]} q_i(t)$, which has size $(nd, n\tau)$. The parameters that correspond to multiple points are roots of R_s which has size $(d^2, d\tau)$. At last, parameters in T_B are roots of a polynomial of size $(d, d^2\tau)$.

We can consider all these algebraic numbers together as roots of a single univariate polynomial (the product of all the corresponding polynomials). It has degree $O(d^2+nd)$ and bitsize $\widetilde{O}(d^2\tau+n\tau)$. Hence, its separation bound is $\widetilde{O}(d^4\tau+nd^3\tau+nd^2\tau+n^2d\tau)$. To sort the list of all the algebraic numbers we have to perform $O(d^2+nd)$ comparisons and each costs $\widetilde{O}(d^4\tau+nd^3\tau+nd^2\tau+n^2d\tau)$. Thus, the overall cost for sorting is $\widetilde{O}_B(d^6\tau+nd^5\tau+n^2d^4\tau+n^2d^3\tau+n^3d^2\tau)$. The overall bit complexities in the worst and expected case follow by summing the previous bounds. □

Following the proof of Thm. 5.4 we notice that the term $d^6\tau$ in the worst case bound is due to the introduction of the intersection points of C with $\mathcal{B}_C$. For visualizing the curve within $\mathcal{B}_C$, these points are essential and we

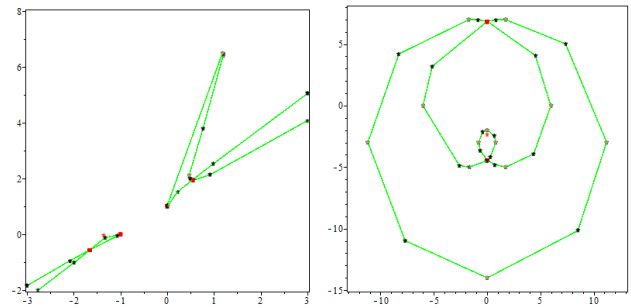

Fig. 1. *The left figure is the output of* PTOPO *for the parametric curve* $\left(\frac{3\,t^2+3\,t+1}{t^6-2\,t^4-3\,t-1}, \frac{(t^4-2\,t+2)t^2}{t^6-2\,t^4-3\,t-1}\right)$, *while the right figure is the output for the curve* $\left(\frac{6\,t^8-756\,t^6+3456\,t^5-31104\,t^3+61236\,t^2-39366}{t^8+36\,t^6+486\,t^4+2916\,t^2+6561}, \frac{-18(6\,t^6-16\,t^5-126\,t^4+864\,t^3-1134\,t^2-1296\,t+4374)t}{t^8+36\,t^6+486\,t^4+2916\,t^2+6561}\right)$.

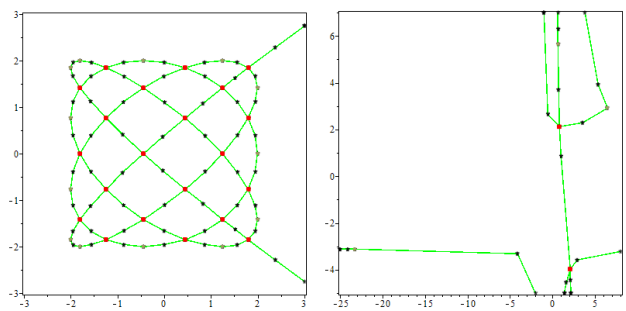

Fig. 2. *The left figure is the output of* PTOPO *for the parametric curve* $(t^8 - 8\,t^6 + 20\,t^4 - 16\,t^2 + 2, t^7 - 7\,t^5 + 14\,t^3 - 7\,t)$ *while the right figure is the output for the curve* $\left(\frac{37\,t^3-23\,t^2+87\,t+44}{29\,t^3+98\,t^2-23\,t+10}, \frac{-61\,t^3-8\,t^2-29\,t+95}{11\,t^3-49\,t^2-47\,t+40}\right)$.

cannot avoid them. However, if we are interested only in the topology of C, i.e., the abstract graph G, these points are not important any more. We sketch a procedure to avoid them and gain a factor of d in the complexity bound:

Assume that we have not computed the points on $C \cap \mathcal{B}_C$. We split again the sorted list $\mathsf{T} = [\mathsf{T}_C, \mathsf{T}_M, \mathsf{T}_E]$ at the real poles, and we add an artificial parameter at the beginning and at the end of each sublist. The rest of the procedure remains unaltered.

To verify the correctness of this approach, it suffices to prove that the graph that we obtain by this procedure, is isomorphic to the graph G. It is immediate to see that the latter holds, possibly up to the dissolution of the vertices corresponding to the first and last artificial vertices. Adding these artificial parameters does not affect the overall complexity, since we do not perform any algebraic operations. Therefore, the bit complexity of the algorithm is determined by the complexity of computing the parameters of the special points (Lem.4.7), and so we have the following theorem:

Theorem 5.5 (PTOPO and an abstract graph). *Consider a proper parametrization ϕ of curve C involving polynomials of degree d and bitsize τ, as (1). Alg. 3 outputs a graph G that, if we embed it in $\mathbb{R}^n$, then it is isotopic to C. It has worst case complexity*

$$\widetilde{O}_B(nd^6 + nd^5\tau + d^4(n^2 + n\tau) + d^3(n^2\tau + n^3) + n^3d^2\tau),$$

while its expected complexity is

$$\widetilde{O}_B(d^6 + d^5(n + \tau) + d^4(n^2 + n\tau) + d^3(n^2\tau + n^3) + n^3d^2\tau),$$

If $n = O(1)$, then bounds become $\widetilde{O}_B(N^6)$, where $N = \max\{d, \tau\}$.

Remark 5. *If we are given a box $\mathcal{B} \subset \mathbb{R}^n$ at the input, we slightly modify* PTOPO*, as follows: We discard the parameter values in $\mathsf{T}_C \cup \mathsf{T}_M \cup \mathsf{T}_E \cup \mathsf{T}_I$ that correspond to points not contained in $\mathcal{B}$. The set of G's vertices is constructed similarly. To connect the vertices, we follow the same method with a minor modification: For any consecutive lements $t_1 < t_2$ in a list T_i with more than two elements, such that $t_1 \in \lambda(v_{i,1})$ and $t_2 \in \lambda(v_{i,2})$, we add the edge $\{v_{i,1}, v_{i,2}\}$ if and only if $\phi(t_1)$, $\phi(t_2)$ are not both on the boundary of $\mathcal{B}$; or in other words t_1 and t_2 are not both in T_B.*

6 IMPLEMENTATION AND EXAMPLES

We have implemented PTOPO in maple[4]. We build upon the real root isolation routines of maple's RootFinding library and the slv package [15], in order to use a certified implementation of general purpose exact computations with one and two real algebraic numbers, like comparison and sign evaluations. PTOPO computes the topology and visualizes parametric curves (currently planar).

To demonstrate its capabilities, we present in Fig. 1 and Fig. 2 the topology of four planar curves from [3]. For a given parametric representation of a curve, PTOPO computes the special points on the curve, the characteristic box, the corresponding graph, and then it visualizes the curve (inside the box). The computation, in all cases, takes less than a second in a MacBook laptop, running maple 2019. The red squares correspond to cusps or multiple points, the khaki squares correspond to extreme points, and the black stars correspond to intermediate or boundary points.

Acknowledgments FR, ET and ZZ are partially supported by Fondation Mathématique Jacques Hadamard PGMO grand ALMA, Agence Nationale de la Recherche ANR-17-CE40-0009, Tübitak 117F100, 118F321, 118C240 and PHC GRAPE.

REFERENCES

[1] S. S. Abhyankar and C. J. Bajaj. Automatic parameterization of rational curves and surfaces IV: Algebraic space curves. *ACM Trans. Graph.*, 8(4):325–334, 1989.

[2] L. Alberti, B. Mourrain, and J. Wintz. Topology and arrangement computation of semi-algebraic planar curves. *CAGD*, 25(8):631 – 651, 2008.

[3] J. G. Alcázar and G. M. Díaz-Toca. Topology of 2D and 3D rational curves. *CAGD*, 27(7):483 – 502, 2010.

[4] S. Basu, R. Pollack, and M-F.Roy. *Algorithms in Real Algebraic Geometry*, volume 10 of *Algorithms and Computation in Mathematics*. Springer-Verlag, 2003.

[5] A. Bernardi, A. Gimigliano, and M. Idà. Singularities of plane rational curves via projections. *J. Symb. Comput.*, 09 2016.

[6] A. Blasco and S. Pérez-Díaz. An in depth analysis, via resultants, of the singularities of a parametric curve. *CAGD*, 68:22–47, 2019.

[7] J.-D. Boissonnat and M. Teillaud, editors. *Effective Computational Geometry for Curves and Surfaces.* Springer-Verlag, Mathematics and Visualization, 2006.

[8] Y. Bouzidi, S. Lazard, G. Moroz, M. Pouget, F. Rouillier, and M. Sagraloff. Solving bivariate systems using Rational Univariate Representations. *J. Complexity*, 37:34–75, 2016.

[9] Y. Bouzidi, S. Lazard, M. Pouget, and F. Rouillier. Rational univariate representations of bivariate systems and applications. In *Proc. 38th Int'l Symp. on Symbolic and Algebraic Computation*, ISSAC '13, pages 109–116, NY, USA, 2013. ACM.

[10] Y. Bouzidi, S. Lazard, M. Pouget, and F. Rouillier. Separating linear forms and rational univariate representations of bivariate systems. *J. Symb. Comput.*, pages 84–119, 2015.

[11] L. Busé, C. Laroche, and F. Yıldırım. Implicitizing rational curves by the method of moving quadrics. *Computer-Aided Design*, 114:101–111, 2019.

[12] J. Caravantes, M. Fioravanti, L. Gonzalez-Vega, and I. Necula. Computing the topology of an arrangement of implicit and parametric curves given by values. In V. P. Gerdt, W. Koepf, W. M. Seiler, and E. V. Vorozhtsov, editors, *Computer Algebra in Scientific Computing*, pages 59–73, Cham, 2014. Springer.

[4] https://webusers.imj-prg.fr/~christina.katsamaki/ptopo/

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 754362.

[13] D. Cox, A. Kustin, C. Polini, and B. Ulrich. A study of singularities on rational curves via syzygies. *Memoirs of the American Mathematical Society*, 222, 02 2011.

[14] D. N. Diatta, S. Diatta, F. Rouillier, M.-F. Roy, and M. Sagraloff. Bounds for polynomials on algebraic numbers and application to curve topology. *arXiv preprint arXiv:1807.10622*, 2018.

[15] D. I. Diochnos, I. Z. Emiris, and E. P. Tsigaridas. On the asymptotic and practical complexity of solving bivariate systems over the reals. *J. Symb. Comput.*, 44(7):818–835, 2009. (Special issue on ISSAC 2007).

[16] R. T. Farouki, C. Giannelli, and A. Sestini. Geometric design using space curves with rational rotation-minimizing frames. In M. Dæhlen, M. Floater, T. Lyche, J.-L. Merrien, K. Mørken, and L. L. Schumaker, editors, *Mathematical Methods for Curves and Surfaces*, pages 194–208. Springer, 2010.

[17] W. Fulton. *Algebraic Curves. An Introduction to Algebraic Geometry.* Addison Wesley, 1969.

[18] X.-S. Gao and S.-C. Chou. Implicitization of rational parametric equations. *J. Symb. Comput.*, 14(5):459 – 470, 1992.

[19] J. Gutierrez, R. Rubio, and D. Sevilla. On multivariate rational function decomposition. *J. Symb. Comput.*, 33(5):545 – 562, 2002.

[20] J. Gutierrez, R. Rubio, and J.-T. Yu. D-resultant for rational functions. *Proc. American Mathematical Society*, 130, 08 2002.

[21] X. Jia, X. Shi, and F. Chen. Survey on the theory and applications of μ-bases for rational curves and surfaces. *J. Comput. Appl. Math.*, 329:2–23, 2018.

[22] A. Kobel and M. Sagraloff. On the complexity of computing with planar algebraic curves. *J. Complexity*, 31, 08 2014.

[23] Y.-M. Li and R. J. Cripps. Identification of inflection points and cusps on rational curves. *CAGD*, 14(5):491 – 497, 1997.

[24] T. Lickteig and M.-F. Roy. Sylvester–Habicht sequences and fast Cauchy index computation. *J. Symb. Comput.*, 31(3):315–341, Mar. 2001.

[25] K. Mahler. On some inequalities for polynomials in several variables. *J. London Mathematical Society*, 1(1):341–344, 1962.

[26] D. Manocha and J. F. Canny. Detecting cusps and inflection points in curves. *CAGD*, 9(1):1 – 24, 1992.

[27] V. Pan. Univariate polynomials: Nearly optimal algorithms for numerical factorization and rootfinding. *J. Symb. Comput.*, 33(5):701–733, 2002.

[28] V. Pan and E. Tsigaridas. Accelerated approximation of the complex roots and factors of a univariate polynomial. *Theor. Computer Science*, 681:138 – 145, 2017.

[29] H. Park. Effective computation of singularities of parametric affine curves. *J. Pure and Applied Algebra*, 173:49–58, 08 2002.

[30] S. Pérez-Díaz. On the problem of proper reparametrization for rational curves and surfaces. *CAGD*, 23(4):307–323, 2006.

[31] S. Pérez-Díaz. Computation of the singularities of parametric plane curves. *J. Symb. Comput.*, 42(8):835 – 857, 2007.

[32] C. A. T. Recio. Plotting missing points and branches of real parametric curves. *Applicable Algebra in Engineering, Communication and Computing*, 18, 02 2007.

[33] R. Rubio, J. Serradilla, and M. Vélez. Detecting real singularities of a space curve from a real rational parametrization. *J. Symb. Comput.*, 44(5):490 – 498, 2009.

[34] A. Schönhage. Probabilistic computation of integer polynomial gcds. *J. Algorithms*, 9(3):365 – 371, 1988.

[35] T. W. Sederberg. Improperly parametrized rational curves. *CAGD*, 3(1):67–75, May 1986.

[36] T. W. Sederberg and F. Chen. Implicitization using moving curves and surfaces. In *Proc. of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, pages 301–308, NY, USA, 1995.

[37] J. R. Sendra. Normal parametrizations of algebraic plane curves. *J. Symb. Comput.*, 33:863–885, 2002.

[38] J. R. Sendra and F. Winkler. Algorithms for rational real algebraic curves. *Fundam. Inf.*, 39(1,2):211–228, Apr. 1999.

[39] J. R. Sendra, F. Winkler, and S. Pérez-Díaz. Rational algebraic curves. *Algorithms and Computation in Mathematics*, 22, 2008.

[40] A. Strzebonski and E. Tsigaridas. Univariate real root isolation in an extension field and applications. *J. Symb. Comput.*, 92:31 – 51, 2019.

[41] J. von zur Gathen and J. Gerhard. *Modern computer algebra.* Cambridge University Press, 3rd edition, 2013.

[42] R. J. Walker. *Algebraic curves.* Springer-Verlag, 1978.