



HAL
open science

Computing the topology of a plane or space hyperelliptic curve

Juan Gerardo Alcázar, Jorge Caravantes, Gema M Diaz-Toca, Elias Tsigaridas

► **To cite this version:**

Juan Gerardo Alcázar, Jorge Caravantes, Gema M Diaz-Toca, Elias Tsigaridas. Computing the topology of a plane or space hyperelliptic curve. Computer Aided Geometric Design, 2020, 10.1016/j.cagd.2020.101830 . hal-01968776v2

HAL Id: hal-01968776

<https://inria.hal.science/hal-01968776v2>

Submitted on 27 Feb 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.

Computing the topology of a plane or space hyperelliptic curve.

Juan Gerardo Alcázar^{a,1,2}, Jorge Caravantes^{a,1}, Gema M. Diaz-Toca^{b,1},
Elias Tsigaridas^{c,3}

^a*Departamento de Física y Matemáticas, Universidad de Alcalá, E-28871 Madrid, Spain*

^b*Departamento de Ingeniería y Tecnología de Computadores, Universidad de Murcia,
E-30100 Murcia, Spain*

^c*Inria Paris and Institut de Mathématiques de Jussieu - Paris Rive Gauche, Sorbonne
Université and Paris Université, France*

Abstract

We present algorithms to compute the topology of 2D and 3D hyperelliptic curves. The algorithms are based on the fact that 2D and 3D hyperelliptic curves can be seen as the image of a planar curve (the Weierstrass form of the curve), whose topology is easy to compute, under a birational mapping of the plane or the space. We report on a **Maple** implementation of these algorithms, and present several examples. Complexity and certification issues are also discussed.

1. Introduction

- 2 Rational curves are widely used in Computer Aided Geometric Design.
3 *Hyperelliptic curves* are not rational, but they are *birationally equivalent* to

Email addresses: juange.alcazar@uah.es (Juan Gerardo Alcázar),
jorge.caravantes@uah.es (Jorge Caravantes), gemadiaz@um.es (Gema M. Diaz-Toca),
elias.tsigaridas@inria.fr (Elias Tsigaridas)

¹Supported by the Spanish Ministerio de Ciencia, Innovación y Universidades and by the European Regional Development Fund (ERDF), under the project MTM2017-88796-P.

²Member of the Research Group ASYNACS (Ref. CCEE2011/R34)

³Partially supported by a Giner de los Ríos Grant of the Universidad de Alcalá, ANR JCJC GALOP (ANR-17-CE40-0009), a public grant as part of the Investissement d'avenir project reference ANR-11-LABX-0056-LMH LabEx LMH (PGMO ALMA), the PHC GRAPE, and from the 2232 International Fellowship for Outstanding Researchers Program from TÜBITAK (Project No: 118C240).

4 planar algebraic curves quadratic in one variable, the corresponding *Weier-*
 5 *strass forms*, where birationally equivalent means that there exists a rational
 6 mapping between the curve and its Weierstrass form with an also rational
 7 inverse. Since Weierstrass forms are quadratic in one variable, hyperelliptic
 8 curves are parametrizable by square-roots. Thus, hyperelliptic curves are one
 9 of the simplest examples of non-rational families of curves. Furthermore, this
 10 type of curves appears frequently in Computer Aided Geometric Design. A
 11 good account of the occurrence of hyperelliptic curves in this field is given in
 12 [8], where the problem of approximating hyperelliptic curves by means of rational
 13 parametrizations is addressed. As a brief summary of [8], non-rational
 14 offsets of rational planar curves and some bisector curves (line/rational curve,
 15 or circle/rational curve) are planar hyperelliptic curves. Contour curves of
 16 canal surfaces, intersections of two quadrics or intersections of a quadric and
 17 a ruled surface are examples of hyperelliptic curves in 3-space. With more
 18 generality, every planar or space algebraic curve \mathcal{C} admitting a square-root
 19 parametrization (see also [27]) is hyperelliptic.

20 In this paper we address the problem of computing the topology of a hyper-
 21 elliptic curve \mathcal{C} . Efficient and fast algorithms to compute the Weierstrass
 22 form \mathcal{G} of \mathcal{C} , as well as a birational mapping $\mathbf{x} : \mathcal{G} \dashrightarrow \mathcal{C}$ can be found in
 23 many computer algebra systems, e.g. Sage, Maple or Magma. Here we will
 24 assume that \mathbf{x}, \mathcal{G} are already known, and in fact that \mathcal{C} is defined by means
 25 of the pair \mathbf{x}, \mathcal{G} , so that \mathcal{C} is seen as the image of the planar algebraic curve \mathcal{G}
 26 under the mapping defined by \mathbf{x} . Since \mathcal{G} is a simple curve, quadratic in one
 27 variable, and therefore the union of the graphs of two univariate functions,
 28 the topology of \mathcal{G} is very easy to capture. Thus, our strategy to compute
 29 the topology of \mathcal{C} is to study how the birational mapping modifies the topol-
 30 ogy of the Weierstrass form. Hence, we might say that the Weierstrass form
 31 “guides” us to build the topology of \mathcal{C} . In more detail, we describe the topol-
 32 ogy of \mathcal{G} by means of a *topological graph* $G_{\mathcal{G}}$, i.e. a graph isotopic to the
 33 curve. Then the topology of \mathcal{C} is described by means of another graph $G_{\mathcal{C}}$
 34 whose vertices are the images of the vertices of $G_{\mathcal{G}}$ under \mathbf{x} , and whose edges
 35 correspond to the branches of $\mathbf{x}(\mathcal{G})$, which are in one-to-one correspondence
 36 with the edges of $G_{\mathcal{G}}$. If \mathbf{x} becomes infinite at a vertex of $G_{\mathcal{G}}$, the image of
 37 such a vertex corresponds to a branch at infinity of \mathcal{C} .

38 Additionally, the pair \mathbf{x}, \mathcal{G} may come for free, or almost for free, in certain
 39 applications; see for instance the introductory example of an intersection
 40 curve at the beginning of Section 2. If the pair \mathbf{x}, \mathcal{G} is known, in order to
 41 determine the topology of \mathcal{C} one might compute an implicit representation

42 of \mathcal{C} using elimination methods. This yields one implicit equation in the
43 plane case, and at least two implicit equations in the space case. In both
44 cases, plane and space, after computing the implicit equation(s) one might
45 use existing algorithms to find the topology of the curve: see for instance
46 [7, 13, 17, 21], among many others, for the planar case, or [5, 12, 14, 18]
47 for the space case. However, such an implicit representation typically has a
48 high degree and big coefficients, which makes it difficult to use. Moreover,
49 many algorithms have additional assumptions, for example generic position,
50 or complete intersection in the space case, that are computationally expensive
51 to fulfill. As a consequence, if the pair \mathbf{x}, \mathcal{G} is known, it is useful to have an
52 alternative method for computing the topology of \mathcal{C} that avoids using an
53 implicit representation.

54 On the other hand, if \mathcal{C} is defined by means of an implicit representation
55 the pair \mathbf{x}, \mathcal{G} can be computed using a computer algebra system. Thus, our
56 algorithm is applicable to that case as well, and provides an alternative to
57 existing algorithms for computing the topology of a plane or space curve.
58 This is specially useful in the space case, since known algorithms to compute
59 the topology of a space case are not so easy to use in practice, and have a
60 high complexity (see Section 6.3).

61 It is worth comparing our paper with some other related papers. In [4]
62 the topology of 2D and 3D rational curves is addressed. In [4] the curve is
63 seen as the image of the real line under a planar or space birational mapping,
64 so somehow the germ of the idea in this paper is already in [4]. In [11], a
65 method to compute the topology of a (non-necessarily rational) offset curve of
66 a rational planar curve is provided. The method exploits similar ideas to [4],
67 but focuses on offset curves, which have special properties. Finally, in [8] the
68 problem of approximating a hyperelliptic curve by means of rational curves is
69 considered. The Weierstrass form is also used in [8], but the goal is different,
70 and in particular the computation of the topology of the hyperelliptic curve
71 is not addressed.

72 Our method has been implemented in the computer algebra system `Maple`
73 2017, and the implementation can be freely downloaded from [29]. In order
74 to certify the topology we need to certify self-intersections, i.e., we need to
75 certify whether or not the image of two points under the birational mapping
76 giving rise to our curve, is the same. This requires to work with algebraic
77 numbers, and is computationally difficult. We address this problem, and we
78 provide a complexity analysis of the algorithm with and without the certifi-
79 cation step. While the complexity bound that we get is not better than the

80 known complexity for the implicit planar case [24], it is, however, definitely
 81 better compared to the implicit space case [15, 12]. It is true, however, that
 82 in [15, 12] the space curve is assumed to be given by an implicit represen-
 83 tation. However, in our paper, even though the algorithm is applicable also
 84 to implicit curves after computing a Weierstrass form of the curve (which is
 85 efficient and fast), we assume a different representation of the curve, namely
 86 as the birational image of a Weierstrass curve.

87 The structure of this paper is the following. We motivate and present the
 88 problem in Section 2, where some preliminary notions and ideas are given.
 89 The planar case is addressed in Section 3, and the space case is studied in
 90 Section 4. In Section 5 we report on the results of our experimentation, car-
 91 ried out in the computer algebra system `Maple 2017`; we refer the interested
 92 reader to the ArXiv version of the paper [3] for the parametrizations used in
 93 the experimentation section. In Section 6, we address the complexity of the
 94 algorithm, we consider certification issues, and we compare the complexity
 95 of our algorithm with the known complexities of algorithms using an implicit
 96 representation of the curve. Section 7 contains our conclusions. The proofs
 97 of some results in Section 3 are postponed to Appendix I, so as not to stop
 98 the flow of the paper.

99 **Acknowledgments:** the authors want to thank the reviewers of the paper
 100 for their comments, which helped improve the quality of the paper.

101 2. Motivation and presentation of the problem.

102 Consider a *biquadratic* patch S , commonly used in Computer Aided Ge-
 103 ometric Design, parametrized by

$$\mathbf{x}(t, s) = (x(t, s), y(t, s), z(t, s)) = \sum_{i=0}^2 \sum_{j=0}^2 \mathbf{c}_{ij} B_i(t) B_j(s), \quad (1)$$

104 where $B_k(u) = \binom{2}{k} u^k (1-u)^{2-k}$ for $k = 0, 1, 2$, and $\mathbf{c}_{ij} \in \mathbb{R}^3$ for $i, j = 0, 1, 2$.
 105 Assume that we want to describe the topology of the intersection curve \mathcal{C}
 106 of S with a general plane Π of equation $Ax + By + Cz + D = 0$, i.e. the
 107 topology of $S \cap \Pi$. In order to do this, substituting the components $x(t, s)$,
 108 $y(t, s)$, $z(t, s)$ of $\mathbf{x}(t, s)$ into the equation of Π we get an algebraic condition
 109 $g(t, s) = 0$; since the components of $\mathbf{x}(t, s)$ have bidegree $(2, 2)$, one can see
 110 that

$$g(t, s) = \Psi_1(t)s^2 + \Psi_2(t)s + \Psi_3(t) = 0, \quad (2)$$

111 where the $\Psi_i(t)$, $i = 1, 2, 3$, are polynomials in the variable t . Then the curve
 112 $\mathcal{C} = S \cap \Pi$ can be described as the closure of the image of the planar curve
 113 \mathcal{G} , defined by $g(t, s) = 0$ in the (t, s) -plane, under the (rational) mapping
 114 \mathbf{x} , i.e. $\mathcal{C} = \mathbf{x}(\mathcal{G})$. Notice that $\mathcal{C} - \mathbf{x}(\mathcal{G})$ reduces to finitely many points
 115 corresponding to either the image of points of \mathcal{G} at infinity, or limit points in
 116 \mathcal{C} corresponding to base points of \mathbf{x} , lying in \mathcal{G} .

117 The situation presented above is an example of the general problem
 118 treated in this paper. Given a planar curve \mathcal{G} , implicitly defined in the plane
 119 (t, s) by a polynomial equation like Eq. (2), of degree 2 in the variable s , our
 120 goal is to compute the topology of the curve $\mathcal{C} = \mathbf{x}(\mathcal{G})$, where $\mathbf{x} : \mathbb{R}^2 \rightarrow \mathbb{R}^n$,
 121 with $n = 2$ or $n = 3$, is *birational* when restricted to \mathcal{G} ; in particular, in that
 122 case the inverse mapping $\mathbf{x}|_{\mathcal{G}}^{-1} : \mathcal{C} \rightarrow \mathcal{G}$ exists and is rational. Writing

$$\mathbf{x} = (x_1, x_2, \dots, x_n),$$

123 we will refer to the functions $x_i : \mathbb{R}^2 \rightarrow \mathbb{R}$ as the *components* of the mapping
 124 \mathbf{x} . Notice that if \mathcal{C} is a rational curve, in which case the curve \mathcal{G} must also be
 125 rational because of the birationality of the mapping $\mathbf{x}|_{\mathcal{G}}$, then the problem
 126 can be solved using already existing methods [4]. Thus, we will assume that
 127 \mathcal{C} , and therefore also \mathcal{G} , is not rational, in which case \mathcal{C} is said to be a
 128 *hyperelliptic curve*.

129 With some generality (see for instance [8]), we say that a curve \mathcal{C} is *hy-*
 130 *perelliptic* if there exists a generically two-to-one map $\mathcal{C} \rightarrow \mathbb{R}$. Furthermore,
 131 such a curve (see for instance [26]) is birationally equivalent to a planar curve

$$s^2 - p(t) = 0, \tag{3}$$

132 where $p(t)$ is a square-free polynomial of degree $2\mathbf{g} + 1$ or $2\mathbf{g} + 2$, where \mathbf{g} is
 133 the *genus* of \mathcal{C} . Recall (see for instance [28]) that the genus \mathbf{g} is a birational
 134 invariant that, in particular, characterizes rational curves: $\mathbf{g} = 0$ corresponds
 135 to rational curves, while for non-rational curves $\mathbf{g} \geq 1$, $\mathbf{g} \in \mathbb{N}$. Additionally,
 136 whenever we work over a field of characteristic different from 2, as it is our
 137 case, one can always get a Weierstrass curve where the degree of $p(t)$ is
 138 $2\mathbf{g} + 1$ (see for instance [26]). Also, Eq. (3) is called the *Weierstrass form* of
 139 \mathcal{C} . Notice (see p. 59 of [8]) that we can always transform the expression Eq.
 140 (2) of our motivating example into an expression like Eq. (3) by considering
 141 a change of parameters

$$t := t, \quad s := \frac{-B(t) + s}{2A(t)}.$$

142 In this paper we will assume that the Weierstrass form has already been
 143 computed, and therefore that the curve \mathcal{G} is described by means of Eq. (3).
 144 Additionally, we will assume that the curve \mathcal{G} is real, i.e. that it contains
 145 infinitely many real points; if \mathcal{G} is not real, then because of the birationality
 146 of $\mathbf{x}|_{\mathcal{G}}$, \mathcal{C} cannot be real either. Observe also that since $s^2 - p(t)$ is an irre-
 147 reducible polynomial in t, s , so is the curve \mathcal{G} ; since irreducibility is a birational
 148 invariant, we deduce that \mathcal{C} is irreducible as well.

149 In order to describe the topology of the curve \mathcal{C} , we will compute, as it
 150 is common, a graph *isotopic* to \mathcal{C} .

151 **Definition 1.** Let $X, Y \subset \mathbb{R}^n$. We say that X, Y are isotopic if there exists
 152 a continuous map $H : X \times [0, 1] \rightarrow \mathbb{R}^n$ satisfying the following conditions:
 153 (1) $H(\bullet; 0)$ is the identity; (2) $H(X; 1) = Y$; (3) for all $\omega \in [0, 1]$, $H(\bullet; \omega)$ is
 154 a homeomorphism from X to $H(X; \omega)$. In this case, H is called an isotopy
 155 between X, Y .

156 If X, Y in Definition 1 are 1-dimensional objects, the fact that X, Y are
 157 isotopic implies that one of them can be deformed into the other without
 158 removing or introducing self-intersections (see for instance [22]). Now we
 159 have the following definition.

160 **Definition 2.** Let $\mathcal{C} \subset \mathbb{R}^n$, where $n = 2$ or $n = 3$. A topological graph of \mathcal{C}
 161 is a graph $G_{\mathcal{C}}$ isotopic to \mathcal{C} whose vertices lie on the curve \mathcal{C} .

162 **Remark 1.** Vertices of $G_{\mathcal{C}}$ with valence equal to one, i.e. belonging only to
 163 one edge, correspond to real branches of \mathcal{C} at infinity. Thus, if $G_{\mathcal{C}}$ contains
 164 some vertex of this type, then \mathcal{C} is not bounded.

165 Thus, our goal is to build an algorithm for computing a topological graph
 166 $G_{\mathcal{C}}$ of \mathcal{C} ; we will refer to $G_{\mathcal{C}}$ as the graph *associated with* \mathcal{C} . In order to
 167 do this, we will not compute $G_{\mathcal{C}}$ directly: instead, we will compute a graph
 168 $G_{\mathcal{G}}$ associated with \mathcal{G} , and we will derive $G_{\mathcal{C}}$ from $G_{\mathcal{G}}$ by studying how the
 169 topology of \mathcal{G} changes when \mathbf{x} is applied. Furthermore, in our analysis we
 170 do not consider isolated real points of \mathcal{C} , which can be generated by complex
 171 branches of \mathcal{G} at infinity. Let us briefly recall how graphs associated with
 172 planar and space curves are computed.

173 GRAPH ASSOCIATED WITH A PLANAR CURVE.

174 Let $f(x, y) = 0$ define a planar algebraic curve \mathcal{F} without vertical asymp-
 175 totes. We say that $P \in \mathcal{F}$ is *regular* if either $f_x(P) \neq 0$ or $f_y(P) \neq 0$;

176 otherwise, we say that P is *singular*. We say that $P \in \mathcal{F}$ is *critical* if P
 177 satisfies that $f(P) = f_y(P) = 0$. A critical point which is not singular is
 178 called a *ramification* point. The topological graph G_f associated with \mathcal{F} can
 179 be described as follows (see Fig. 1, left):

- 180 • The **vertices** of the graph G_f are: (1) the critical points of \mathcal{F} ; (2) the
 181 points of \mathcal{F} lying on the vertical lines through the critical points of \mathcal{F}
 182 (we call these vertical lines, *critical lines*); (3) the points of \mathcal{F} lying on
 183 vertical lines placed: (3.1) between two consecutive critical lines, (3.2)
 184 at the left of the left-most critical point, and (3.3) at the right of the
 185 right-most critical point.
- 186 • Two vertices of G_f are connected by an **edge** of G_f iff there is a real
 187 branch of \mathcal{F} connecting the corresponding points on \mathcal{F} .

188 The problem of computing a topological graph of an implicit planar curve
 189 is well-studied. The interested reader can check the references [7, 13, 17,
 190 21], among others, for further information on the problem. Although it is
 191 customary, in most papers dealing with the problem of computing the graph
 192 G_f , to start with the assumption that \mathcal{F} does not have vertical asymptotes
 193 or vertical components, one can adapt the strategy without assuming these
 194 properties; see for instance [6].

195 GRAPH ASSOCIATED WITH A SPACE CURVE.

196 Let $\{f_1(x, y, z) = 0, \dots, f_m(x, y, z) = 0\}$ define a space algebraic curve
 197 \mathcal{F} : (i) without asymptotes parallel to the z -axis; (ii) such that the projection
 198 $\pi_{xy}(\mathcal{F})$ of \mathcal{F} onto the xy -plane is birational. Hypothesis (iii) ensures that
 199 there are not two different real branches of \mathcal{F} projecting onto a same branch of
 200 $\pi_{xy}(\mathcal{F})$. Taking advantage of Hypothesis (ii), the usual strategy to compute
 201 a topological graph G_f isotopic to \mathcal{F} is to birationally project \mathcal{F} onto some
 202 plane, say, the xy -plane, then compute a graph isotopic to the projection
 203 $\pi_{xy}(\mathcal{F})$, which is a planar algebraic curve, and later “lift” the graph associated
 204 with $\pi_{xy}(\mathcal{C})$ to a space graph: this is the strategy followed in papers like
 205 [12, 14, 18], and we will follow this strategy here as well. Since the projection
 206 π_{xy} is birational, one can be sure that every edge of the graph associated with
 207 $\pi_{xy}(\mathcal{F})$ lifts to one, and just one, edge of the graph associated with \mathcal{F} . More
 208 precisely, the graph G_f associated with \mathcal{F} can be described as follows (see
 209 Fig. 1, right):

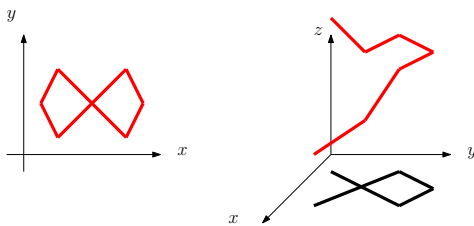


Figure 1: Graphs associated with planar and space curves

- 210 • The **vertices** of the graph G_f are the points of \mathcal{F} projecting as vertices
 211 of the graph associated with $\pi_{xy}(\mathcal{F})$.
- 212 • Two vertices of G_f are connected by an **edge** of G_f iff the correspond-
 213 ing points of \mathcal{C} are connected by a real branch of \mathcal{C} . Furthermore, if the
 214 vertices are not singularities of $\pi_{xy}(\mathcal{C})$, we connect them iff their projec-
 215 tions are connected in the graph associated with $\pi_{xy}(\mathcal{F})$. For vertices
 216 corresponding to singularities of $\pi_{xy}(\mathcal{C})$ the process is more complicated,
 217 since we can have two non-overlapping branches of \mathcal{C} whose projections
 218 onto the xy -plane overlap (see Fig. 1, left); for references on how to
 219 deal with this problem, one can check [14, 18].

220 The problem of computing a topological graph associated with an implicit
 221 space algebraic curve has received some attention in the literature, although
 222 less than the planar case. The interested reader can check the references
 223 [5, 12, 14, 18] for more details on the problem. Again, as it also happens in
 224 the planar case, the strategy can be adapted to the case when \mathcal{F} has vertical
 225 components or vertical asymptotes.

226 IN OUR CASE.

227 In our case, we need to compute the graph $G_{\mathcal{G}}$ associated with \mathcal{G} plus
 228 some extra vertices $Q_i = (t_i, s_i) \in \mathcal{G}$. In particular, we need to include points
 229 $Q_i \in \mathcal{G}$ giving rise to certain notable points $P_i \in \mathcal{C}$, as we will see in the
 230 next sections. And we also need to include the points $Q_i \in \mathcal{G}$ where some
 231 component of \mathbf{x} has the indeterminacy $\frac{0}{0}$, or becomes infinite. After including
 232 these vertices, we observe that \mathbf{x} is continuous over each portion of the curve
 233 \mathcal{G} corresponding to each edge of $G_{\mathcal{G}}$. Then, the key idea is that since the
 234 image of any connected subset of \mathcal{G} is also connected, every edge e of $G_{\mathcal{G}}$
 235 gives rise to an edge \tilde{e} of $G_{\mathcal{C}}$, namely the edge connecting the images of the

236 vertices of e . Hence, the topology of \mathcal{G} guides us to compute the topology of
 237 \mathcal{C} .

238 The fact that \mathbf{x} is birational over \mathcal{G} guarantees that all the edges of $G_{\mathcal{C}}$
 239 are obtained this way, since there cannot be any real branch of \mathcal{C} coming
 240 from a complex branch of \mathcal{G} : indeed, if $\mathcal{B} \subset \mathcal{G}$ is a complex branch such that
 241 $\mathbf{x}(\mathcal{B})$ is real, then $\mathbf{x}(\mathcal{B}) = \overline{\mathbf{x}(\mathcal{B})}$, where $\overline{\mathbf{x}(\mathcal{B})}$ denotes the conjugate of $\mathbf{x}(\mathcal{B})$.
 242 But then there are infinitely many points of \mathcal{C} with at least two pre-images,
 243 which cannot happen because $\mathbf{x}|_{\mathcal{G}}$ is birational.

244 Therefore, the rough idea in order to build $G_{\mathcal{C}}$ is to compute the graph
 245 $G_{\mathcal{G}}$ (by using any of the well-known algorithms to do this), and the images
 246 P_i of the vertices V_i of $G_{\mathcal{G}}$. Then we connect the P_i according to how their
 247 preimages $V_i = \mathbf{x}|_{\mathcal{G}}^{-1}(P_i)$ are connected in \mathcal{G} . If some component of $\mathbf{x}(V_i)$
 248 becomes infinite, then we have an open branch of \mathcal{C} , i.e. a branch of \mathcal{C} going
 249 to infinity; in particular, in that case \mathcal{C} is not bounded.

250 Fig. 2 represents the idea of computing $G_{\mathcal{C}}$ from $G_{\mathcal{G}}$, for the case $n = 2$:
 251 each edge, marked with a different color, of the graph $G_{\mathcal{G}}$ (left), gives rise to
 252 an edge, marked with the same color, of the graph $G_{\mathcal{C}}$ (right).

253 Observe that since \mathcal{G} is implicitly defined by Eq. (3), the leading coef-
 254 ficient in the variable s is constant, so \mathcal{G} has no asymptotes parallel to the
 255 s -axis, which we take as the vertical axis in the (t, s) plane. Additionally,
 256 since the Weierstrass form implies that $p(t)$ is square-free, one can see that \mathcal{G}
 257 is regular, and that the only critical points are the points $\{s = 0, p(t) = 0\}$,
 258 all of which are ramification points, i.e. points where the tangent line to
 259 \mathcal{G} is vertical. Because of this, \mathcal{G} consists of open branches and/or closed
 260 components, without self-intersections. As a projective variety, though, \mathcal{G} has
 261 a singular point, namely the point at infinity of \mathcal{G} (in the direction of the
 262 s -axis).

263 Certainly, there can also be some points of \mathcal{C} which do not belong to $\mathbf{x}(\mathcal{G})$.
 264 The points in $\mathcal{C} - \mathbf{x}(\mathcal{G})$ correspond to the images of the point at infinity of
 265 \mathcal{G} , and the limit points coming from the base points of \mathbf{x} lying in \mathcal{G} , i.e.
 266 points of \mathcal{G} where all the numerators and denominators of the components
 267 of \mathbf{x} vanish simultaneously. Since \mathcal{G} is regular over its affine part, we can
 268 be sure that \mathbf{x} extends to its base points (see Theorem 1.2 of [23]), so that
 269 base points give rise to either affine points of \mathcal{C} , or points at infinity of \mathcal{C} .
 270 The effective computation of the images of base points of \mathbf{x} on \mathcal{G} is analyzed
 271 in the next section. On the other hand, \mathcal{G} has one singular point at infinity
 272 with two different branches, i.e. two different *places* centered at this point
 273 (see [31] for further information on places). This implies that the point at

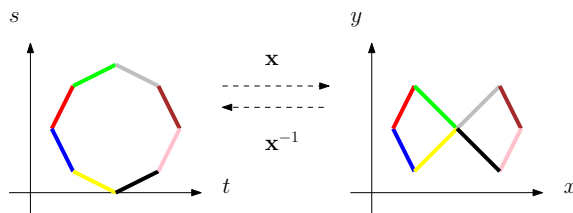


Figure 2: $G_{\mathcal{G}}$ and $G_{\mathcal{C}}$

274 infinity of \mathcal{G} can give rise to at most two points of \mathcal{C} , that can be affine, or at
 275 infinity. We denote these points by $P_{\infty}, P_{-\infty}$, that may or may not coincide.
 276 This notation responds to the fact that these points are reached by analyzing
 277 the behavior of $\mathbf{x}|_{\mathcal{G}}$ when $t \rightarrow \infty$ and $t \rightarrow -\infty$. In the next section, we will
 278 consider the computation of these points, that we will represent in a more
 279 compact way by $P_{\pm\infty}$.

280 3. The planar case.

281 Let $\mathbf{x} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$, where

$$\mathbf{x}(t, s) = (x(t, s), y(t, s)) = \left(\frac{A_1(t, s)}{B_1(t, s)}, \frac{A_2(t, s)}{B_2(t, s)} \right),$$

282 and let $\mathcal{C} = \mathbf{x}(\mathcal{G})$, where \mathcal{G} is implicitly defined by an equation $g(t, s) =$
 283 $s^2 - p(t) = 0$ like Eq. (3). The functions $x(t, s), y(t, s)$ are the *components* of
 284 $\mathbf{x}(t, s)$. We require \mathbf{x} to be a rational mapping satisfying that the restriction
 285 $\mathbf{x}|_{\mathcal{G}}$ is birational, so that $\mathbf{x}|_{\mathcal{G}}^{-1} : \mathcal{C} \rightarrow \mathcal{G}$ is well-defined, and therefore rational.
 286 We can always check this assumption with a probabilistic algorithm; we take
 287 a random point $(t_0, s_0) \in \mathcal{G}$, compute the point $P = \mathbf{x}(t_0, s_0)$, and finally
 288 determine the preimages of $\mathbf{x}(t_0, s_0)$: if we get only one preimage belonging
 289 to \mathcal{G} , then with probability one the required hypothesis holds. Additionally,
 290 using repeatedly the fact that $s^2 = p(t)$, we can write $\mathbf{x}|_{\mathcal{G}}(t, s)$ in the following
 291 form:

$$\mathbf{x}|_{\mathcal{G}}(t, s) = \left(\frac{A_1(t, s)}{B_1(t, s)}, \frac{A_2(t, s)}{B_2(t, s)} \right) = \left(\frac{a_{11}(t) + sa_{12}(t)}{b_{11}(t) + sb_{12}(t)}, \frac{a_{21}(t) + sa_{22}(t)}{b_{21}(t) + sb_{22}(t)} \right), \quad (4)$$

292 where we can assume that A_i, B_i are relatively prime for $i = 1, 2$. Observe
 293 that this implies $\gcd(a_{11}, a_{12}, b_{11}, b_{12}) = 1$ and $\gcd(a_{21}, a_{22}, b_{21}, b_{22}) = 1$. No-
 294 tice also that in general $b_{11}(t) \neq b_{21}(t)$, $b_{12}(t) \neq b_{22}(t)$.

295 As observed in Section 2, we first need to describe the topology of \mathcal{G} by
 296 means of a graph $G_{\mathcal{G}}$ isotopic to it, with some additional vertices. We need
 297 to include the following points as vertices of $G_{\mathcal{G}}$:

- 298 (i) *Critical points of $g(t, s) = 0$, i.e. points of \mathcal{G} where $g_s = 0$.*
- 299 (ii) *Points of \mathcal{G} giving rise to critical points of \mathcal{C} .*
- 300 (iii) *Points of \mathcal{G} where some component of \mathbf{x} is not defined.*
- 301 (iv) *Starting and ending points for open branches of \mathcal{G} .*

302 The points in (i) are the solutions of $g = g_s = 0$, i.e. the points $\{s =$
 303 $0, p(t) = 0\}$. The points in (iv) can be easily computed by taking a t -value at
 304 the left (resp. right) of the left-most (resp. the right-most) solution of $g = g_s$.
 305 The points in (iii) are the points $(t, s) \in \mathcal{G}$ such that $B_1(t, s) \cdot B_2(t, s) = 0$.
 306 In particular, some of the points in (iii) may generate asymptotes of \mathcal{C} ; also,
 307 *base points* of \mathbf{x} in \mathcal{G} , i.e. the points of \mathcal{G} where

$$A_1(t, s) = B_1(t, s) = A_2(t, s) = B_2(t, s) = g(t, s) = 0,$$

308 are included in (iii). The topology of \mathcal{G} is easy to capture (see for instance
 309 [8]), and can be computed by using known algorithms for planar curves like
 310 [7, 13, 17, 21].

311 3.1. Computing the points of \mathcal{G} giving rise to critical points of \mathcal{C}

312 For simplicity, in this section we will assume that \mathbf{x} has no base points on
 313 \mathcal{G} . These points, which may also generate critical points of \mathcal{C} , will be analyzed
 314 in the next subsection. Some observations on how to use the results in this
 315 subsection in the presence of base points will be done at the end of the
 316 subsection. Additionally, if the points $P_{\pm\infty}$ are affine they may be critical
 317 points of \mathcal{C} as well. The behavior of $P_{\pm\infty}$ will be studied in Subsection 3.3.

318 Now in Section 2 we recalled that the critical points of \mathcal{C} are either singu-
 319 larities, or ramification points, i.e. points where the tangent line is vertical.
 320 It is useful to distinguish two types of singularities: *local singularities*, which
 321 correspond to singular points $P \in \mathcal{C}$ with just one branch of \mathcal{C} through P ,
 322 and *self-intersections* of \mathcal{C} , which correspond to points $P \in \mathcal{C}$ with at least
 323 two different branches of \mathcal{C} through P . In Fig. 3 we show three examples of
 324 local singularities, two of them cuspidal (first two curves, starting from the

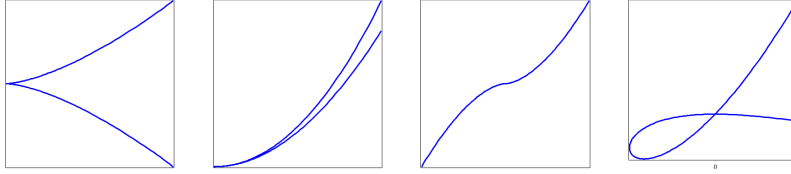


Figure 3: Local singularities (three local singularities and q self-intersection (right-most curve)).

325 left) and one of them non-cuspidal (third curve, starting from the left), and
 326 a self-intersection (right-most curve); see [1] for more information on local
 327 singularities.

328 In order to compute the points of \mathcal{G} giving rise to local singularities and
 329 ramification points of \mathcal{C} , we analyze $\mathbf{x}(\mathcal{G})$, where \mathcal{G} is implicitly defined by
 330 $g(t, s) = 0$. The differential of \mathbf{x} defines a mapping between the tangent
 331 space to \mathcal{G} and the tangent space to \mathcal{C} , at corresponding points. Denoting
 332 a generic element of the tangent space to \mathcal{C} by $\mathbf{v} = (v_1, v_2)$, we have the
 333 following relationship; here, x_t represents the partial derivative of $x(t, s)$
 334 with respect to the variable t , and similarly for y_t, x_s, y_s, g_t, g_s :

$$\begin{bmatrix} x_t & x_s \\ y_t & y_s \end{bmatrix} \cdot \begin{bmatrix} g_s \\ -g_t \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \quad (5)$$

335 The above relationship follows from differentiating with respect to t the
 336 components of $\mathbf{x}|_{\mathcal{G}}$. Whenever $g_s \neq 0$ (i.e. whenever (t, s) is not a ramification
 337 point of \mathcal{G}), $g(t, s) = 0$ implicitly defines a differentiable function $s = s(t)$,
 338 where $\frac{ds}{dt} = -\frac{g_t}{g_s}$. Now differentiating $\mathbf{x}(t, s) = 0$ where $s = s(t)$ is the
 339 function defined by $g(t, s) = 0$, and using the Chain Rule, we get a vector \mathbf{w}
 340 which is parallel to the vector \mathbf{v} in Eq. (5). For the points where $g_s = 0$, we
 341 can proceed in the same way, reaching the same result, differentiating with
 342 respect to s instead. Since all affine points of \mathcal{G} are regular, i.e. either g_t or
 343 g_s are nonzero, Eq. (5) holds.

344 **Lemma 3.** *Suppose that \mathbf{x} has no base points lying on \mathcal{G} , and let $P \in \mathcal{C}$,
 345 $P \neq P_{\pm\infty}$, $P = \mathbf{x}(t_0, s_0)$, where $(t_0, s_0) \in \mathcal{G}$. If P is either a local singularity
 346 or a ramification point of \mathcal{C} , then (t_0, s_0) satisfies that*

$$g = x_t g_s - x_s g_t = 0. \quad (6)$$

347 **Remark 2.** For the local singularities we have

$$g = x_t g_s - x_s g_t = y_t g_s - y_s g_t = 0. \quad (7)$$

348 However, Lemma 3 does not necessarily provide the self-intersections of
 349 \mathcal{C} . In order to find these last singularities, we imitate the strategy in [2].
 350 First we define

$$\begin{aligned} \xi_1(x, t) &= \text{square-free part of } \text{Res}_s(\text{num}(x - x(t, s)), g(t, s)), \\ \xi_2(x, y, t) &= \text{square-free part of } \text{Res}_s(\text{num}(x - x(t, s)), \text{num}(y - y(t, s))), \end{aligned} \quad (8)$$

351 where $\text{num}(\bullet)$ denotes the numerator of the rational function \bullet . Notice that
 352 in general, eliminating t in $\xi_1(x, t) = 0$, $\xi_2(x, y, t) = 0$ by means of the
 353 resultant $\text{Res}_t(\xi_1(x, t), \xi_2(x, y, t))$, we obtain a polynomial in x, y containing,
 354 as a factor, the implicit equation of \mathcal{C} . Using the definition of the resultant,
 355 one can easily check that $\xi_1(x, t)$ is a quadratic polynomial in x , and $\xi_2(x, y, t)$
 356 is quadratic as a polynomial in x, y , and linear in x and in y (i.e. $\xi_2(x, y, t)$
 357 is bilinear).

358 Now the key idea to find the self-intersections of \mathcal{C} is that these points
 359 are among the points $(x, y) \in \mathcal{C}$ where $t = \mathbf{x}|_{\mathcal{G}}^{-1}(x, y)$ is not defined. For
 360 a generic point $(x_0, y_0) \in \mathcal{C}$, we can find $t_0 = \mathbf{x}|_{\mathcal{G}}^{-1}(x_0, y_0)$ as the *only* root
 361 of $\text{gcd}(\xi_1(x_0, t), \xi_2(x_0, y_0, t))$. In order to find the *function* $t = t(x, y) =$
 362 $\mathbf{x}|_{\mathcal{G}}^{-1}(x, y)$, we can compute the gcd of $\xi_1(x, t)$ and $\xi_2(x, y, t)$ as polynomials
 363 in the variable t whose coefficients are real polynomials in x, y , with the
 364 additional condition $f(x, y) = 0$, where f is the implicit equation of \mathcal{C} . More
 365 formally, one sees $\xi_1(x, t)$ and $\xi_2(x, y, t)$ as elements of $\mathbb{R}(\mathcal{C})[t]$, where $\mathbb{R}(\mathcal{C})$
 366 is the field of real rational functions of \mathcal{C} . Since \mathcal{C} is irreducible $\mathbb{R}(\mathcal{C})$ is a
 367 Euclidean domain. Therefore

$$D(x, y, t) = \underset{\mathbb{R}(\mathcal{C})[t]}{\text{gcd}}(\xi_1, \xi_2)$$

368 is well-defined and can be computed, for instance, by means of the Euclidean
 369 algorithm. Since $\mathbf{x}|_{\mathcal{G}}$ is proper, $D(x, y, t)$ is linear in t and solving $D(x, y, t) =$
 370 0 for t , one gets $t = \mathbf{x}|_{\mathcal{G}}^{-1}(x, y)$.

371 Following the ideas of [2], one can compute $\mathbf{x}|_{\mathcal{G}}^{-1}(x, y)$ more efficiently as
 372 follows (see [2] for further detail). By the fundamental property of subresultants,
 373 $D(x, y, t)$ is the first subresultant different from zero (modulo $f(x, y)$)
 374 in the subresultant chain of ξ_1, ξ_2 , seen as elements of the domain $\mathbb{R}[x, y][t]$.

375 If the degrees of ξ_1, ξ_2 as elements of $\mathbb{R}[x, y][t]$ are n_1, n_2 , the elements of the
 376 subresultant chain are represented as

$$\{\mathbf{Subres}_i(\xi_1, n_1, \xi_2, n_2)_{i \geq 0}\},$$

377 with $0 \leq i \leq \inf(n_1, n_2) - 1$, and can be defined as determinants of order
 378 $n_1 + n_2 - i$ of Sylvester-like matrices whose entries are related to the coeffi-
 379 cients of ξ_1, ξ_2 (see Section 2.2 of [2]). Since $\deg(\mathbf{Subres}_i(\xi_1, n_1, \xi_2, n_2)) \leq i$,
 380 and by the birationality of $\mathbf{x}|_{\mathcal{G}}$ we have $\deg(G(x_0, y_0, t)) = 1$ for almost all
 381 $(x_0, y_0) \in \mathcal{C}$, we deduce that $D(x, y, t)$ is equal to $\mathbf{Subres}_1(\xi_1, n_1, \xi_2, n_2)$; no-
 382 tice that $\mathbf{Subres}_1(\xi_1, n_1, \xi_2, n_2)$ can be computed without actually knowing
 383 the implicit equation of \mathcal{C} . Writing

$$\mathbf{Subres}_1(\xi_1, n_1, \xi_2, n_2)(t) = \mathbf{sres}_1(x, y) t + \mathbf{sr}_1(x, y),$$

384 we have that

$$t = \mathbf{x}|_{\mathcal{G}}^{-1}(x, y) = -\frac{\mathbf{sr}_1(x, y)}{\mathbf{sres}_1(x, y)}. \quad (9)$$

385 The polynomial $\mathbf{sres}_1(x, y)$ is called the first principal subresultant of ξ_1, n_1
 386 and ξ_2, n_2 . Finally we get the following result.

387 **Theorem 4.** *Suppose that \mathbf{x} has no base points lying on \mathcal{G} , and let $P \in \mathcal{C}$,
 388 $P = \mathbf{x}(t_0, s_0)$, $P \neq P_{\pm\infty}$. If P is a self-intersection, then (t_0, s_0) is a solution
 389 of the bivariate polynomial system*

$$\mathbf{sres}_1(x(t, s), y(t, s)) = 0, \quad g(t, s) = 0. \quad (10)$$

390 The next result shows that, in fact, *all* the singularities of \mathcal{C} , i.e. the local
 391 singularities and the self-intersections, except perhaps for $P_{\pm\infty}$, are solutions
 392 of Eq. (10). The proof of this result is given in Appendix I, so as not to stop
 393 the flow of the paper.

394 **Proposition 5.** *Let $(t_0, s_0) \in \mathcal{G}$ be a point such that*

$$(x_0, y_0) = (x(t_0, s_0), y(t_0, s_0)) \in \mathcal{C}$$

395 *is not a self-intersection, with*

$$x_t(t_0, s_0)g_s(t_0, s_0) - x_s(t_0, s_0)g_t(t_0, s_0) = y_t(t_0, s_0)g_s(t_0, s_0) - y_s(t_0, s_0)g_t(t_0, s_0) = 0. \quad (11)$$

396 *Then $\mathbf{sres}_1(x_0, y_0) = 0$.*

397 Proposition 5 provides the following result.

398 **Theorem 6.** *Suppose that \mathbf{x} has no base points lying on \mathcal{G} . Then every*
 399 *singularity of \mathcal{C} , except perhaps for $P_{\pm\infty}$, is a solution of Eq. (10).*

400 The analysis of $P_{\pm\infty}$ is postponed to Section 3.3. Additionally, there is
 401 another point missing in the discussion before. In order for the subresultant
 402 chain of ξ_1, ξ_2 not to vanish completely, we must require that ξ_1, ξ_2 do not
 403 share any factor depending on t . We identify the cases when this happens in
 404 the following two results. The proofs of these results are given in Appendix
 405 I.

406 **Lemma 7.** *The polynomials $\xi_1(x, t)$ and $\xi_2(x, y, t)$ have a common factor*
 407 *$t - t_0$ iff t_0 corresponds to a base point of \mathbf{x} , lying on \mathcal{G} .*

408 **Lemma 8.** *The polynomials $\xi_1(x, t)$ and $\xi_2(x, y, t)$ have a common factor*
 409 *$\eta(x, t)$ depending on both x, t iff $x(t, s)$ depends only on t .*

410 In the case of Lemma 7, if \mathbf{x} has some base point lying on \mathcal{G} we remove
 411 the common factor depending on t , and perform the procedure presented
 412 before. In the case of Lemma 8, we replace $\xi_2(x, y, t)$ by

$$\tilde{\xi}_2(y, t) = \text{square-free part of } \text{Res}_s(\text{num}(y - y(t, s)), g(t, s)),$$

413 and proceed as before.

414 3.2. Behavior of \mathcal{C} around the base points of $\mathbf{x}|_{\mathcal{G}}$.

415 Let $Q = (t_0, s_0) \in \mathcal{G}$ be a base point of $\mathbf{x}|_{\mathcal{G}}$. Notice that by Lemma
 416 7, $t = t_0$ must be a root of the content of ξ_1, ξ_2 with respect to t , and
 417 therefore has been previously determined. In this case, $\mathbf{x}(t_0, s_0) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$.
 418 Although the fact that the \mathcal{G} does not have affine singularities guarantees
 419 that $\mathbf{x}(t_0, s_0)$ is defined as a projective point (see Theorem 1.2 of [23]), we
 420 still need to determine the behavior of \mathbf{x} when the point (t_0, s_0) is approached;
 421 in particular, we need to check if we get an affine point or a point at infinity, in
 422 which case we get an infinite branch of \mathcal{C} . In order to do this, we distinguish
 423 two situations:

- 424 (i) *The point (t_0, s_0) is not a critical point of \mathcal{G} :* in this case, by the Implicit
 425 Function Theorem $s^2 - p(t) = 0$ implicitly defines $s = s(t)$ at $t = t_0$.

426 In fact, we can easily find the Taylor expansion of the function $s(t)$ at
 427 $t = t_0$, and then study the limits

$$\lim_{t \rightarrow t_0} x(t, s(t)), \lim_{t \rightarrow t_0} y(t, s(t)).$$

428 If both limits are finite, then (t_0, s_0) generates an affine point of \mathcal{C} .
 429 Otherwise we have a branch going to infinity, which is an asymptote of
 430 \mathcal{C} whenever one of the above limits is finite.

431 (ii) *The point (t_0, s_0) is a critical point of \mathcal{G} :* in this case t_0 is a root of $p(t)$,
 432 so $s_0 = 0$. Now we consider $s = \pm\sqrt{p(t)}$ and we study each branch
 433 $s = \sqrt{p(t)}$ and $s = -\sqrt{p(t)}$ separately. We address in more detail
 434 the case $s = \sqrt{p(t)}$; for $s = -\sqrt{p(t)}$ the analysis is similar. Now if
 435 $s = \sqrt{p(t)}$, for the component $x(t, s)$ we have

$$x\left(t, \sqrt{p(t)}\right) = \frac{a_{11}(t) + \sqrt{p(t)}a_{12}(t)}{b_{11}(t) + \sqrt{p(t)}b_{12}(t)}.$$

436 We are interested in analyzing the behavior of this function when
 437 $t \rightarrow t_0$. Since $(t_0, 0)$ is a base point of $x(t, s)$, $a_{11}(t_0) = b_{11}(t_0) = 0$.
 438 Additionally, since $a_{11}(t)$, $a_{12}(t)$, $b_{11}(t)$, $b_{12}(t)$ are relatively prime, it
 439 cannot be $a_{12}(t_0) = 0$ and $b_{12}(t_0) = 0$ simultaneously. Furthermore,
 440 $t = t_0$ is a root of $p(t)$, and since $p(t)$ does not have multiple roots,
 441 the multiplicity of t_0 is 1. Hence we can factor out $(t - t_0)^{1/2}$ in the
 442 numerator and denominator of $x(t, \sqrt{p(t)})$, and we get

$$x\left(t, \sqrt{p(t)}\right) = \frac{\tilde{a}_{11}(t) + \sqrt{\tilde{p}(t)}a_{12}(t)}{\tilde{b}_{11}(t) + \sqrt{\tilde{p}(t)}b_{12}(t)},$$

443 where $\tilde{a}_{11}(t) = \frac{a_{11}(t)}{(t - t_0)^{1/2}}$, $\tilde{b}_{11}(t) = \frac{b_{11}(t)}{(t - t_0)^{1/2}}$, and $\tilde{p}(t) = \frac{p(t)}{t - t_0}$.

444 Observe that since $a_{11}(t_0) = b_{11}(t_0) = 0$ and $a_{11}(t), b_{11}(t)$ are polyno-
 445 mials, $\tilde{a}_{11}(t_0) = \tilde{b}_{11}(t_0) = 0$. Therefore, when $t \rightarrow t_0$ the limit of the
 446 function $x(t, \sqrt{p(t)})$ is equal to the limit of $a_{12}(t)/b_{12}(t)$ when $t \rightarrow t_0$.
 447 Since not both $a_{12}(t_0), b_{12}(t_0)$ are zero, the limit is defined whenever
 448 $b_{12}(t_0) \neq 0$, and is infinite (in which case we have a branch at infin-
 449 ity) whenever $b_{12}(t_0) = 0$. Similarly for the component $y(t, s)$, and for
 450 $s = -\sqrt{p(t)}$.

451 Notice that these ideas can be also used at points (t_0, s_0) where only one
 452 component of $\mathbf{x}|_{\mathcal{G}}(t, s)$ is undefined. Observe also that when working in a
 453 projective setting, the point at infinity of the curve \mathcal{G} , $(0 : 1 : 0)$, which gives
 454 rise to $P_{\pm\infty}$, is also a base point of the mapping \mathbf{x} (see Eq. (4)). The analysis
 455 of the behavior of the mapping \mathbf{x} around this point is carried out in the next
 456 subsection.

457 3.3. Computation and study of $P_{\pm\infty}$.

458 The point at infinity of the curve \mathcal{G} is the center of two *places*, i.e. two
 459 branches of \mathcal{G} . In turn, these two branches generate two branches of \mathcal{C} via \mathbf{x} ,
 460 which can be centered at affine points or points at infinity denoted by $P_{\pm\infty}$.
 461 In order to compute whether or not the $P_{\pm\infty}$ are affine, we must study the
 462 (four) limits

$$\lim_{t \rightarrow \pm\infty} \mathbf{x} \left(t, \sqrt{p(t)} \right), \quad \lim_{t \rightarrow \pm\infty} \mathbf{x} \left(t, -\sqrt{p(t)} \right). \quad (12)$$

463 Notice that we can have at most two different finite values in these limits,
 464 corresponding to the case when all $P_{\pm\infty}$ are affine. In order to compute these
 465 limits, after performing elementary calculations we arrive to an expression
 466 $\frac{\mu_1(t)}{\mu_2(t)}$ where one of the $\mu_i(t)$ is a polynomial, and the other $\mu_i(t)$ involves
 467 polynomials and one radical term. Then the limit can be evaluated by just
 468 comparing the degrees of the numerator and the denominator; notice that the
 469 degree can be a non-integer, rational number in the case of the numerator
 470 or denominator involving a square-root. In our experimentation we have
 471 checked that a computer algebra system like Maple 18 perfectly computes
 472 these limits in almost no time.

473 It can happen that all $P_{\pm\infty}$, only some of them, or none of them, is affine.
 474 If all $P_{\pm\infty}$ are affine and equal, then $P_{\pm\infty}$ is a self-intersection of \mathcal{C} . In this
 475 case, if the branches at infinity of \mathcal{G} are real, then there are at least two real
 476 branches of \mathcal{C} passing through $P_{\pm\infty}$; if the branches are complex and $P_{\pm\infty}$ is
 477 real, then $P_{\pm\infty}$ is an isolated point of \mathcal{C} . If some $P_{\pm\infty}$ is affine, it can also
 478 be a self-intersection of \mathcal{C} when there exists an affine point of \mathcal{G} whose image
 479 under $\mathbf{x}(t, s)$ coincides with this $P_{\pm\infty}$. This can be checked by solving the
 480 bivariate system $\{\mathbf{x}(t, s) = P_{\pm\infty}, g(t, s) = 0\}$.

481 Additionally, when some of the $P_{\pm\infty}$ are affine, we can check whether
 482 they are local singularities by checking whether the limit for $t \rightarrow \pm\infty$ of the
 483 derivative of $\mathbf{x}(t, \pm\sqrt{p(t)})$ vanishes.

484 3.4. Construction of $G_{\mathcal{C}}$.

485 Let $Q_1 = (t_1, s_1), \dots, Q_r = (t_r, s_r)$ be the points of \mathcal{G} computed in (i)-
 486 (iv) (see the beginning of Section 3). Since the Q_i belong to \mathcal{G} and the
 487 graph associated with \mathcal{G} can be computed by means of well-known methods
 488 [7, 13, 17, 21], we know how to connect the Q_i to each other. Furthermore,
 489 from the preceding sections the behavior of \mathbf{x} around the Q_i is clear. Now
 490 the vertices of $G_{\mathcal{C}}$ are the images $P_i = \mathbf{x}(Q_i)$, whenever $\mathbf{x}(Q_i)$ (or the limit of
 491 $\mathbf{x}(t, s)$ as $(t, s) \rightarrow Q_i$, in the case of base points) is defined, and we connect
 492 two of these vertices iff their preimages Q_i are connected to each other in $G_{\mathcal{G}}$.
 493 Furthermore, we also include as vertices of $G_{\mathcal{C}}$ the points $P_{\pm\infty} \in \mathcal{C}$ coming
 494 from the point at infinity of \mathcal{G} , in case they are affine.

495 Additionally, the graph associated with \mathcal{G} can have open edges (representing
 496 branches tending to infinity), corresponding to the edges of \mathcal{G} with
 497 some vertex where some component of \mathbf{x} becomes infinite, or branches of \mathcal{G}
 498 tending to infinity, in the case when some $P_{\pm\infty}$ is at infinity. Also, we must
 499 check that the edges of the graph associated with \mathcal{C} do not intersect except
 500 at the self-intersections of \mathcal{C} . This is not impossible. However, we can check
 501 whether this happens by computing the number of self-intersections of the
 502 edges of the graph, and checking whether this number agrees with the number
 503 of self-intersections, which has been computed previously. Notice that in
 504 order to check whether two segments intersect it is not necessary to explicitly
 505 find the equations of the lines containing the segments, or solving a linear
 506 system of equations. It can be decided directly from the coordinates of the
 507 vertices, and is a usual operation in Computational Geometry, negligible in
 508 terms of computation time. If the number of crossings between the edges is
 509 higher than the number of self-intersections of \mathcal{C} , previously determined, we
 510 just introduce additional vertices in the graph until the spurious crossings
 511 are avoided. In the following theorem, we will assume that this test has been
 512 carried out, so that the number of self-intersections is correct.

513 **Theorem 9.** *Let $G_{\mathcal{C}}$ be the graph associated with \mathcal{C} according to the descrip-*
 514 *tion in the preceding subsections. Then $G_{\mathcal{C}}$ and \mathcal{C} are isotopic.*

515 *Proof.* Once we compute the points of \mathcal{G} where \mathbf{x} becomes infinite, \mathcal{G} is
 516 segmented into finitely many portions ℓ_1, \dots, ℓ_p where \mathbf{x} is continuous. Each
 517 ℓ_i is connected, and by continuity $\mathbf{x}(\ell_i)$ is connected as well. Furthermore,
 518 by the birationality of $\mathbf{x}|_{\mathcal{G}}$ the correspondence between the ℓ_i and the $\mathbf{x}(\ell_i)$ is
 519 $1 : 1$. Since $\mathcal{C} = \mathbf{x}(\mathcal{G})$ and $\mathbf{x}(\mathcal{C})$ coincides with the union of the $\mathbf{x}(\ell_i)$, we just

520 need to show that the graph $G_{\mathcal{C}}$ is isotopic to the union of the $\mathbf{x}(\ell_i)$. Since
521 in $G_{\mathcal{C}}$ we are just deforming each $\mathbf{x}(\ell_i)$ into a segment, in order to show that
522 $G_{\mathcal{C}}$ and \mathcal{C} are isotopic we just need to show that no self-intersections of \mathcal{C}
523 are missed, and that no other self-intersections are introduced. The former
524 is guaranteed by construction, since in the process of computing $G_{\mathcal{C}}$ all the
525 self-intersections of \mathcal{C} are identified. The latter is guaranteed by checking
526 that two edges do not intersect at a point which is not a self-intersection of
527 \mathcal{C} . \square

Example 1. *Let*

$$g(t, s) = s^2 + t^4 - t^3 - 27t^2 + 25t + 50 = 0,$$

528 *and let*

$$\mathbf{x}(t, s) = (x(t, s), y(t, s)) = \left(\frac{t^4 - t^3 + t^2 + 5s - t}{t^6 + 1}, \frac{t^4 + t^3 - t^2 - 5s + t}{t^6 + 1} \right).$$

529 *The curve $\mathcal{C} = \mathbf{x}(\mathcal{G})$ is a hyperelliptic curve of genus one.*

530 *First we compute the real points $(t, s) \in \mathcal{G}$ generating the vertices of $G_{\mathcal{C}}$:*

(i) *Critical points of $g(t, s) = 0$, i.e. points $(t, 0)$ with $p(t) = 0$:*

$$Q_1 = (-5, 0), Q_2 = (-1, 0), Q_3 = (2, 0) \text{ and } Q_4 = (5, 0).$$

(ii) *Points of \mathcal{G} giving rise to critical points of \mathcal{C} . Local singularities and ramification points are generated by the points (t, s) solutions of the system*

$$g(t, s) = 0, \quad x_t g_s - x_s g_t = 0.$$

The real solutions (written only with two digits) are:

$$Q_5 = (-4.98, -2.05), Q_6 = (-3.21, -13.00), Q_7 = (-1.16, -3.47),$$

$$Q_8 = (-1.12, 3.08), Q_9 = (2.15, 3.11), Q_{10} = (2.24, -3.97),$$

$$Q_{11} = (3.76, -9.54), Q_{12} = (4.96, -2.52).$$

531 *Now we compute the points of \mathcal{G} giving rise to self-intersections of \mathcal{C} .*

532 *We have:*

$$\xi_1(x, t) = (t^{12} + 2t^6 + 1)x^2 + (-2t^{10} + 2t^9 + \dots)x + t^8 + \dots + 1250,$$

and

$$\xi_2(x, y, t) = (t^6 + 1)(x + y) - 2t^4.$$

533 The self-intersections of \mathcal{C} are generated by the real solutions of the
 534 system $\{\mathbf{sres}_1(x(t, s), y(t, s)) = 0, g(t, s) = 0\}$, which are $Q_{13} =$
 535 $(-3.75, -13.14)$, $Q_{14} = (-2.32, -10.61)$, $Q_{15} = (2.32, -4.62)$ and $Q_{16} =$
 536 $(3.75, 9.53)$.

537 The points Q_{13} and Q_{16} both generate the same point, P_{13} , and the
 538 points Q_{14} and Q_{15} both generate the point P_{14} (see Figure 5).

539 (iii) Points of \mathcal{G} where some component of \mathbf{x} is not defined: there are neither
 540 base points nor vertical asymptotes.

541 (iv) Starting and ending points for open branches of G : There are not open
 542 branches. In particular, in this case we do not need to analyze the
 543 points $P_{\pm\infty}$, since they are either non-real, or real isolated points of \mathcal{C} ,
 544 which we do not consider.

545 Finally, we compute the images $P_i = \mathbf{x}(Q_i)$, and we connect them according
 546 to how the Q_i are connected in \mathcal{G} . The graph associated with \mathcal{G} is shown in
 547 Fig. 4 (left). The graph associated with \mathcal{C} is also shown in Fig. 4 (right).
 548 Additionally, in the graph associated there are several points very close to
 549 each other: some details on the topology of \mathcal{C} are given in Fig. 5.

550 4. The space case.

551 Here we consider $\mathbf{x} : \mathbb{R}^2 \rightarrow \mathbb{R}^3$, where

$$\mathbf{x}(t, s) = (x(t, s), y(t, s), z(t, s)) = \left(\frac{A_1(t, s)}{B_1(t, s)}, \frac{A_2(t, s)}{B_2(t, s)}, \frac{A_3(t, s)}{B_3(t, s)} \right).$$

552 We let $\mathcal{C} = \mathbf{x}(\mathcal{G})$, where \mathcal{G} is defined by Eq. (2). In this case, we follow
 553 the same strategy already used in papers like [12, 14, 18]: first, birationally
 554 project \mathcal{C} onto the xy -plane, then compute the topology of the projection (in
 555 our case, using the results in Section 3), and then lift this projection to get
 556 the topology of the curve \mathcal{C} .

557 Let $\mathcal{C}^* = \pi_{xy}(\mathcal{C})$, where π_{xy} denotes the projection onto the xy -plane, and
 558 let $\tilde{\mathbf{x}} = \pi_{xy} \circ \mathbf{x}$. Fig. 6 illustrates the relationship between \mathcal{G} , \mathcal{C} and \mathcal{C}^* . We
 559 need two hypotheses this time:

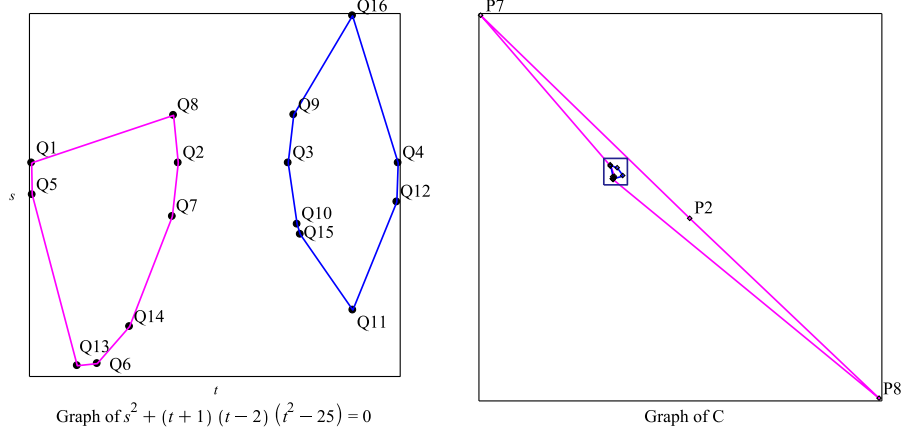


Figure 4: Correspondence between the edges of $G_{\mathcal{G}}$ and $G_{\mathcal{C}}$.

- 560 (H1) The restriction $\tilde{\mathbf{x}}|_{\mathcal{G}}$ is birational.
- 561 (H2) The curve \mathcal{C}^* does not have any asymptotes parallel to either the y -axis,
562 or the z -axis.

563 It is also customary, when computing the topology of a space curve \mathcal{C} , to
564 require that \mathcal{C} has no component parallel to the z -axis. However, in our case
565 \mathcal{C} is irreducible, i.e. \mathcal{C} consists of only one component. If \mathcal{C} reduces to a line
566 parallel to the z -axis, then the only possibility is that both $x(t, s), y(t, s)$ are
567 constant, which is a trivial case.

568 Hypothesis (H1) implies that \mathbf{x} itself is birational when restricted to \mathcal{G} ,
569 and that π_{xy} is also birational when restricted to \mathcal{C} ; in turn, this means that
570 there are not two different branches of \mathcal{C} projecting as a same branch of \mathcal{C}^* ,
571 and therefore that the branches of \mathcal{C} are the result of lifting to space the
572 branches of the projection $\mathcal{C}^* = \pi_{xy}(\mathcal{C})$. Hypothesis (H1) can be checked, as
573 observed in Section 3, by taking a random point $(t_0, s_0) \in \mathcal{G}$ and determining
574 the preimages of $\tilde{\mathbf{x}}(t_0, s_0)$. Hypothesis (H2) can be checked by testing whether
575 or not $B_2(t, s) = g(t, s) = 0$ has some solution where $A_2(t, s) \cdot B_1(t, s) \neq$
576 0 , and whether or not $A_2(t, s) = g(t, s) = 0$ has some solution where
577 $A_1(t, s) \cdot B_2(t, s) \neq 0$. Both hypotheses, (H1) and (H2), guarantee that:
578 (i) the topology of \mathcal{C}^* could be computed by applying the ideas in Section 3;
579 (ii) the topology of \mathcal{C} could be computed from the topology of \mathcal{C}^* , by lifting

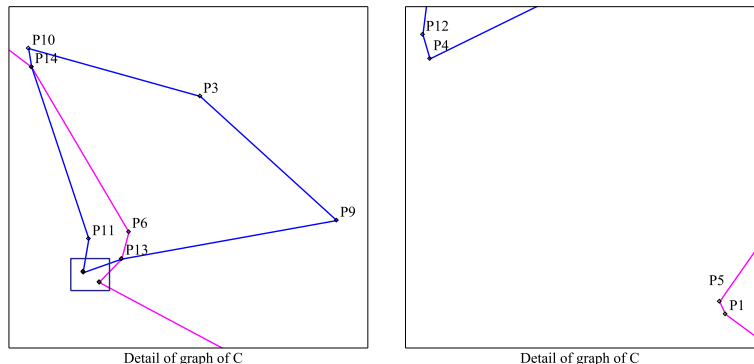


Figure 5: Details

580 a (planar) graph isotopic to \mathcal{C}^* . In our case, however, we do not need to
 581 compute first the topology of \mathcal{C}^* ; instead, as in Section 3, we determine all
 582 the points $(t, s) \in \mathcal{G}$ giving rise to “notable” points of \mathcal{C} , and incorporate
 583 those points as vertices of $G_{\mathcal{G}}$. Then the edges of $G_{\mathcal{G}}$ are mapped onto edges
 584 of $G_{\mathcal{C}}$ as we did in Section 3.

585 Hypotheses (H1) and (H2) can always be achieved when $\mathbf{x}|_{\mathcal{G}}$ is birational.
 586 Indeed, under this assumption, for almost all random affine changes of co-
 587 ordinates ϕ and renaming $\mathbf{x} := \mathbf{x} \circ \phi$, $\pi_{xy}|_{\mathcal{C}}$ is birational, i.e. two different
 588 branches of \mathcal{C} do not project as a same branch of \mathcal{C}^* . As a consequence $\tilde{\mathbf{x}}|_{\mathcal{G}}$
 589 must be birational.

590 In this case, we need to include the following points as vertices of $G_{\mathcal{G}}$:

- 591 (i) *Critical points of $g(t, s) = 0$, i.e. points of \mathcal{G} where $g_s = 0$.*
- 592 (ii) *Points of \mathcal{G} giving rise to critical points of \mathcal{C}^* .*
- 593 (iii) *Points of \mathcal{G} where some component of \mathbf{x} is not defined.*
- 594 (iv) *Starting and ending points for open branches of \mathcal{G} .*

595 The points in (i), (ii), (iii) are computed as in Section 3; observe that the
 596 pairs (t, s) generating singularities and points of \mathcal{C} with tangent parallel to
 597 the z -axis are among the critical points of \mathcal{C}^* (see [5, 4]). Once the points
 598 $Q_i = (t_i, s_i)$, $i = 1, \dots, r$ in (i)-(iv) are computed, we can find, whenever
 599 they are defined, the images $P_i = \mathbf{x}(Q_i)$ or the limit points and proceed as
 600 in Section 3 in order to connect the Q_i .

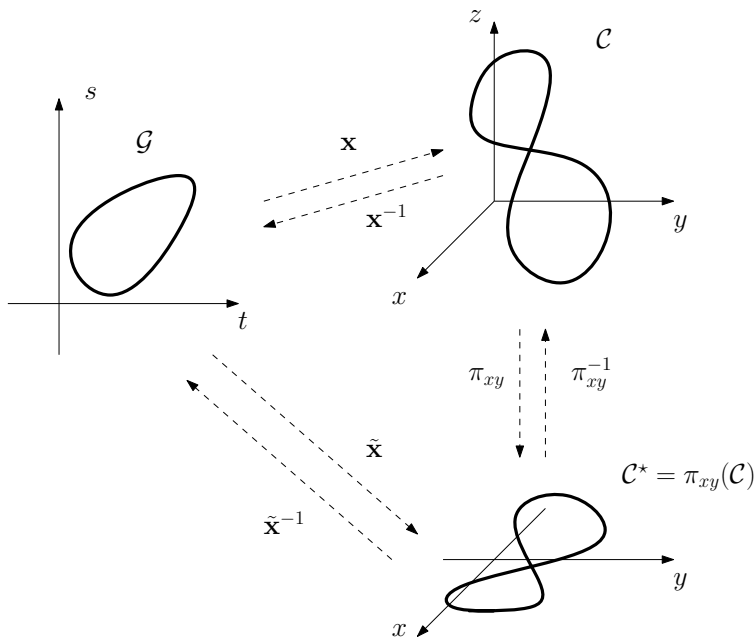


Figure 6: Relationship between the curves \mathcal{G} , \mathcal{C} , and \mathcal{C}^* .

601 **5. Experimentation.**

602 In this section we report on the experimentation carried out in the case
 603 of both 2D and 3D curves. The algorithms have been implemented in **Maple**
 604 2017, and the examples run on an Intel Core i3 processor with speeds revving
 605 up to 3.06 GHz.

606 Next, we first present examples of the 2D algorithm. In Table 1, we
 607 include for each curve, the genus, the total degree (d_i) and the number of
 608 terms of the implicit equation (n.terms), the timings in seconds (t_0) taken
 609 by our algorithm, and the timings in seconds (t_1) corresponding to the algo-
 610 rithm in [21], also implemented in **Maple**, which uses the implicit equation
 611 of the curve. Additionally, in Table 1 we checkmark whether each example
 612 corresponds to a case where the points $P_{\pm\infty}$ are affine (the column $P_{\pm\infty}$ aff.),
 613 and whether the curve has self-intersections (S.I.). The last column provides
 614 some extra comments on the existence of base points or asymptotes. The
 615 parametrizations corresponding to these examples are given in Appendix II
 616 of [3], the ArXiv version of this paper. The graphs corresponding to the
 617 examples in Table 1 are shown in Figure (7); from left to right, we have

618 Examples 1, 2, 3 in the first row, 4, 5, 6 in the second row and 7, 8, 9 in the
 619 third row.

Example	genus	d_i	n.terms	$P_{\pm\infty}$ aff.	S.I.	t_0	t_1	Obs.
1	0	10	57	✓	✓	0.310	0.270	Asymptotes
2	1	14	81		✓	0.625	*	Asymptotes
3	2	6	26	✓	✓	0.398	0.110	
4	1	12	81	✓	✓	0.529	*	Base points
620 5	2	12	75	✓	✓	0.543	*	
6	2	11	75		✓	0.777	*	
7	2	12	75	✓	✓	0.443	*	
8	1	6	23	✓	✓	0.484	0.108	
9	2	9	55	✓	✓	1.069	0.308	

621 **Table 1:** 2D Examples.

622 *: Computation was cancelled after fifteen minutes.

623 Notice that when the algorithm in [21] succeeds, it provides better timings
 624 than our algorithm. However, in most cases the implicit equation of the curve
 625 is too big, and the algorithm in [21] gets stuck.

626 Finally, we present examples of the 3D algorithm. In Table 2, for each
 627 curve we include the genus, the total degree (d_i) and the number of terms
 628 of the implicit equation of the projection onto the xy -plane (n.terms), and
 629 the timing in seconds taken by our algorithm (t_0); the parametrizations cor-
 630 responding to each curve are given in Appendix III of [3], the ArXiv version
 631 of this paper. Additionally, we include two columns on the nature of $P_{\pm\infty}$
 632 and the existence of self-intersections, as in Table 1. In the last column
 633 we include some observations on how we generated the example, in some
 634 interesting cases.

635

Example	genus	d_i	n.terms	$P_{\pm\infty}$ aff.	S.I.	t_0	Obs.
1	4	10	66			1.543	
2	2	6	16	✓		0.344	Int. con. and quadric
3	7	16	153		✓	78.252	Int. ruled and quadric
4	3	8	42			0.537	Int. ruled and quadric
5	2	12	91			4.238	Int. bicubic patch and plane
6	1	4	9			0.201	
7	1	10	34			0.352	
8	2	19	61	✓	✓	1.031	
9	2	9	55	✓	✓	0.949	

Table 2: 3D Examples.

The pictures corresponding to these curves are shown in Figure 8. Notice that the timing in Ex. 3 is considerably higher, which is expectable because both the Weierstrass curve and the mapping $\mathbf{x}(t, s)$ are dense and with high degree.

6. Complexity and certification issues.

In this section we present the complexity of the algorithms presented in the previous sections, and we elaborate on how to certify the topology of the curves. To certify the topology we must be sure whether two different points $(t_i, s_i) \neq (t_j, s_j)$, both belonging to \mathcal{G} , satisfy $\mathbf{x}(t_i, s_i) = \mathbf{x}(t_j, s_j)$, that is whether they give rise to the same point $P \in \mathcal{C}$. We first analyze the complexity of the algorithm without the certification step: in particular, the timings corresponding to Section 5 do not include this certification. Then, we address certification issues and provide the complexity of the algorithm including the certification step. We analyze the algorithm for 3D curves: the complexity bound is the same for 2D and 3D curves.

6.1. Complexity (I)

In this section we present the bit complexity analysis of the algorithm without the certification step. This is the algorithm for which we perform experiments in Section 5. We denote the maximum bitsize by $\mathcal{L}(f)$ of the coefficients of a polynomial f . Additionally, we denote by \mathcal{O} , $\tilde{\mathcal{O}}$, $\tilde{\mathcal{O}}_B$ the arithmetic complexity, the arithmetic complexity neglecting logarithmic factors, and the bit complexity (also neglecting logarithmic factors), respectively.

660 Let

$$\mathbf{x}(t, s) = \left(\frac{a_{11}(t) + sa_{12}(t)}{b_{11}(t) + sb_{12}(t)}, \frac{a_{21}(t) + sa_{22}(t)}{b_{21}(t) + sb_{22}(t)}, \frac{a_{31}(t) + sa_{32}(t)}{b_{31}(t) + sb_{32}(t)} \right).$$

661 We consider the following 3 polynomials:

$$\begin{aligned} X(t, s) &= (b_{11}(t) + sb_{12}(t))x - (a_{11}(t) + sa_{12}(t)), \\ Y(t, s) &= (b_{21}(t) + sb_{22}(t))y - (a_{21}(t) + sa_{22}(t)), \\ Z(t, s) &= (b_{31}(t) + sb_{32}(t))z - (a_{31}(t) + sa_{32}(t)). \end{aligned}$$

662 We also recall that $g(t, s) = s^2 - p(t)$. We assume that all the univariate
663 polynomials in t , that is the $a_{ij}(t), b_{ij}(t)$, and $p(t)$, have degree at most d ,
664 and that their coefficients are integers of maximum bitsize at most τ .

665 The process of the algorithm goes as follows:

666 (*Step 1*) Compute the resultants

$$E_0 = \text{res}_s(X, Y), \quad E_1 = \text{res}_s(X, g).$$

667 The polynomial E_0 satisfies that $E_0 \in \mathbb{Z}[x, y, t]$. The degree of E_0 with
668 respect to x and y is 1 and with respect to t is $\leq 2d = \mathcal{O}(d)$; moreover
669 $\mathcal{L}(E_0) = \tilde{\mathcal{O}}(\tau)$. The polynomial E_1 satisfies that $E_1 \in \mathbb{Z}[x, t]$. The degree
670 of E_1 with respect to x is 2 and with respect to t is $\leq 3d = \mathcal{O}(d)$; also
671 $\mathcal{L}(E_1) = \tilde{\mathcal{O}}(\tau)$.

672 Since the degree of X, Y, Z and g with respect to x, y, s is at most 2, we
673 can compute the resultants E_0 and E_1 by performing a constant number of
674 multiplications of univariate polynomials in t . By recalling that the maxi-
675 mum degree with respect to t is $\tilde{\mathcal{O}}(d)$, we deduce that the cost of computing
676 E_0 and E_1 is $\tilde{\mathcal{O}}_B(d\tau)$ [30].

677 (*Step 2*) Compute the subresultant sequence of E_0 and E_1 with respect to t .

678 From the subresultant sequence we are interested in the polynomial of
679 degree 1 with respect to t . This is the first subresultant polynomial; we can
680 compute it in $\tilde{\mathcal{O}}_B(d^4\tau)$ [16, Lemma 8]. Let the coefficient of degree 1 of this
681 polynomial be $\mathbf{sres}_1 \in \mathbb{Z}[x, y]$ (i.e. the first principal subresultant). It has
682 degree $\tilde{\mathcal{O}}(d)$ and bitsize $\tilde{\mathcal{O}}(d\tau)$ [16, Lemma 8].

683 (*Step 3*) Substitute the parametrization $\mathbf{x}(t, s)$ in \mathbf{sres}_1 .

684 After clearing denominators we obtain a polynomial $M(t, s) \in \mathbb{Z}[t, s]$. The
685 degree of $M(t, s)$ with respect to t and s is $\tilde{\mathcal{O}}(d)$ and its bitsize is $\tilde{\mathcal{O}}(d^2\tau)$. This

686 calculation of $M(t, s)$ involves $\mathcal{O}(d)$ multiplications of bivariate polynomials
 687 in s and t . This cost is $\tilde{\mathcal{O}}_B(d^5\tau)$ [25, 30].

688 (*Step 4*) Solve the polynomial system $M(t, s) = g(t, s) = 0$.

689 We can solve the system in $\tilde{\mathcal{O}}_B(d^7\tau)$ (or $\tilde{\mathcal{O}}_B(d^8\tau)$) [19, 10].

690 After solving the system, we compute the images under the birational
 691 mapping $\mathbf{x}(t, s)$ of all the points (t, s) computed along the way, and connect
 692 them properly.

693 The whole complexity is dominated by the complexity of solving the poly-
 694 nomial system $(\Sigma)\{M(t, s) = g(t, s) = 0\}$, so we get a final bound of $\tilde{\mathcal{O}}_B(d^7\tau)$
 695 (or $\tilde{\mathcal{O}}_B(d^8\tau)$), without including certification.

696 6.2. Certification and complexity (II)

697 In this subsection we consider certification strategies, and we present the
 698 complexity of the algorithm including this certification. We perform the
 699 certification by exploiting the rational univariate representation of the real
 700 roots of the polynomial system $(\Sigma)\{M(t, s) = g(t, s) = 0\}$.

701 Within the complexity bound given in the previous subsection for solving
 702 the bivariate system (Σ) , we can compute both an isolating interval represen-
 703 tation of the real roots, as a well a (sparse) rational univariate representation
 704 (SRUR) [10], see also [25]. The latter represents the tuples (t, s) of the so-
 705 lutions os (Σ) as $\left(\frac{F_1(\theta)}{F_0(\theta)}, \frac{F_2(\theta)}{F_0(\theta)}\right)$, where θ runs over all the (real) roots of a
 706 (univariate) polynomial $F(\theta)$ and F_0, F_1 , and F_2 are univariate polynomi-
 707 als. This representation involves univariate polynomials of degree $\tilde{\mathcal{O}}(d^2)$ and
 708 bitsize $\tilde{\mathcal{O}}(d^3\tau)$.

709 Now we want to identify which tuples of solutions of the polynomial
 710 system $M(t, s) = g(t, s) = 0$ give rise to the same point on space curve. Or
 711 in other words, we want to *certify* when two tuples give rise to the same
 712 point on the space curve.

713 Say that (α_1, β_1) and (α_2, β_2) are two different solutions of the polynomial
 714 system (Σ) . Assume further that they correspond to the roots θ_1 and θ_2 of
 715 the polynomial $F(\theta)$. Thus, their rational univariate representation is

$$\left(\frac{F_1(\theta_1)}{F_0(\theta_1)}, \frac{F_2(\theta_1)}{F_0(\theta_1)}\right) \quad \text{and} \quad \left(\frac{F_1(\theta_2)}{F_0(\theta_2)}, \frac{F_2(\theta_2)}{F_0(\theta_2)}\right),$$

716 with $F(\theta_1) = 0, F(\theta_2) = 0$.

We check if they correspond to the same point by exploiting the parametrization \mathbf{x} . For example, to test if they result in the same x -coordinate, we should test whether or not

$$\frac{a_{11}(\alpha_1) + \beta_1 a_{12}(\alpha_1)}{b_{11}(\alpha_1) + \beta_1 b_{12}(\alpha_1)} = \frac{a_{11}(\alpha_2) + \beta_2 a_{12}(\alpha_2)}{b_{11}(\alpha_2) + \beta_2 b_{12}(\alpha_2)}.$$

717 Clearing denominators, we get $\widehat{G}(\alpha_1, \alpha_2) = 0$. Now if we substitute the
 718 rational univariate representation of the roots and clear denominators, then
 719 we get a new bivariate polynomial G , and we need to test whether or not
 720 $G(\theta_1, \theta_2) = 0$.

721 The degree of G is $\widetilde{\mathcal{O}}(d^3)$, in θ_1 and θ_2 and its bitsize is $\widetilde{\mathcal{O}}(d^4\tau)$. The
 722 complexity of computing G involves the multiplication of $\widetilde{\mathcal{O}}(d)$ univariate
 723 polynomials and is $\widetilde{\mathcal{O}}_B(d^8\tau)$. The cost of this bivariate sign evaluation is
 724 $\widetilde{\mathcal{O}}_B(d^{15}\tau)$.

725 We must perform this bivariate sign evaluation for every pair (θ_i, θ_j) of
 726 roots of F , and test for all coordinates (x, y, z) . There are $\widetilde{\mathcal{O}}(d^4)$ pairs of
 727 solutions to test and the total cost is $\widetilde{\mathcal{O}}_B(d^{19}\tau)$. This complexity bound of
 728 certification dominates the overall complexity of the algorithm.

729 We have implemented the certification part and the timings we get are in
 730 agreement with this complexity: although there can be examples where the
 731 computing time is reasonable, in general the timings are very high and further
 732 research needs to be done. It seems plausible to improve the complexity of
 733 certification by exploiting more carefully aggregate separation bounds for
 734 the real roots of polynomial systems [20]. For example, we can apply this
 735 aggregation when we perform the time consuming sign evaluation of G over
 736 all the roots of the polynomial F . There should be a gain of a factor d^2 with
 737 this approach.

738 However, the most promising direction is to use more advanced (proba-
 739 bilistic) tests for checking equality of real algebraic numbers [9]. The reader
 740 might notice that we do not really need the actual sign evaluation of G at
 741 two real algebraic numbers. What we really need is to test whether or not
 742 the evaluation of $G(\theta_1, \theta_2)$ is zero or not.

743 6.3. Comparison of complexities with implicit algorithms.

744 A possibility to compute the topology of \mathcal{C} is to compute first an implicit
 745 representation of the curve, and then to apply an algorithm to complete the
 746 topology of an implicit curve. In the planar case, the implicit representation

747 requires just one bivariate polynomial $f(x, y)$, that can be computed using
 748 Gröbner bases. Denoting the degree of $f(x, y)$ by n , and denoting by τ_f the
 749 bitsize of the coefficients of f , the complexity of computing the topology of
 750 $f(x, y) = 0$ is $\tilde{\mathcal{O}}_B(n^6 + n^5\tau_f)$. In our case $n = \tilde{\mathcal{O}}(d)$ and $\tau_f = \tilde{\mathcal{O}}(d\tau)$, so we
 751 reach a complexity of $\tilde{\mathcal{O}}_B(d^6\tau)$, certainly better than the bound we give in
 752 Subsection 6.2.

753 In the space case, however, the situation is much more difficult. An im-
 754 plicit representation of \mathcal{C} requires to compute a basis for the ideal of the
 755 curve, which might have more than two polynomials. Even if \mathcal{C} is implicitly
 756 defined by only two polynomials $f_i(x, y, z)$, with $i = 1, 2$, the known com-
 757 plexities for implicit algorithms are worse than ours. In [15], one has the
 758 bound $\tilde{\mathcal{O}}(n^{21}\tau_f)$, where n, τ_f are bounds for the degrees and bitsizes of the
 759 f_i , respectively. For the same case, in [12] one has the bound $\tilde{\mathcal{O}}(n^{37}\tau_f)$.

760 7. Conclusion.

761 We have presented algorithms to compute the topology of 2D and 3D
 762 hyperelliptic curves that do not require to compute or make use of the implicit
 763 representation of the curve. The main idea is to see the hyperelliptic curve
 764 as the image of a planar curve, the Weierstrass form of the curve, under a
 765 birational mapping of the plane or the space. Seeing the curve this way,
 766 the algorithms determines how the topology of the Weierstrass form changes
 767 when the birational mapping is applied. While a not completely certified
 768 algorithm produces good and fast results, a completely certified algorithm
 769 is much slower, although it is competitive in the space case, in terms of
 770 complexity, with algorithms using an implicit representation of the curve.
 771 Some lines of improvement to speed up the certification are suggested in the
 772 paper. We plan to exploit these ideas in the future to get a faster, certified,
 773 algorithm.

774 References

775 References

- 776 [1] J. G. Alcázar and J.R. Sendra. Local shape of offsets to algebraic curves.
 777 *Journal of Symbolic Computation*, 42:338–351, 2007.
- 778 [2] Juan Gerardo Alcázar, Jorge Caravantes, and Gema M Díaz-Toca. A
 779 new method to compute the singularities of offsets to rational plane

- 780 curves. *Journal of Computational and Applied Mathematics*, 290:385–
781 402, 2015.
- 782 [3] Juan Gerardo Alcázar, Jorge Carvantes, Gema María Díaz Toca, and
783 Elias Tsigaridas. ArXiv 1812.11498, 2018.
- 784 [4] Juan Gerardo Alcázar and Gema María Díaz-Toca. Topology of 2d and
785 3d rational curves. *Computer Aided Geometric Design*, 27(7):483–502,
786 2010.
- 787 [5] Juan Gerardo Alcázar and J Rafael Sendra. Computation of the topol-
788 ogy of real algebraic space curves. *Journal of Symbolic Computation*,
789 39(6):719–744, 2005.
- 790 [6] Eric Berberich, Pavel Emeliyanenko, Alexander Kobel, and Michael
791 Sagraloff. Arrangement computation for planar algebraic curves. In
792 *Proceedings of the 2011 International Workshop on Symbolic-Numeric*
793 *Computation*, pages 88–98. ACM, 2012.
- 794 [7] Eric Berberich, Pavel Emeliyanenko, Alexander Kobel, and Michael
795 Sagraloff. Exact symbolic–numeric computation of planar algebraic
796 curves. *Theoretical Computer Science*, 491:1–32, 2013.
- 797 [8] Michal Bizzarri, Miroslav Lávička, and Jan Vršek. Piecewise rational ap-
798 proximation of square-root parameterizable curves using the weierstrass
799 form. *Computer Aided Geometric Design*, 56:52–66, 2017.
- 800 [9] Johannes Blomer. Computing sums of radicals in polynomial time.
801 In *Foundations of Computer Science, 1991. Proceedings., 32nd Annual*
802 *Symposium on*, pages 670–677. IEEE, 1991.
- 803 [10] Yacine Bouzidi, Sylvain Lazard, Guillaume Moroz, Marc Pouget, Fabrice
804 Rouillier, and Michael Sagraloff. Solving bivariate systems using rational
805 univariate representations. *J. Complex.*, 37(C):34–75, December 2016.
- 806 [11] Jorge Carvantes, Gema M Díaz-Toca, Laureano González-Vega, and
807 Ioana Necula. An algebraic framework for computing the topology of
808 offsets to rational curves. *Computer Aided Geometric Design*, 52:28–47,
809 2017.

- 810 [12] Jin-San Cheng, Kai Jina, and Daniel Lazard. Certified rational para-
811 metric approximation of real algebraic space curves with local generic
812 position method. *Journal of Symbolic Computation*, 58:18–40, 2013.
- 813 [13] Jinsan Cheng, Sylvain Lazard, Luis Peñaranda, Marc Pouget, Fabrice
814 Rouillier, and Elias Tsigaridas. On the topology of planar algebraic
815 curves. In *Proceedings of the twenty-fifth annual symposium on Computa-*
816 *tational geometry*, pages 361–370. ACM, 2009.
- 817 [14] Diatta Niang Daouda, Bernard Mourrain, and Olivier Ruatta. On the
818 computation of the topology of a non-reduced implicit space curve. In
819 *Proceedings of the twenty-first international symposium on Symbolic and*
820 *algebraic computation*, pages 47–54. ACM, 2008.
- 821 [15] Daouda Diatta. *Calcul effectif de la topologie de courbes et surfaces*
822 *algebriques reelles*. Ph. D. Thesis, Universite de Limoges, 2009.
- 823 [16] Dimitrios I Diochnos, Ioannis Z Emiris, and Elias P Tsigaridas. On the
824 asymptotic and practical complexity of solving bivariate systems over
825 the reals. *Journal of Symbolic Computation*, 44(7):818–835, 2009.
- 826 [17] Arno Eigenwillig, Michael Kerber, and Nicola Wolpert. Fast and exact
827 geometric analysis of real algebraic plane curves. In *Proceedings of the*
828 *2007 international symposium on Symbolic and algebraic computation*,
829 pages 151–158. ACM, 2007.
- 830 [18] Mohammed El Kahoui. Topology of real algebraic space curves. *Journal*
831 *of Symbolic Computation*, 43(4):235–258, 2008.
- 832 [19] Pavel Emeliyanenko and Michael Sagraloff. On the complexity of solv-
833 ing a bivariate polynomial system. In *Proceedings of the 37th Interna-*
834 *tional Symposium on Symbolic and Algebraic Computation*, pages 154–
835 161. ACM, 2012.
- 836 [20] Ioannis Z Emiris, Bernard Mourrain, and Elias P Tsigaridas. The dmm
837 bound: Multivariate (aggregate) separation bounds. In *Proceedings of*
838 *the 2010 International Symposium on Symbolic and Algebraic Compu-*
839 *tation*, pages 243–250. ACM, 2010.

- 840 [21] Laureano González-Vega and Ioana Necula. Efficient topology deter-
841 mination of implicitly defined algebraic plane curves. *Computer aided*
842 *geometric design*, 19(9):719–743, 2002.
- 843 [22] Morris Hirsch. *Differential Topology*. Springer-Verlag, 1976.
- 844 [23] Shafarevich I.R. *Basic Algebraic Geometry 1 (Third edition)*. Springer-
845 Verlag, 2013.
- 846 [24] Alexander Kobel and Michael Sagraloff. On the complexity of computing
847 with planar algebraic curves. *Journal of Complexity*, 31(2):206–236,
848 2015.
- 849 [25] Angelos Mantzaflaris, Éric Schost, and Elias Tsigaridas. Sparse rational
850 univariate representation. In *ISSAC 2017-International Symposium on*
851 *Symbolic and Algebraic Computation*, page 8, 2017.
- 852 [26] Alfred Menezes, Robert Zuccherato, and Yi-Hong Wu. *An elementary*
853 *introduction to hyperelliptic curves*. Reseach Report CORR 96-19, Fac-
854 ulty of Mathematics, University of Waterloo, 1996.
- 855 [27] J Rafael Sendra, David Sevilla, and Carlos Villarino. Algebraic and algo-
856 rithmic aspects of radical parametrizations. *Computer Aided Geometric*
857 *Design*, 55:1–14, 2017.
- 858 [28] Juan Rafael Sendra, Franz Winkler, and Sonia Pérez-Díaz. *Rational*
859 *Algebraic Curves: A Computer Algebra Approach*. Springer Verlag, 2007.
- 860 [29] G.M. Díaz Toca. [http://webs.um.es/gemadiaz/miwiki/doku.php?](http://webs.um.es/gemadiaz/miwiki/doku.php?id=papers)
861 [id=papers](http://webs.um.es/gemadiaz/miwiki/doku.php?id=papers), 2018.
- 862 [30] Joachim Von Zur Gathen and Jürgen Gerhard. *Modern computer alge-*
863 *bra*. Cambridge university press, 2013.
- 864 [31] R.J. Walker. *Algebraic Curves*. Princeton University Press, 1950.

865 **8. Appendix I: remaining proofs.**

866 In this section we provide the proofs of some results in Section 3. We
867 start with Proposition 5.

868 *Proof.* (of Proposition 5) Let \mathcal{V} be the variety (the curve) in $\mathbb{R}^4(t, s, x, y)$
869 defined as

$$\mathcal{V} = V(\text{num}(x - x(t, s)), \text{num}(y - y(t, s)), g(t, s)),$$

870 and let $\widehat{\mathcal{V}} = \Pi_{txy}(\mathcal{V})$ be the projection of \mathcal{V} onto $\mathbb{R}^3(t, x, y)$; notice that
871 $\widehat{\mathcal{V}} \subset V(\xi_1, \xi_2)$. Suppose that (t_0, s_0, x_0, y_0) is smooth in \mathcal{V} . Using the Jacobian
872 matrix of $F_1(t, s, x) = \text{num}(x - x(t, s))$, $F_2(t, s, y) = \text{num}(y - y(t, s))$, $g(t, s)$
873 and condition (11), we observe that the tangent line to \mathcal{V} at (s_0, t_0, x_0, y_0) is
874 parallel to $(-g_s(t_0, s_0), g_t(t_0, s_0), 0, 0)$. If $g_s(t_0, s_0) \neq 0$ (i.e. if $s_0 \neq 0$) then
875 the point (t_0, x_0, y_0) is regular in $\widehat{\mathcal{V}}$ and the tangent line to $\widehat{\mathcal{V}}$ at (t_0, x_0, y_0)
876 is $\{x = x_0, y = y_0\}$, which is parallel to the t -axis. Therefore, $\xi_1(t, x_0) = 0$
877 and $\xi_2(t, x_0, y_0) = 0$ share the root t_0 with multiplicity higher than 1, and
878 $\mathbf{sres}_1(x_0, y_0) = 0$. If $g_s(t_0, s_0) = 0$ (i.e. if $s_0 = 0$) then (t_0, x_0, y_0) is singular
879 in $\widehat{\mathcal{V}}$ and we can derive the same conclusion.

880 If, however, (s_0, t_0, x_0, y_0) is a singular point of $\widehat{\mathcal{V}}$, then the tangent space
881 to \mathcal{V} at (s_0, t_0, x_0, y_0) , i.e. the kernel of the Jacobian matrix, consists of the
882 vectors $(\alpha, \beta, 0, 0)$ with $\alpha, \beta \in \mathbb{C}$. Therefore, the line $\{x = x_0, y = y_0\}$ is
883 tangent to $\widehat{\mathcal{V}}$ at (t_0, x_0, y_0) and, therefore, all $\xi_i(t, x_0, y_0)$, $i = 1, 2$ have a
884 multiple root at $t = t_0$. This implies that $\mathbf{sres}_1(x_0, y_0) = 0$. \square

885 Now we prove Lemma 7. From definitions of ξ_1, ξ_2 in Eq. (8) and taking
886 into account that \mathbf{x} can be written as in Eq. (4), the polynomial $\xi_1(t, x)$ is
887 the square-free part of the resultant with respect to s of $g(t, s) = s^2 - p(t)$
888 and

$$\begin{aligned} h(t, s, x) &:= \text{num}(x - x(t, s)) = \\ &= x(b_{11}(t) + sb_{12}(t)) - (a_{11}(t) + sa_{12}(t)) = \\ &= s(xb_{12}(t) - a_{12}(t)) + xb_{11}(t) - a_{11}(t). \end{aligned}$$

889 Since $\text{degree}_s(g) = 2$ and $\text{degree}_s(h) \leq 1$, it is easy to compute such a
890 resultant; if $\text{degree}_s(h) = 1$, i.e. if $x(t, s)$ explicitly depends on s , then

$$\text{Res}_s(h, g) = (b_{11}^2 - pb_{12}^2)x^2 - 2(a_{11}b_{11} - pa_{12}b_{12})x + a_{11}^2 - pa_{12}^2, \quad (13)$$

891 where $b_{ij} = b_{ij}(t)$, $a_{ij} = a_{ij}(t)$ for $i = 1, 2$, $j = 1, 2$. If $\text{degree}_s(h) = 0$, i.e. if
 892 $x(t, s)$ does not depend on s , then

$$\text{Res}_s(h, g) = h = x(b_{11}(t) + sb_{12}(t)) - (a_{11}(t) + sa_{12}(t)). \quad (14)$$

893 As for $\xi_2(t, x, y)$, that is is the square-free part of the resultant with
 894 respect to s of $h(t, s, x)$ and

$$\begin{aligned} j(t, s, y) &:= \text{num}(y - y(t, s)) = \\ &= y(b_{21}(t) + sb_{22}(t)) - (a_{21}(t) + sa_{22}(t)) = \\ &= s(yb_{22}(t) - a_{22}(t)) + yb_{21}(t) - a_{21}(t). \end{aligned}$$

895 If $\text{degree}_s(h) = \text{degree}_s(j) = 1$, i.e. if both $x(t, s)$ and $y(t, s)$ explicitly depend
 896 on s , then

$$\text{Res}_s(h, j) = (a_{22}b_{11} - a_{21}b_{12})x + (a_{11}b_{22} - a_{12}b_{21})y + (b_{12}b_{21} - b_{11}b_{22})xy - a_{11}a_{22} + a_{12}a_{21}, \quad (15)$$

897 where $b_{ij} = b_{ij}(t)$, $a_{ij} = a_{ij}(t)$ for $i = 1, 2$, $j = 1, 2$. If $\text{degree}_s(h) = 0$, i.e. if
 898 $x(t, s)$ does not depend on s , then

$$\text{Res}_s(h, j) = h = x(b_{11}(t) + sb_{12}(t)) - (a_{11}(t) + sa_{12}(t)), \quad (16)$$

899 and if $\text{degree}_s(j) = 0$, i.e. if $y(t, s)$ does not depend on s , then

$$\text{Res}_s(h, j) = j = yb_{21}(t) - a_{21}(t). \quad (17)$$

900 *Proof.* (of Lemma 7) “ \Leftarrow ” Suppose that t_0 corresponds to a base point. The
 901 resultant of $h(t, s, x)$ and $g(t, s)$ is equal to Equation (13), and considered as
 902 a polynomial in x , it is easy to see that all its coefficients vanish at $t = t_0$.
 903 Thus, $t - t_0$ divides $\xi_1(x, t)$. Likewise, the resultant of $h(t, s, x)$ and $j(t, s, y)$
 904 is equal to Equation (15), and we can check that all its coefficients vanish in
 905 $t = t_0$. Thus, $t - t_0$ divides also $\xi_2(x, t)$.

906 “ \Rightarrow ” If $t - t_0$ divides ξ_1 , then, by properties of resultants, since the leading
 907 coefficient of $g(t, s)$ with respect to s is 1, there is s_0 with $g(t_0, s_0) = 0$ and

$$h(t_0, s_0, x) = x(b_{11}(t_0) + s_0b_{12}(t_0)) - (a_{11}(t_0) + s_0a_{12}(t_0)) = 0;$$

908 thus, $b_{11}(t_0) + s_0b_{12}(t_0) = a_{11}(t_0) + s_0a_{12}(t_0) = 0$.

Next, if $t - t_0$ divides ξ_2 , then either the leading coefficients of both $h(t, s, x)$ and $j(t, s, y)$ with respect to s vanish at $t = t_0$, or there exists s_1 such that $h(t_0, s_1, x) = j(t_0, s_1, y) = 0$ for all x, y . In the first case, we would have

$$b_{12}(t_0) = a_{12}(t_0) = b_{22}(t_0) = a_{22}(t_0) = 0.$$

However, since also $b_{11}(t_0) + s_0 b_{12}(t_0) = a_{11}(t_0) + s_0 a_{12}(t_0) = 0$, we should have

$$a_{11}(t_0) = a_{12}(t_0) = b_{12}(t_0) = b_{11}(t_0) = 0,$$

909 but this cannot happen because $\gcd(a_{11}, a_{12}, b_{11}, b_{12}) = 1$. Therefore, there
910 exists s_1 such that for all x, y

$$h(t_0, s_1, x) = x(b_{11}(t_0) + s_1 b_{12}(t_0)) - (a_{11}(t_0) + s_1 a_{12}(t_0)) = 0;$$

911

$$j(t_0, s_1, y) = y(b_{21}(t_0) + s_1 b_{22}(t_0)) - (a_{21}(t_0) + s_1 a_{22}(t_0)) = 0.$$

912 Then,

$$\begin{aligned} b_{11}(t_0) + s_1 b_{12}(t_0) &= a_{11}(t_0) + s_1 a_{12}(t_0) = 0, \\ b_{21}(t_0) + s_1 b_{22}(t_0) &= a_{21}(t_0) + s_1 a_{22}(t_0) = 0. \end{aligned}$$

913 Since we also know that $b_{11}(t_0) + s_0 b_{12}(t_0) = a_{11}(t_0) + s_0 a_{12}(t_0) = 0$, with
914 $(t_0, s_0) \in \mathcal{G}$, we deduce that either $s_1 = s_0$, or $b_{12}(t_0) = a_{12}(t_0) = 0$. However,
915 $b_{12}(t_0) = a_{12}(t_0) = 0$ implies that $b_{11}(t_0) = a_{11}(t_0) = 0$, which cannot happen
916 because $\gcd(a_{11}, a_{12}, b_{11}, b_{12}) = 1$. $s_0 = s_1$ with $g(t_0, s_0) = 0$. So, we can
917 conclude that t_0 corresponds to a base point of \mathbf{x} . \square

918 Finally, we prove Lemma 8.

919 *Proof.* (of Lemma 8) “ \Leftarrow ” If $x(t, s) = x(t)$, then $\xi_1(t, x) = \xi_2(t, x, y) =$
920 $b_{11}(t)x - a_{11}(t)$, and the result follows.

921 “ \Rightarrow ” By way of contradiction, suppose that $\xi_1(t, x)$ and $\xi_2(t, x, y)$ have
922 a factor $\eta(t, x)$ depending on both x, t and that $x(t, s)$ also depends on s .
923 Notice that taking Eq. (17) into account, if $\xi_1(t, x)$ and $\xi_2(t, x, y)$ have a
924 factor $\eta(t, x)$ depending on both x, t then $y(t, s)$ must depend on s as well.
925 So both $x(t, s)$ and $y(t, s)$ depend on s . Then $\xi_2(t, x, y)$ is the square-free
926 part of Eq. (15), so $\eta(t, x)$ must be linear in x . Therefore either $\xi_2(t, x, y)$
927 coincides with $\eta(t, x)$, or $\xi_2(t, x, y)$ has another factor $\gamma(t, y)$ whose degree in
928 y is at most 1. Now we distinguish two cases:

929 (i) If $\text{degree}_y(\gamma) = 1$, then for all (t_0, y_0) such that $\gamma(t_0, y_0) = 0$, either the
930 leading coefficients of h, j with respect to s vanish at (t_0, y_0) for all x ,
931 or there exists s_0 such that h, j integrally vanish at (t_0, s_0, y_0) for all
932 x . The first possibility implies that both leading coefficients are zero
933 modulo $\gamma(t, y)$, and this cannot happen because the leading coefficient
934 of h with respect to s depends on x . But the second possibility cannot
935 happen either, because that would imply that $x(t, s)$ has infinitely many
936 base points.

937 (ii) If $\text{degree}_y(\gamma) = 0$, then for all (t_0, x_0) such that $\eta(t_0, x_0) = 0$, either the
938 leading coefficients of h, j with respect to s vanish at (t_0, x_0) for all y ,
939 or there exists s_0 such that h, j integrally vanish at (t_0, s_0, y_0) for all y .
940 Then we argue as before, this time with j and $y(t, s)$.

941 Thus we conclude that $x(t, s)$ cannot depend explicitly on s , and the result
942 follows. □

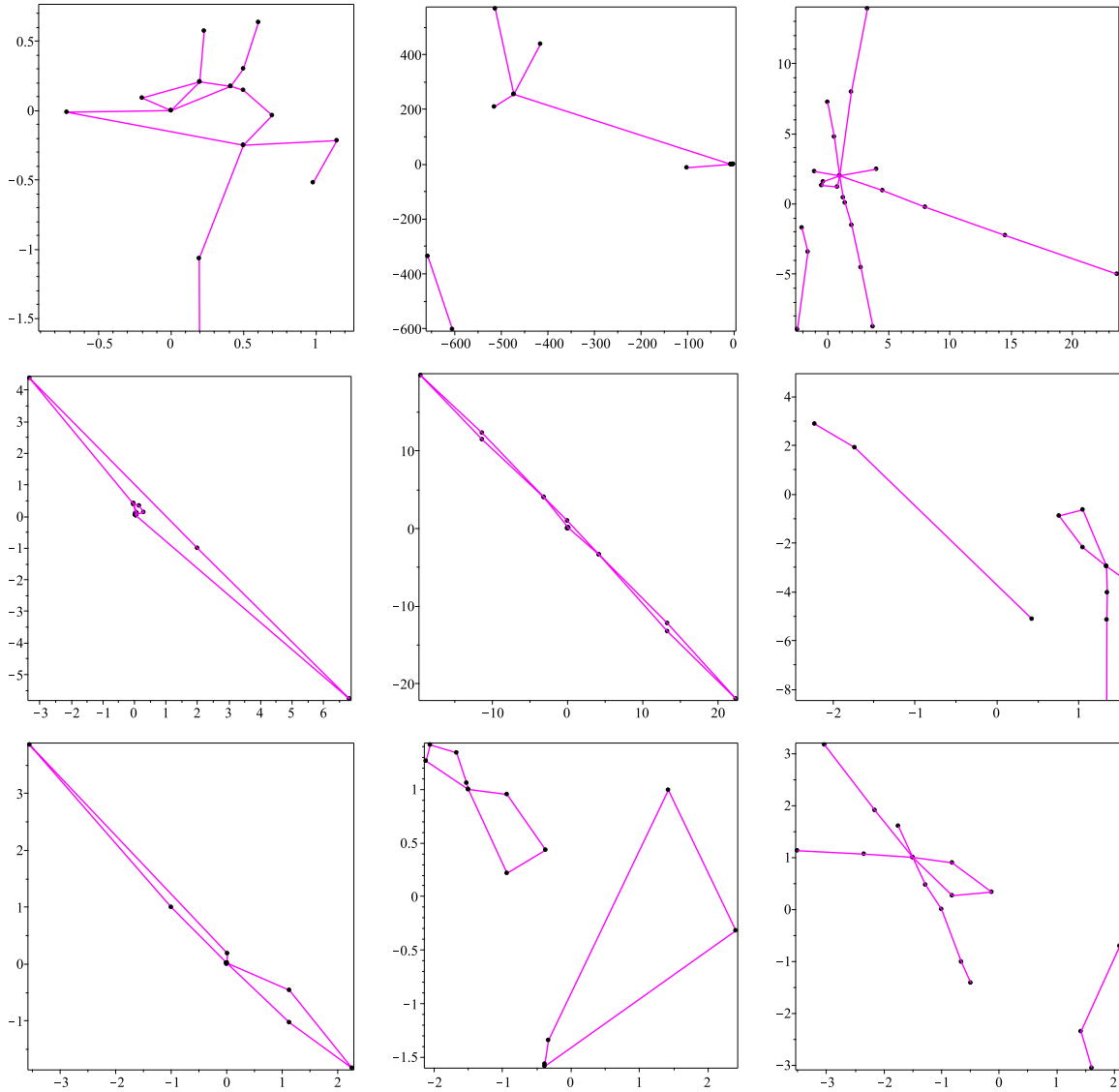


Figure 7: Examples of the 2D algorithm.

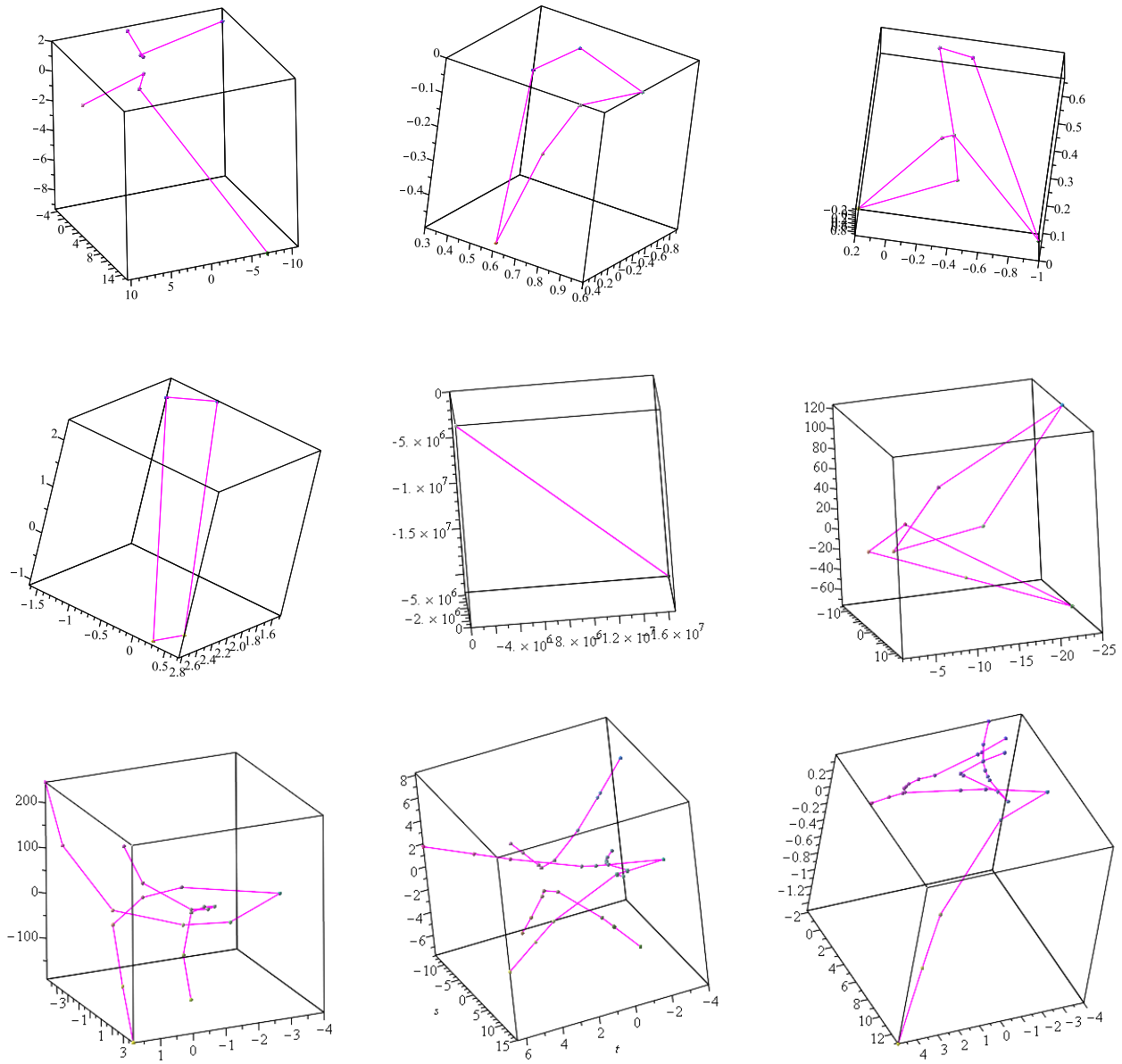


Figure 8: Examples of the 3D algorithm.