Computing the topology of three-dimensional algebraic curves

Grégory Gatelier, Abder Labrouzy, Bernard Mourrain, Jean-Pierre Técourt

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Computing the topology of three-dimensional algebraic curves

G. Gatellier — A. Labrouzy — B. Mourrain — J.P. Técourt

N° 5194

Mai 2004

Thème SYM

ISSN 0249-6399

INRIA
SOPHIA ANTIPOLIS

Computing the topology of three-dimensional algebraic curves

G. Gatellier , A. Labrouzy , B. Mourrain , J.P. Técourt

Thème SYM — Systèmes symboliques
Projet GALAAD

Rapport de recherche n° 5194 — Mai 2004 — 21 pages

Abstract: In this report, we present a new method for computing the topology of curves defined as the intersection of two implicit surfaces. The main ingredients are projection tools, based on resultant constructions and 0-dimensional polynomial system solvers. We describe a lifting method for points on the projection of the curve on a plane, even in the case of multiple preimages on the 3D curve. Reducing the problem to the comparison of coordinates of so-called critical points, we propose an approach which combines control and efficiency. An emphasis in this work is put on the experimental validation of this new method. Examples treated with the tools of the library AXEL* (Algebraic Software-Components for gEometric modeLing) are showing the potential of such techniques.

Key-words: curve, implicit surface, topology, polynomial equation, resultant, projection, critical point, certified computation, symbolic-numeric computation, geometric software

This work is partially supported by GAIA II IST-2001-34919 and ECG IST-2000-26473 European projects

* http://www-sop.inria.fr/galaad/logiciels/axel/

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : +33 4 92 38 77 77 —Télécopie : +33 4 92 38 77 65

Sur le calcul de la topologie de courbes algébriques en dimension 3

Résumé : Dans ce rapport, nous décrivons une nouvelle méthode pour calculer la topologie de courbes définies comme intersection de deux surfaces implicites. Les ingrédients principaux sont des techniques de projection, basées sur des calculs de résultants et des outils de résolutions de systèmes d'équations polynomiales. Nous présentons une méthode de relèvement de points sur la projection de la courbe dans un plan, utilisable même dans le cas de préimages multiples sur la courbe 3D. Ramenant le problème à la comparaison des coordonnées de points dits critiques, nous proposons une approche qui combine certification et efficacité. Une attention particulière est donnée dans ce travail à la validation expérimentale. Des exemples, traités avec les outils de la bibliothèque AXEL[†] (Algebraic Software-Components for gEometric modeLing) montrent tout le potentiel de cette nouvelle approche.

Mots-clés : courbe, surface implicite, topologie, équation polynomiale, résultant, projection, point critique, calcul certifié, lien symbolique-numérique, logiciel géométrique

[†] http://www-sop.inria.fr/galaad/logiciels/axel/

1 Introduction

Numerical modeling plays an increasingly role in fields at the border between data processing and mathematics. This is the case for example in CAD (Computer-aided design, where the objects of a scene or a piece to be built are represented by parameterized curves or surfaces such as NURBS), robotics (problem of the parallel robot, or vision), or molecular biology (rebuilding of a molecule starting from the matrix of the distances between its atoms obtained by NMR). A fundamental operation in this context is the intersection of geometric models, which leads to algebraic questions.

In this paper, we focus on the problem of computing the topology of the intersection of two algebraic surfaces. Such a question is critical in many solid geometry operations, involved in the digital modeling or construction process of shapes. In the case of two parameterized surfaces, in order to reduce to such a situation, we may compute the implicit equation of one of the rational surfaces [4]. This reduces the problem of intersection to the case of an implicit and a parameterized representation, which boils down, by substitution, to the case of a curve defined by an implicit equation in the plane of parameters. Our main concern will be the case of implicit curves, either in the plane or defined by two polynomial equations, in a 3-dimensional space.

This intersection problem received a lot of attention in the past literature. See for instance [9, 19, 16]. Different techniques (subdivision, lattice evaluation, marching methods) have been experimented [18], [10], [1], [13], [12], [19], but they suffer from the problem of certifying the topology of the result.

In this paper, we present a new method to compute the topology of an algebraic curve in 3D, based on an extension of the 2D approach [13], [11]. Our objective is to devise a certified and output-sensitive method, in order to combine control and efficiency. We show that it reduces to the comparison of coordinates of points of intersections of two curves or three surfaces. This task can be fulfilled by using exact methods, such as the one described in [3], [8], which reduces to comparison of roots of univariate problems. Our approach is combining symbolic and numeric techniques, in order to filter the numerical computation. We present preliminary experiments in the library AXEL[1] (Algebraic Software-Components for gEometric modeLing), devoted to algebraic tools for geometric modeling. We are interested in the efficiency and also in the numerical behavior and stability of the method. The experiments are made using the package SYNAPS[2] [6] (SYmbolic and Numeric APplicationS), for the polynomial solvers it provides. We apply in particular solvers, based on algebraic manipulations [20], or resultant constructions. This leads to eigenvalue computations, which are performed by LAPACK subroutines [2]. For more details on the polynomial solving algorithms, we refer to [8].

The main objective being the description of the curve of intersection of two implicit surfaces, the method that we present yields "only" the topology of such a curve, that is a graph of 3D points, connected by segments, with the same topology as the algebraic

[1] `http://www-sop.inria.fr/galaad/logiciels/axel/`
[2] `http://www-sop.inria.fr/galaad/logiciels/synaps/`

curve. Producing a good geometric approximation of the curve, which is the next step of a complete method, will not be considered here. It consists in applying marching techniques on the regular branches of the curve.

In the next section, we will describe quickly the algebraic ingredients that we need. The algorithm will be detailed in section 3 and some implementation topics and experimentations are presented in the last section.

2 Algebraic tools

In this section, we introduce different algebraic tools that will be used later.

2.1 Resultant and projection

For any polynomials $p_1, \ldots, p_k \in \mathbb{R}[z]$, we denoted by $\langle p_1, \ldots, p_k \rangle$ the vector space spanned by these polynomials. Let $\mathbb{R}[z]_d$ be the space of polynomials of degree less or equal to d, with basis $\{1, \ldots, z^d\}$.

Let us consider two univariate polynomials P and Q of $\mathbb{R}[z]$.

$$P = \sum_{k=0}^{p} a_k z^k, \quad Q = \sum_{k=0}^{q} b_k z^k.$$

We need first the following definition:

Definition 2.1 *The Sylvester matrix of P and Q is the matrix of the application*

$$\begin{array}{ccc} \mathbb{R}[z]_{q-1} \oplus \mathbb{R}[z]_{p-1} & \longrightarrow & \mathbb{R}[z]_{p+q-1} \\ (u, v) & & P\,u + Q\,v \end{array}$$

of the form

$$\mathrm{Syl}(P,Q) = \begin{pmatrix} a_p & & & b_q & & \\ a_{p-1} & \ddots & a_p & b_{q-1} & \ddots & b_q \\ \vdots & & a_{p-1} & \vdots & & b_{q-1} \\ a_0 & & \vdots & b_0 & & \vdots \\ & \ddots & a_0 & & \ddots & b_0 \end{pmatrix}$$

Definition 2.2 *Determinantal polynomial. Let $\mathcal{M}$ be a matrix $k \times l$ with $l \leq k$. We define the determinantal polynomial of $\mathcal{M}$:*

$$detpol(\mathcal{M}) = \det(\mathcal{M}_k)\, z^{k-l} + \cdots + \det(\mathcal{M}_l)$$

where $\mathcal{M}_j$ denotes the submatrix of $\mathcal{M}$ consisting of the $l-1$ first rows of $\mathcal{M}$ followed by the j^{th}.

Definition 2.3 *The polynomial subresultant of order i associated to P and Q is:*

$$S_i = detpol(z^{q-i-1}\,P, \ldots, P, z^{p-i-1}\,Q, \ldots, Q) = \sum_{k=0}^{l} S_{i,k} z^k$$

See [21]. These polynomials can be computed efficiently by Sturm-Habicht sequences [3]. Notice that S_0 is the determinant of the Sylvester matrix (i.e the resultant) of P and Q.

We will use the following result:

Proposition 2.4 *The last polynomial S_k associated to P and Q with $S_{k,k} \neq 0$ is the greatest common divisor of P and Q.*

Another important property is:

Proposition 2.5 *The corank of the Sylvester matrix associated to P and Q is the degree of $gcd(P, Q)$.*

Proof. Let $D = gcd(P, Q)$ and $\delta = \deg(D)$. Then we have

$$\text{corank}(\text{Syl}(P, Q)) = p + q - \dim\langle P, z\,P, \ldots, z^{q-1}\,P, \ldots, Q, z\,Q, \ldots, z^{p-1}Q\rangle.$$

As $\dim\langle P, z\,P, \ldots, z^{q-1}\,P, \ldots, Q, z\,Q, \ldots, z^{p-1}Q\rangle = \dim\langle D, z\,D, \ldots, z^{p+q-1-\delta}\,D\rangle$, we deduce that corank(Syl(P,Q)) $= \delta$. □

2.2 Solving 0-dimensional systems

In this section, we are interested in solving polynomial systems of dimension 0 i.e that admit a finite number of (complex) solutions. Different approaches exist to solve such systems [8]. We focus on the algebraic approach that transforms the resolution problem into linear algebra problems.

Here are some notations: $R = \mathbb{R}[x, y, z]$, $f_1 = 0, \ldots, f_m = 0$ with $f_i \in R$, the equations we want to solve, $I = (f_1, \ldots, f_m)$ is the ideal generated by these polynomials, $\mathcal{A} = R/I$ the quotient algebra. We denote by

$$\mathcal{V}(f_1, \ldots, f_m) = \{(x, y, z) \in \mathbb{C}^3, f_i(x, y, z) = 0,\ i = 1, \ldots, m\},$$

the variety of $\mathbb{C}^3$ defined by the equations $f_i(x, y, z) = 0$.

We deduce from the structure of the quotient algebra $\mathcal{A}$, the solutions $\mathcal{V}(I)$, from the following theorem:

Theorem 2.6 *Assume that $\mathcal{V}(I) = \{\xi_1, \ldots, \xi_d\}$. We have:*

- *Let $a \in \mathcal{A}$. The eigenvalues of the operator M_a (and M_a^t) are $a(\xi_1), \ldots, a(\xi_d)$.*
- *The common eigenvectors of $(M_a^t)_{a\in\mathcal{A}}$ are (up to a scalar) $1_{\xi_1}, \ldots, 1_{\xi_d}$ where 1_{ξ_i} is the linear form $1_{\xi_i} : p \longrightarrow p(\xi_i)$.*

This theorem reduces the resolution to a linear algebra problem [8] if we are able to work in $\mathcal{A}$. In order to turn this theorem into an effective method, we have to construct the matrices of multiplication in $\mathcal{A}$. For this purpose, we compute so called normal forms. One way is to use Gröbner bases but for numerical stability, we prefer to use general normal forms [17], [20].
We summarize the main stages of the resolution process, in the following algorithm:

Algorithm 2.7 — Solving 0-dimensional system. *Input:* $I = (f_1, \ldots, f_m)$.

- *Compute a basis of $\mathcal{A}$ and polynomials which yield a normal form reduction modulo I.*
- *Deduce the matrices of multiplication by x, y, z in the basis of $\mathcal{A}$.*
- *Compute simultaneous eigenvectors of M_x^t, M_y^t, M_z^t and the corresponding eigenvalues[8].*

Output: $\mathcal{V}(I) = \{\xi_i \text{(with multiplicity)}, i = 1, \ldots, \dim \mathcal{A}\}$.

3 Topology of algebraic curves

By definition, a three dimensional algebraic curve $\mathcal{C}_{\mathbb{C}} = \mathcal{V}(f_1, \ldots, f_m)$ ($f_i \in \mathbb{R}[x, y, z]$) is an algebraic variety of dimension 1 in $\mathbb{C}^3$. We denote by $I(\mathcal{C}_{\mathbb{C}}) \subset \mathbb{R}[x, y, z]$, the ideal of the curve $\mathcal{C}_{\mathbb{C}}$ (that is the set of polynomials which vanish on $\mathcal{C}_{\mathbb{C}}$) and by $g_1, \ldots, g_s \in \mathbb{R}[x, y, z]$ a set of generators: $I(\mathcal{C}_{\mathbb{C}}) = (g_1, \ldots, g_s)$. By Hilbert's Nullstellensatz [5, 15], we have $I(\mathcal{V}(f_1, \ldots, f_k)) = \sqrt{I} \subset \mathbb{R}[x, y, z]$. It can be proved [7], [15], that 3 polynomials g_1, g_2, g_3 $\in \mathbb{R}[x, y, z]$ are enough to generate $I(\mathcal{C}_{\mathbb{C}})$.

For simplicity reasons, we will consider here that the curve is described as the intersection of two surfaces $P_1(x, y, z) = 0$, $P_2(x, y, z) = 0$, with $P_1, P_2 \in \mathbb{R}[x, y, z]$. We assume that the gcd of P_1 and P_2 in $\mathbb{R}[x, y, z]$ is 1, so that $\mathcal{V}(P_1, P_2) = \mathcal{C}_{\mathbb{C}}$ is of dimension 1, and all its irreducible components are of dimension 1. We are interested in describing the topology of the real part

$$\mathcal{C}_{\mathbb{R}} = \{(x, y, z) \in \mathbb{R}^3,\ P_1(x, y, z) = 0,\ P_2(x, y, z) = 0\},$$

that we will denote hereafter by $\mathcal{C}$.

In this paper, we assume that $I(\mathcal{C}) = (P_1, P_2)$ or equivalently that (P_1, P_2) *is a reduced ideal*: $(P_1, P_2) = \sqrt{(P_1, P_2)}$.

We will not consider examples such as $P_1 = x^2 + y^2 - 1, P_2 = x^2 + y^2 + z^2 - 1$, where $(P_1, P_2) = (x^2 + y^2 - 1, z^2)$ and $I(\mathcal{C}) = (x^2 + y^2 - 1, z)$, so that the curve $\mathcal{C}$ is defined "twice" by the equations $P_1 = 0, P_2 = 0$ (the two surfaces intersect tangently along $\mathcal{C}$). Such a property can be tested by projecting into a generic direction and testing if the equation computed from the resultant of P_1, P_2, is squarefree, or by more general methods such as computing the radical of (P_1, P_2) [14].

The general idea behind the algorithm that we are going to describe is as follows: we use a sweeping plane in a given direction (say parallel to the (y, z) plane) to detect the critical positions where *something* happens. We also compute the positions where something

happens in projection on the (x, y) and (x, z) plane. Then, we connect the points of the curve of $\mathcal{C}$ on these critical planes. This yields a graph of points, connected by segments, with the same topology as the curve $\mathcal{C}$.

3.1 Critical points and generic position

In this section, we precise what we mean by the points where *something* happens. These points will be called hereafter critical points.

Definition 3.1 *Let $I(\mathcal{C}) = (g_1, g_2, \ldots, g_s)$ and let M be the $s \times 3$ Jacobian matrix with rows $\partial_x g_i, \partial_y g_i, \partial_z g_i$.*

- *A point $p \in \mathcal{C}$ is* regular *(or smooth) if the rank of M evaluated at p is 2.*
- *A point $p \in \mathcal{C}$ which is not regular is called* singular.
- *A point $p = (\alpha, \beta, \gamma) \in \mathcal{C}$ is x-*critical *(or critical for the projection on the x-axis) if the curve $\mathcal{C}$ is tangent at this point to a plane parallel to the (y, z)-plane i.e the multiplicity of intersection of the plane with $I(\mathcal{C})$ at p is greater or equal to 2. The corresponding α is called a x-*critical value.

A similar definition applies for the orthogonal projection onto the y and z axis or onto any line of the space. Notice that a singular point is critical for any direction of projection.

If $I(\mathcal{C}) = (P_1, P_2)$, then the x-critical points are the solutions of the system

$$P_1(x, y, z) = 0,\ P_2(x, y, z) = 0,\ (\partial_y P_1 \partial_z P_2 - \partial_y P_2 \partial_z P_1)(x, y, z) = 0. \qquad (1)$$

In the case of a planar curve defined by $P(x, y) = z = 0$, with $P(x, y)$ squarefree so that $I(\mathcal{C}) = (P(x, y), z)$, this yields the following definitions: a point (α, β)

- is *singular* if $P(\alpha, \beta) = \partial_x P(\alpha, \beta) = \partial_y P(\alpha, \beta) = 0$.
- is *x-critical* if $P(\alpha, \beta) = \partial_y P(\alpha, \beta) = 0$.

This allows us to describe the genericity condition that we require for the curve $\mathcal{C}$, in order to be able to apply the algorithm:

Definition 3.2 *Let*

$$N_x(\alpha) \quad = \quad \#\{(\beta, \gamma) \in \mathbb{R}^2 \ st.\ (\alpha, \beta, \gamma) \text{ is a x} - \text{critical point of } \mathcal{C}\}.$$

We say that $\mathcal{C}$ is in a generic position for the x-direction, if

- *$\forall \alpha \in \mathbb{R},\ N_x(\alpha) \leqslant 1$, and*
- *there is no asymptotic direction of $\mathcal{C}$ parallel to the (y, z)-plane.*

We will show that by a random change of variables, the curve can be put in a generic position. In practice, instead of changing the variables, we may choose a random direction for the sweeping plane.

3.2 The projected curves

The algorithm that we are going to describe, uses the singular points of the projection of $\mathcal{C}$ onto the (x,y) and (x,z)-planes. We denote by $\mathcal{C}'$ (resp. $\mathcal{C}''$) the projection of the curve $\mathcal{C}$ onto the (x,y) (resp. (x,z))-plane. The equation of the curve $\mathcal{C}'$ is obtained as follows. We decompose the polynomials P_1, P_2 in terms of the variable z:

$$\begin{aligned} P_1(x,y,z) &= a_{d_1}(x,y)z^{d_1} + ... + a_0(x,y) \\ P_2(x,y,z) &= b_{d_2}(x,y)z^{d_2} + ... + b_0(x,y) \end{aligned}$$

with $\mathrm{a}_{d_1}(x,y) \neq 0$ and $\mathrm{b}_{d_2}(x,y) \neq 0$. Then, the resultant polynomial

$$G(x,y) = \mathrm{Res}_z(P_1, P_2)$$

vanishes on the projection of the curve $\mathcal{C}$ on the plane (x,y). Conversely, by the resultant theorem [8], $G(x,y) = 0$ defines exactly the projection $\mathcal{C}'$ of the curve $\mathcal{C}$ if $a_{d_1}(x,y)$ and $b_{d_2}(x,y)$ do not vanish simultaneously on a component of dimension 1 of $\mathcal{C}'$, that is, if the gcd $c(x,y)$ of $a_{d_1}(x,y)$ and $b_{d_2}(x,y)$ in $\mathbb{R}[x,y]$ is a 1. If it's not the case, G is a non-trivial multiple of the implicit equation of $\mathcal{C}'$. Such a situation can be avoided, by a linear change of variables. Nevertheless, since the critical points of the curve defined by $G(x,y) = 0$ contains the critical points of $\mathcal{C}'$, we will see hereafter that this change of variables is not necessary.

Notice, that $G(x,y)$ is not necessarily a squarefree polynomial. Consider for instance the case $P_1 = x^2 + y^2 - 1, P_2 = x^2 + y^2 + z^2 - 2$, where $g(x,y) = (x^2 + y^2 - 1)^2$. In this case, there are generically two (complex) points of $\mathcal{C}$ above a point of $\mathcal{C}'$.

We can easily compute the gcd of $G(x,y)$ and $\partial_y G(x,y)$ (using proposition 2.4), in order to get the squarefree part $g(x,y) = G(x,y)/gcd(G(x,y), \partial_y G(x,y))$ of $G(x,y)$.

Similarly, for the projection $\mathcal{C}''$ of $\mathcal{C}$ on the (x,z)-plane, we compute

$$H(x,z) = \mathrm{Res}_y(P_1, P_2)$$

and its square-free part $h(x,z)$ from the gcd of $H(x,z)$ and $\partial_z H(x,z)$. The equation $h(x,z) = 0$ defines a curve which is exactly $\mathcal{C}''$, if the gcd of the leading components of P_1, P_2 in y is 1. Its set of singular points contains those of $\mathcal{C}''$.

In order to analyze locally the projection of the curve $\mathcal{C}$, we recall the following definition:

Definition 3.3 *[22] Let X be an algebraic subset of $\mathbb{R}^n$ and let p be a point of X. The tangent cone at p to X is the set of points u in $\mathbb{R}^n$ such that there exists a sequence of points x_k of X converging to p and a sequence of real numbers t_k such that $\lim_{k\to+\infty} t_k(x_k - p) = u$.*

Notice, that at a smooth point of $\mathcal{C}$, the tangent cone is a line.

Proposition 3.4 *Let $p' = (\alpha, \beta)$ be a x-critical point of $\mathcal{C}'$, which is not singular. Then α is the x-coordinate of a x-critical point of $\mathcal{C}$.*

Proof. Let V be the set of points $p \in \mathcal{C}$, which project onto p'. From the previous definition, we directly deduce that the projection of the tangent cone at $p \in \mathcal{C}$ is contained in the tangent cone of the projection of p. Thus the tangent cone of $\mathcal{C}'$ at $p' = (\alpha, \beta)$ contains the projection of the tangent cones of the points $p \in V$. Since p' is regular, its tangent cone is a line parallel to the y direction. Therefore, the tangent cones of the points $p \in V$ are in the plane $x - \alpha = 0$, parallel to the plane (y, z). This implies that the intersection of $\mathcal{C}$ with the plane $x - \alpha = 0$ contains a point of multiplicity ≥ 2, that is a x-critical point. In other words, α is the x-coordinate of a x-critical point of $\mathcal{C}$. □

3.3 Lifting a point of $\mathcal{C}'$

The problem we want to tackle here is the following: Assume we are given two surfaces defined by two implicit equations $P_1 = 0$ and $P_2 = 0$. Let us consider the projection of the curve of intersection of the two surfaces on the (x, y)-plane. Starting from a point (x_0, y_0) of the projected curve, how can we find the z-coordinate of the point(s) above (x_0, y_0) ?

We note $P(z) = P_1(x_0, y_0, z)$, $Q(z) = P_2(x_0, y_0, z)$ and $p = \deg(P)$, $q = \deg(Q)$. Consider the Sylvester submatrix $\mathrm{Syl}_1(x_0, y_0)$ of the application

$$\begin{array}{ccc} \mathbb{R}[z]_{q-2} \oplus \mathbb{R}[z]_{p-2} & \longrightarrow & \mathbb{R}[z]_{p+q-2} \\ (u, v) & \mapsto & P\,u + Q\,v \end{array}$$

If ξ is a common root of P and Q then $(1, \xi, \ldots, \xi^{p+q-2})$ is in the kernel of the transpose of $\mathrm{Syl}_1(x_0, y_0)$. If we assume that $\mathrm{Syl}_1(x_0, y_0)$ is of maximal rank, and if Δ_i denotes the minor of $\mathrm{Syl}_1(x_0, y_0)$ obtained by removing the row i, then the (non-zero) vector $[\Delta_1, -\Delta_2, \ldots, (-1)^{p+q-1}\Delta_{p+q-1}]$ is in the kernel of the transpose of $\mathrm{Syl}_1(x_0, y_0)$. Thus $(1, \xi, \ldots, \xi^{p+q-2})$ and $[\Delta_1, -\Delta_2, \ldots, (-1)^{p+q-1}\Delta_{p+q-2}]$ are linearly dependent. We deduce that $\xi = -\frac{\Delta_{p+q-1}}{\Delta_{p+q-2}} = -\frac{S_{1,0}(x_0,y_0)}{S_{1,1}(x_0,y_0)}$.

This method allows us to lift a point on $\mathcal{C}$, if there is only one point above (x_0, y_0), but it can be generalized when there are several points above. This generalization is closely related to the subresultant construction of univariate polynomials [21]. Here we want to exploit linear algebra tools from a numerical perspective. The aim is to make the matrix of multiplication by z in the quotient algebra $\mathbb{R}[z]/(P_1(x_0, y_0, z), P_2(x_0, y_0, z))$ appear, in order to compute its eigenvalues which yields z-coordinate of the points above (x_0, y_0) [8].

We proceed as follows: Given a point (x_0, y_0) of the projected curve $\mathcal{C}'$, we construct the Sylvester matrix associated to $P(z)$, $Q(z)$. By construction, the columns of this matrix are $P, z\,P, \ldots, z^{q-1}\,P, Q, z\,Q, \ldots, z^{p-1}\,Q$, written in the basis $1, z, \ldots, z^{p+q-1}$. Assume that the kernel of the transposed Sylvester matrix $\mathrm{Syl}(x_0, y_0)$ has dimension d and is generated by $\Lambda_1, \ldots, \Lambda_d$.

By transposition, we can interpret the Λ_i $(i = 1 \ldots d)$ as linear forms over $\mathbb{K}_{p+q-1}[z]$ vanishing on $P, z\,P, \ldots, z^{q-1}\,P, Q, z\,Q, \ldots, z^{p-1}\,Q$. We can extend the Λ_i over $\mathbb{R}[z]$, considering that these forms vanish over all the ideal generated by P and Q. So they can be considered as elements of the dual of $\mathcal{A} = \mathbb{R}[z]/(P(z), Q(z))$. As the linear forms Λ_i are independent, they also form a basis of this dual space. The coefficients of Λ_i in the dual basis $(1^*, \ldots, (z^{d-1})^*)$

of the monomial basis $\{1, z, \ldots, z^{d-1}\}$ of $\mathcal{A}$ are $[\Lambda_i(1), \Lambda_i(z), \ldots, \Lambda_i(z^{d-1})]$. By definition of the transposed operator, for any $a \in \mathcal{A}$, $M^t(\Lambda_i)(a) = \Lambda_i(M_z(a)) = \Lambda_i(z\,a)$. Thus we have the relation:

$$\begin{pmatrix} \Lambda_1(z) & \ldots & \Lambda_d(z) \\ \vdots & & \vdots \\ \Lambda_1(z^d) & \ldots & \Lambda_d(z^d) \end{pmatrix} = M_z^t \begin{pmatrix} \Lambda_1(1) & \ldots & \Lambda_d(1) \\ \vdots & & \vdots \\ \Lambda_1(z^{d-1}) & \ldots & \Lambda_d(z^{d-1}) \end{pmatrix}$$

where M_z is the operator of multiplication by z in $\mathbb{R}[z]/(P(z), Q(z))$. As $d = \dim \ker(\mathrm{Syl}(x_0, y_0)) = \dim \mathcal{A}$, and as $(1, z, ..., z^{d-1})$ form a basis of the quotient space, the matrix

$$\begin{pmatrix} \Lambda_1(1) & \ldots & \Lambda_d(1) \\ \vdots & & \vdots \\ \Lambda_1(z^{d-1}) & \ldots & \Lambda_d(z^{d-1}) \end{pmatrix}$$

is invertible. We deduce that computing the generalized eigenvalues of the previous matrices yields the eigenvalues of the operator M_z of multiplication by z in $\mathcal{A}$, that is the z-coordinate of the points above (x_0, y_0).

We summarize the algorithm here:

Algorithm 3.5 — Lifting the projection.

- *Compute the Sylvester matrix* $S = \mathrm{Syl}(x_0, y_0)$.
- *Compute a basis* $\Lambda_1, \ldots, \Lambda_d$ *of the kernel of* S^t.
- *Extract the submatrix* A_0 *of the coordinates of* $\Lambda_1, \ldots, \Lambda_d$ *corresponding to the evaluations in* $1, \ldots, z^{d-1}$.
- *Extract the submatrix* A_1 *of the coordinates of* $\Lambda_1, \ldots, \Lambda_d$ *corresponding to the evaluations in* $z, \ldots, z^d$.
- *Compute the generalized eigenvalues of* A_1 *and* A_0 *and output the corresponding* z*-coordinates of the point above* (x_0, y_0).

The last step can be replaced by the computation of $\det(A_1 - z\,A_0)$ *and an univariate root finding step.*

3.4 Computing points of $\mathcal{C}$ at critical values

In this section, we are going to describe how we check the genericity condition and how we compute a finite set of points, which will allow us to deduce the topology of $\mathcal{C}$.

First, we check that there is no asymptotic direction parallel to the (y, z) -plane, by testing if the curve $\mathcal{C}$ has a point at infinity in the plane $x = 0$. This is done by checking if the system

$$\frac{P_1^\top}{\Delta}(0, y, z) = \frac{P_2^\top}{\Delta}(0, y, z) = 0$$

has a non-trivial solution, where $P^\top$ is the homogeneous component of highest degree of a polynomial P and $\Delta = \gcd(P_1^\top, P_2^\top)$. It reduces to computing the projective resultant of these two homogeneous polynomials. Since the number of asymptotic directions of $\mathcal{C}$ is finite, by a generic linear change of variables, we can avoid the cases where $\mathcal{C}$ has an asymptotic direction parallel to the (y,z) plane.

Next, we compute the x-critical points of $\mathcal{C}$ by solving the system (1), using algorithm 2.7. This computation allows us to check that the system is zero-dimensional and that the x-coordinates of the real solutions are distinct. If this is not the case, we perform a generic change of coordinates.

The cases for which we have to do a change of coordinates are those where a component of $\mathcal{C}$ is in a plane parallel to (y,z) or where a plane parallel to (y,z) is tangent to $\mathcal{C}$ in two distinct points. Such cases are avoided by a generic change of coordinates.

We denote by $\Sigma_0 = \{\sigma_1^0, \ldots, \sigma_{k_0}^0\}$ the x-coordinates of the x-critical points: $\sigma_1^0 < \cdots < \sigma_{k_0}^0$.

Next, we compute the singular points of $\mathcal{C}'$ as (a subset of) the real solutions of the system

$$g(x,y) = 0, \partial_x g(x,y) = 0, \partial_y g(x,y) = 0, \tag{2}$$

and of $\mathcal{C}''$, as (a subset of) the real solutions of

$$h(x,z) = 0, \partial_x h(x,z) = 0, \partial_z h(x,z) = 0. \tag{3}$$

We denote by $\Sigma_1 = \{\sigma_1^1, \ldots, \sigma_{k_1}^1\}$ the x-coordinates of these singular points: $\sigma_1^1 < \cdots < \sigma_{k_1}^1$.

Let us denote by $\Sigma = \Sigma_0 \cup \Sigma_1 = \{\sigma_1, \ldots, \sigma_l\}$ (with $\sigma_1 < \cdots < \sigma_l$) the sequence of all the x-coordinates computed so far.

An important property of the projected curves $\mathcal{C}'$ and $\mathcal{C}''$, that will be used in the algorithm, is the following:

Proposition 3.6 *The arcs of the curve $\mathcal{C}'$ (resp. $\mathcal{C}''$) above $]\sigma_i, \sigma_{i+1}[$ do not intersect.*

Proof. By definition, the arcs of $\mathcal{C}'$ above $]\sigma_i, \sigma_{i+1}[$ can only intersect at the x-critical points of $\mathcal{C}'$. Let σ be the x-coordinate of such a point. According to proposition 3.4, σ is either

- the x-coordinate of a x-critical point of $\mathcal{C}$ ($\in \Sigma_0$),
- or the x-coordinate of a singular point of $\mathcal{C}'$ ($\in \Sigma_1$).

Thus, $\sigma \in \Sigma$ and $\sigma \notin]\sigma_i, \sigma_{i+1}[$, which implies that the arcs of $\mathcal{C}'$ above $]\sigma_i, \sigma_{i+1}[$ do not intersect. The same proof applies for $\mathcal{C}''$. □

3.5 Connecting the branches

The approach that we are going to describe now for the branch connection, can be seen as an extension of the approach of [13], [11] to the three-dimensional case.

The previous step yields a sequence of strictly increasing values

$$\Sigma = \{\sigma_1, \ldots, \sigma_l\},$$

such that above $]\sigma_i, \sigma_{i+1}[$, the branches of $\mathcal{C}$ are smooth and the arcs of $\mathcal{C}'$, $\mathcal{C}''$ do not intersect. We will use this property to connect the points of $\mathcal{C}$ above the values σ_i. Notice that proposition 3.6 is still true if we refine the sequence $\sigma_1, \ldots, \sigma_l$. In particular, it is valid if we consider the x-coordinates of the singular points of a curve, defined by a multiple of the equation of $\mathcal{C}'$ (resp. $\mathcal{C}''$). It is also valid, if we insert new values in between these critical values: $\delta_0 < \sigma_1 < \mu_1 < \cdots < \sigma_l < \delta_1$, where $\mu_i := \frac{\sigma_i + \sigma_{i+1}}{2}$ for $i = 0, \ldots, l-1$, and δ_0, δ_1 are any value such that $]\delta_0, \delta_1[$ contains Σ. We denote by

$$\alpha_0 < \cdots < \alpha_m$$

this new refined sequence of values and by L_i, the set of points on $\mathcal{C}$ above α_i, for $i = 0, \ldots, m$. These points are computed, either

- by substituting $x = \alpha_i$ and solving the 2-dimensional system $P_1(\alpha_i, y, z) = 0$, $P_2(\alpha_i, y, z) = 0$.
- or by computing the points of $\mathcal{C}'$ above α_i and by lifting them to $\mathcal{C}$ (algorithm 3.5).

This construction implies the following lemma, which is used in the next theorem, in order to describe how the computed points have to be connected:

Lemma 3.7 *Two distinct points of a regular section of $\mathcal{C}$ with the same y-coordinate (resp. z-coordinate) are connected to two points of the next section, with the same y-coordinate (resp. z-coordinate) or to a critical point.*

Proof. We denote by L the regular section at $x = \alpha$ of $\mathcal{C}$ and by L' the next section, at $x = \alpha'$. Let $p = (\alpha, \beta, \gamma) \in L, q = (\alpha, \beta, \delta) \in L$ with $\gamma \neq \delta$. They are connected by $\mathcal{C}$ respectively to $p' = (\alpha', \beta', \gamma'), q' = (\alpha', \epsilon', \delta') \in L'$. Assume that $\beta' \neq \epsilon'$. Then there are two arcs of the projection $\mathcal{C}'$, connecting (α, β) to (α', β') and to (α', ϵ'), above $[\alpha, \alpha']$. This implies that there exists a point $r \in \mathcal{C}'$ with $x(r) \in [\alpha, \alpha'[$ belonging to 3 branches. Such a point cannot be regular, in contradiction with the fact that $\mathcal{C}'$ is smooth above $[\alpha, \alpha'[$. Exchanging the role of y and z, we get the same property for the z-coordinates. □

Theorem 3.8 *Under the genericity condition of definition 3.2, the curve $\mathcal{C}$ can connect the points L_i to the points L_{i+1}, only in one way.*

Proof. By construction, for any pair (α_i, α_{i+1}), at least one of the two values is not in Σ. Let us assume that $\alpha_i \notin \Sigma$ and $\alpha_{i+1} \in \Sigma$ (the treatment of the other possibilities being symmetric). To simplify the notations, let $L = L_i \subset \mathcal{C}$ and $L' = L_{i+1} \subset \mathcal{C}$.

By the genericity assumption, L' contains at most one x-critical point c of $\mathcal{C}$. Since $\alpha_i \notin \Sigma$, each point in L is regular. Moreover, by construction, the arcs of $\mathcal{C}$ above $]\alpha_i, \alpha_{i+1}[$ have no x-critical point. Since there is no asymptotic direction of $\mathcal{C}$ in the (y, z)-direction,

by the implicit function theorem, a (regular) point $p \in L$ is connected by an arc of $\mathcal{C}$, to a point of L'. Conversely, each regular point of L' is connected by an arc of $\mathcal{C}$ to a single point of L, which implies that $|L| \geq |L'|$.

We are going to prove by induction on $|L'|$ that there is a unique way to connect the points of L to the points of L', if the arcs of the (x, y) and (x, z) projections of $\mathcal{C}$ do not intersect above $]\alpha_i, \alpha_{i+1}[$.

If $|L'| = 1$, the curve connects any point of L to the unique element of L'.

Let us assume that the induction hypothesis is true for the cases where $|L'| < |L_{i+1}|$. Let q' be the greatest point of L', for the lexicographic order with $x > y > z$ and let V' be the set of points of L' which y-coordinate is $y(q')$. Let s be the cardinal of V'.

Assume first that $c \notin V'$. This implies that the points of V' are regular. We denote by V the set of s greatest points of L for the lexicographic order with $x > y > z$.

We denote by p the point of V with the greatest y-coordinate among those with greatest z-coordinate. We are going to proof that $\mathcal{C}$ has to connect $p \in L$ and $q' \in L'$.

Assume the converse, so that we have $p \in L$ connected to $p' \in L'$ and $q \in L$ connected to $q' \in L'$, with $p \neq q$, $p' \neq q'$. We consider the following possible cases:

1. $q \in V, p' \in V'$. Then we have $z(p') < z(q')$ (since q' is the greatest point of V' for the lexicographic ordering). From lemma 3.7, we deduce that $z(p) \neq z(q)$ and as p has the greatest z-coordinate of V, we have $z(q) < z(p)$. This implies that the projection on the (x, z)-plane of the arcs of $\mathcal{C}$ connecting p to p' and q to q' intersect. It contradicts the hypothesis that $\mathcal{C}''$ is smooth above $]\alpha_i, \alpha_{i+1}[$.

2. $q \notin V, p' \in V'$. Since every (regular) point of V' is connected to a single point in L and $|V| = |V'|$, there exists a point r of V connected to a point $r' \notin V'$. By definition of V', we have $y(r') < y(q')$, thus by lemma 3.7, we have $y(r) \neq y(q)$ and as the y-coordinate of the points not in V are smaller that those in V, we have $y(r) > y(q)$. This implies that the projection on the (x, y)-plane of the arcs of $\mathcal{C}$ connecting r to r' and q to q' intersect. This contradicts the hypothesis that $\mathcal{C}'$ is smooth above $]\alpha_i, \alpha_{i+1}[$.

3. $q \in V, p' \notin V'$. Then there exists a point r' of V' connected to a point $r \notin V$. By definition of V', we have $y(r') > y(p')$, thus by lemma 3.7 we deduce $y(r) \neq y(p)$. As $r \notin V$ and as the y-coordinate of the points not in V are smaller that those in V, we have $y(r) < y(p)$. It leads to another contradiction, since $\mathcal{C}'$ is smooth above $]\alpha_i, \alpha_{i+1}[$.

4. $q \notin V, p' \notin V'$. As $p' \notin V'$, we have $y(p') < y(q')$, thus by lemma 3.7, $y(p) \neq y(q)$. As $q \notin V$ and as the y-coordinate of the points not in V are smaller that those in V, we have $y(q) < y(p)$.It leads to another contradiction, since $\mathcal{C}'$ is smooth above $]\alpha_i, \alpha_{i+1}[$.

In all these cases, we obtain a contradiction. Thus p and q' have to be connected by an arc of $\mathcal{C}$, above $[\alpha_i, \alpha_{i+1}]$. Removing these points respectively from L and L', we apply the induction hypothesis to proof the result.

Suppose now that c is in V' but not in the set W' of points with the same y-coordinate as the lowest point of L'. We apply the same proof, replacing greatest by smallest in the previous constructions.

The last case, which remains to be treated, is the case where $L' = V' = W'$ is the set of points with the same y-coordinate as the x-critical point c. We define p and q', similarly, as the points of L and L', with greatest z-coordinate.

If $q' \neq c$, then p has to be connected to q'. Otherwise $p \in L$ is connected to $p' \in L'$ and $q \in L$ connected to $q' \in L'$, with $p \neq q$, $p' \neq q'$. Then we have $z(p') < z(q')$. By lemma 3.7 we deduce $z(p) \neq z(q)$ and as p has the greatest z-coordinate: $z(p) > z(q)$, which contradicts the fact that $\mathcal{C}''$ is smooth above $]\alpha_i, \alpha_{i+1}[$. Thus the curve $\mathcal{C}$ connects p to q'. Removing these points respectively from L and L', we apply the induction hypothesis to conclude.

If $q' = c$ and the point with the smallest z-coordinate in L' is not c, we apply the same construction, replacing greatest by smallest.

The remaining case is when $|L'| = 1$, which has already been treated. □

To summarize, the connection of the branches from one plane section of $\mathcal{C}$ to the next one, is performed as follows:

Algorithm 3.9 — Connecting the branches.
If there is no x-critical point in L_i and possibly a x-critical point c of $\mathcal{C}$ in L_{i+1}, do the following:

1. *Decompose L_{i+1} into the subsets $V'_1, \ldots, V'_k$ of the points with the same y-coordinate, listed by increasing y. Let $s_j = |V'_j|$.*
2. *Compute the index j_0 such that $c \in V'_{j_0}$. Decompose L_i into the subsets $V_1, \ldots, V_k$ in the following way:*
   - *For $j > j_0$, V_j is the set of s_j greatest points for the lexicographic order with $x > y > z$, among $L_i - \cup_{l>j} V_l$.*
   - *For $j < j_0$, V_j is the set of s_j smallest points for the lexicographic order, among $L_i - \cup_{l<j} V_l$.*
   - *V_{j_0} is the remaining set of points $L_i - \cup_{l \neq j_0} V_l$.*
3. *For $j \neq j_0$ connect the points of V_j to the points of V'_j, according to there z-coordinates, by segments.*
4. *For $j = j_0$, let A'_{j_0} (resp. B'_{j_0}) be the set of regular points of V'_{j_0}, with z-coordinate $< z(c)$ (resp. $> z(c)$).*
   - *Connect the $|A'_{j_0}|$ points of smallest z-coordinates in V_{j_0} to the points in A'_{j_0}, according to their z-coordinate, by segments.*
   - *Connect the $|B'_{j_0}|$ points of greatest z-coordinates in V_{j_0} to the points in B'_{j_0}, according to their z-coordinate, by segments.*
   - *Connect the remaining points in V_{j_0} to c, by segments.*

If there is a x-critical point of $\mathcal{C}$ in L_i, exchange the role of L_i and L_{i+1} in the previous steps.

Proposition 3.10 *Assume that we are in a generic position. Then the topology of the curve above the segment* $[\alpha_i, \alpha_{i+1}]$ *is the same as the set of segments produced by the algorithm 3.9.*

Proof. Since we are in a generic position, by theorem 3.8, the algorithm 3.9 produces the only way the arcs of the curve $\mathcal{C}$ above $[\alpha_i, \alpha_{i+1}]$ connect the points of L_i to the points of L_{i+1}. Since the algorithm only involves the coordinates of the end points, the projection of these segments on the (x, y) and (x, z) planes either coincide or do not intersect above $]\alpha_i, \alpha_{i+1}[$. Consequently, the curve $\mathcal{C}$ above $[\alpha_i, \alpha_{i+1}]$ is homeomorph to the set of segments joining the corresponding end points. □

3.6 The algorithm

We summarize the complete algorithm below:

Algorithm 3.11 — Representation of the curve $\mathcal{C}$ defined by $P_1(x, y, z) = P_2(x, y, z) = 0$.
Input: polynomials $P_1(x, y, z), P_2(x, y, z)$.

- *Compute the x-critical points of $\mathcal{C}$ and their x-coordinates $\Sigma := \{\sigma_1^0, \ldots, \sigma_k^0\}$ with $\sigma_1^0 < \cdots < \sigma_k^0$.*
- *Check the generic position; If the curve is not in a generic position, apply a random change of variables and restart from the first step.*
- *Compute the square-free part $g(x, y)$ of $\mathrm{Res}_z(P_1, P_2)$.*
- *Compute the square-free part $h(x, z)$ of $\mathrm{Res}_y(P_1, P_2)$.*
- *Compute the singular points of the curves $g(x, y) = 0$ and $h(x, z) = 0$ and insert their x-coordinate in Σ.*
- *Compute the $\mu_i, \delta_0, \delta_1$ and the ordered sequence $\alpha_1 < \cdots < \alpha_l$. Above each α_i for $i = 1, \ldots, l$, compute the set of points L_i on the curve $\mathcal{C}$.*
- *For each $i = 0, \ldots, l-1$, connect the points L_i to those of L_{i+1} by algorithm 3.9.*

Output: the graph of 3D points connected by segments, with the same topology as the curve $\mathcal{C}$.

Remark 3.12 *This algorithm can be easily adapted to the computation of the topology of $\mathcal{C}$ in a box (resp. bounded domain), by considering the points on the border of the box (resp. domain) as x-critical points.*

Remark 3.13 *By a generic change of variables, the set of x-coordinates of the x-critical points of $\mathcal{C}'$ will contain those of $\mathcal{C}$ and the resolution of the system (1) can be replaced by the computation of the x-critical points of $\mathcal{C}'$ and by a lifting operation on $\mathcal{C}$. This allows us to treat unreduced curves, such that $I(\mathcal{C}) \neq (P_1, P_2)$, by using only the squarefree part of $G(x, y)$ and $H(x, z)$. However the verification, a posteriori, of the correctness of the result is more delicate.*

4 Implementation and experiments

The previous algorithm has been implemented in the C++ library called AXEL[3] (Algebraic Software-Components for gEometric modeLing), where classes for implicit curves and surfaces are available:

```
namespace implicit
{
  template<class C, class R> curve2d<C,R=MPol<C> >;
  template<class C, class R>
      curve3d<C,R=shared_object<std::vector<MPol<C> > > >;
  template<class C, class R> surface<C,R=MPol<C> >;
}
```

where `C` is the type of coefficient and `R` is the internal representation used to store the object. In the case of a planar curve `curve2d`, the default value is a bivariate polynomial `MPol<C>` from the SYNAPS[4] library. For a 3D curve, the default value is a vector of multivariate polynomials. For a 3D surface, the default value is a multivariate polynomial `MPol<C>` from the SYNAPS library.

Since the algorithm depends heavily on the algebraic solver used to recover the critical points of $\mathcal{C}$, we parameterize the implementation as

```
template <class M>
class Projection
{
 ...
 template <class G, class Surface>
 void topology(G & graph, const Surface & s0, const Surface & s1);

 template <class G, class Curve3d>
 void topology(G & graph, const Curve3d & c0);
 ...
}
```

[3] `http://www-sop.inria.fr/galaad/logiciels/axel/`
[4] `http://www-sop.inria.fr/galaad/logiciels/synaps/`

where the parameter `M` is the type of method used to solve the 0-dimensional systems. The class `Projection` represents the type of method that we used, to compute the topology of the curve.

Our tests are based on solvers provided by the SYNAPS library, such as `Newmac` (see [20]). Here is an illustration of the way it can be used:

```
...
typedef double coef;
MPol<coef> P=..., Q= ...;
vector<MPol<coef> > v; v.push_back(P); v.push_back(Q);
implicit::curve3d c(v);

affine::point_graph<coef> g;

Projection<Newmac<coef> > Method;
Method.topology(g,c);
...
```

Other examples which are currently under test are:

```
Projection<Newmac<QQ> >().topology(g,c);
Projection<Sylvester<double> >().topology(g,c);
Projection<Subdivision<double> >().topology(g,c);
```

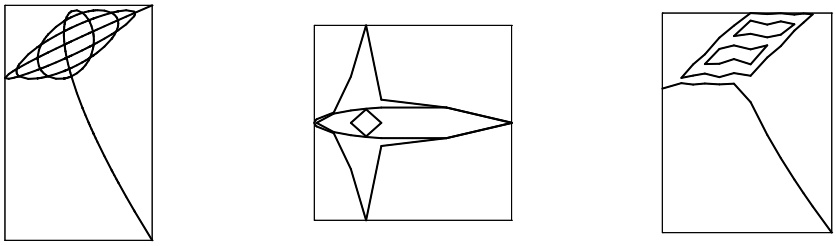
Notice that the solutions given from equations (1) are computed numerically, and we have to sort the lists of critical values to check the generic position of the curve and the points in the lists L_i. For this purpose, we introduce two thresholds ϵ_x and ϵ_y which are the precision on the x-coordinate and the y-coordinate. We have $\epsilon_y \gg \epsilon_x$ because the computation of the y-coordinate has already noisy input (the x-coordinate with a precision ϵ_x). Concretely, this means that given a curve $\mathcal{C}$ we are able to compute correctly the topology, if two critical points are separated at least by $2\epsilon_x$ on the x-coordinate, and two points on $\mathcal{C}$ with the same x-coordinate are separated by ϵ_y at least.

In the cases where we are not able to distinguish within the precision ϵ_x, ϵ_y, two strategies can be applied. Either we use an exact method for representing the solution of the corresponding polynomial system, assuming exact input. Or we consider input polynomial with approximate coefficients and we identify the x-points which are within the prescribed precision. This is what has been experimented.

We run a non-optimized implementation of the algorithm, on cases case where the resultant does not define twice or more the projected curves. The solver that we use is the one of Ph. Trébuchet [20], giving the biggest precision for the smallest $\epsilon_x, \epsilon_y, \epsilon_z$ (about 10^{-6}), compared with the resultant solvers. Further experiments are required to analyze the behavior of such solvers and to compare them correctly, in the context of the topology computation problem. The experimentations have been performed on a Pentium 2Ghz workstation.

4.1 Examples of plane curves

The algorithm that we describe, can be easily specialized to the planar curves. Here are some illustrations of it. The topology of the curve is represented by a graph of 2D points. The time needed to compute this graph of points is given in seconds.



(a) (b) (c)

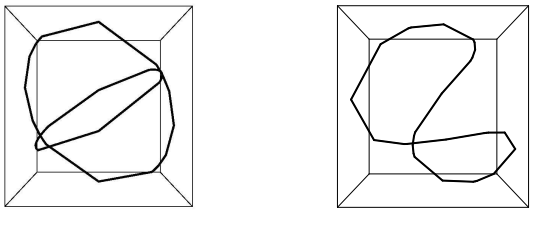
a) $P = -y^8 + x^7 - 7x^6y + 21x^5y^2 - 35x^4y^3 + 35x^3y^4 - 21x^2y^5 + 7xy^6 - y^7 + 8y^6 - 7x^5 + 35x^4y - 70x^3y^2 + 70x^2y^3 - 35xy^4 + 7y^5 - 20y^4 + 14x^3 - 42x^2y + 42xy^2 - 14y^3 + 16y^2 - 7x + 7y - 2$
time: 0.77s

b) $P = 35.9x^6 + 2589.4x^4y^2 + 46728x^2y^4 + 1296y^6 + 217x^5 + 15588x^3y^2 + 2.7994e+05xy^4 - 2303.9x^4 - 72774x^2y^2 + 3.7066e+05y^4 - 15583x^3 - 5.5969e+05xy^2 + 26044x^2 - 7.4529e+05y^2 + 2.7976e+05x + 3.7333e+05$
time: 0.16s

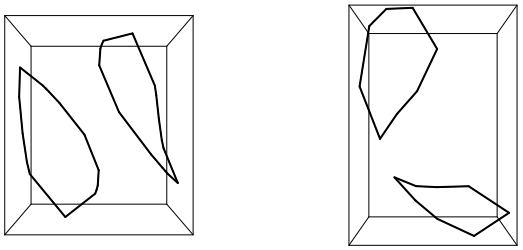
c) $P = -8y^7 - 7x^6 + 42x^5y - 105x^4y^2 + 140x^3y^3 - 105x^2y^4 + 42xy^5 - 7y^6 + 48y^5 + 35x^4 - 140x^3y + 210x^2y^2 - 140xy^3 + 35y^4 - 80y^3 - 42x^2 + 84xy - 42y^2 + 32y + 7$
time: 0.28s

4.2 Examples of 3D curves

Here are some experimentations on 3D curves defined by two polynomials, showing the set of segments describing the topology of the curve and the time needed to compute them (in seconds).



(a) (b)



(c) (d)

a) $P = 0.85934x^2 + 0.259387xy + 0.880419y^2 + 0.524937xz - 0.484008yz + 0.510242z^2 - 1$
$Q = 0.95309x^2 + 0.303149xy + 0.510242y^2 - 0.200075xz + 0.64647yz + 0.786669z^2 - 1$
time: 0.17s

b) $P = -0.125x^2 - 0.0583493xy + 0.493569y^2 + 0.966682xz - 1.5073yz - 0.368569z^2 - 0.865971x - 0.433067y - 0.250095z$
$Q = x^2 + y^2 + z^2 - 2$
time: 0.15s

c) $P = 2x^2 + y^2 + z^2 - 4$
$Q = x^2 + 2xy + y^2 - 2yz - 2z^2 + 2zx$
time: 0.13s

d) $P = x^4 + y^4 + 2x^2y^2 + 2x^2 + 2y^2 - x - y - z$
$Q = x^4 + 2x^2y^2 + y^4 + 3x^2y - y^3 + z^2$
time: 1.21s

References

[1] K. Abdel-Malek and H.-J. Yeh. On the determination of starting points for parametric surface intersections. *Computer-Aided Design*, 28:21–35, 1997.

[2] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling an A. McKenney, S. Ostrouchov, and D. Sorensen. *LAPACK Users' Guide*. SIAM, Philadelphia, 1992. `http://www.netlib.org/lapack/`.

[3] S. Basu, R. Pollack, and M.-F. Roy. *Algorithms in Real Algebaric Geometry.* Springer-Verlag, Berlin, 2003. ISBN 3-540-00973-6.

[4] L. Busé, M. Elkadi, and B. Mourrain. Using projection operators in computer aided geometric design. In *Topics in Algebraic Geometry and Geometric Modeling,*, pages 321–342. Contemporary Mathematics, 2003.

[5] D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra.* Undergraduate Texts in Mathematics. Springer Verlag, New York, 1992.

[6] G. Dos Reis, B. Mourrain, R. Rouillier, and Ph. Trébuchet. An environment for symbolic and numeric computation. In *Proc. of the International Conference on Mathematical Software 2002*, World Scientific, pages 239–249, 2002.

[7] D. Eisenbud. *Commutative Algebra with a view toward Algebraic Geometry*, volume 150 of *Graduate Texts in Math.* Berlin, Springer-Verlag, 1994.

[8] M. Elkadi and B. Mourrain. *Introduction à la résolution des systèmes d'équations algébriques*, 2003. Notes de cours, Univ. de Nice (310 p.).

[9] G. Farin. An ssi bibliography. In *Geometry Processing for Design and Manufacturing*, pages 205–207. SIAM, Philadelphia, 1992.

[10] T. Garrity and J. Warren. Geometric continuity. *Computer Aided Geometric Design*, 8:51–65, 1991.

[11] Laureano González-Vega and Ioana Necula. Efficient topology determination of implicitly defined algebraic plane curves. *Comput. Aided Geom. Design*, 19(9):719–743, 2002.

[12] T. A. Grandine. Applications of contouring. *SIAM Review*, 42:297–316, 2000.

[13] T. A. Grandine and F. W. Klein. A new approach to the surface intersection problem. *Computer Aided Geometric Design*, 14:111–134, 1997.

[14] Gert-Martin Greuel and Gerhard Pfister. *A* **Singular** *introduction to commutative algebra.* Springer-Verlag, Berlin, 2002. With contributions by Olaf Bachmann, Christoph Lossen and Hans Schönemann, With 1 CD-ROM (Windows, Macintosh, and UNIX).

[15] J. Harris. *Algebraic Geometry, a first course*, volume 133 of *Graduate Texts in Math.* New-York, Springer-Verlag, 1992.

[16] S. Krishnan and D. Manocha. An efficient intersection algorithm based on lower dimensional formulation. *ACM Transactions on Computer Graphics*, 16:74–106, 1997.

[17] B. Mourrain. A new criterion for normal form algorithms. In M. Fossorier, H. Imai, Shu Lin, and A. Poli, editors, *Proc. AAECC*, volume 1719 of *LNCS*, pages 430–443. Springer, Berlin, 1999.

[18] J. Owen and A. Rockwood. Intersection of general implicit surfaces. In *Geometric Modeling: Algorithms and New Trends*, pages 335–345. SIAM, Philadelphia, 1987.

[19] M. P. Patrikalakis and T. Maekawa. *Shape Interrogation for Computer Aided Design and Manufacturing*. Springer Verlag, 2002.

[20] Ph. Trébuchet. *Vers une résolution stable et rapide des équations algébriques*. PhD thesis, Université Pierre et Marie Curie, 2002.

[21] J. von zur Gathen and J. Gerhard. *Modern computer algebra*. Cambridge University Press, New York, 1999.

[22] Hassler Whitney. *Complex analytic varieties*. Addison-Wesley Publishing Co., Reading, Mass.-London-Don Mills, Ont., 1972.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399