Subdivision Methods for the Topology of 2d and 3d Implicit Curves

Chen Liang & B. Mourrain & J.P. Pavone

GALAAD, INRIA
BP 93, 06902 Sophia Antipolis
France

Abstract. In this paper, we describe a subdivision method for handling algebraic implicit curves in 2d and 3d. We use the representation of polynomials in the Bernstein basis associated with a given box, to check if the topology of the curve is determined inside this box, from its points on the border of the box. Subdivision solvers are used for computing these points on the faces of the box, and segments joining these points are deduced to get a graph isotopic to the curve. Using envelop of polynomials, we show how this method allow to handle efficiently and accurately implicit curves with large coefficients. We report on implementation aspects and experimentations on 2d curves such as ridge curves or self intersection curves of parameterized surfaces, and on silhouette curves of implicit surfaces, showing the interesting practical behavior of this approach.

1 Introduction

In this paper, we address the problem of computing the topology of 3D curves resulting from the intersection of two algebraic surfaces. Algebraic curves and surfaces are compact representations of shapes, which can be complex and have numerous advantages over parametric ones, such as easy determination of inside/outside of the surface. This is particularly useful when we have to apply logical operations (union, subtraction, etc.) between two solid objects, defined implicitly. In such problems, computing the intersection of two surfaces is a critical operation, which has to be performed efficiently and accurately. Implicit curves and surfaces have also disadvantages such as difficulty in performing graphical display, but the method that we propose in this paper is step towards handling such problems, since it allows fast display of this implicitly 2d and 3d curves. On the other hand, dealing with parameterized surfaces naturally leads to the computation of implicit curves. Let us mention in particular, the computation of the intersection curve of two surfaces, self-intersection curves, plane sections and ridge curves (which are defined implicitly on these parameterised surfaces, though they are usually approximated by parameterised curves). Such problems reduce to the analysis of a curve defined by $n-1$ polynomial equations, in a space of dimension n (here $n = 2, 3, 4$).

One major obstacle for adopting implicit representations instead of parametric representations concerns the piecewise linear approximation of such curves

or surfaces for visualization purposes. A brute force approach would be an exhaustive evaluation for approximating the zero level set, which is obviously very inefficient. A typical alternative scenario is to adopt a divide-and-conquer approach. Larger undetermined domains are broken down to smaller predictable domains in which the topological feature and eventually, the curve/surface itself can be inferred efficiently. An objective of this paper is to describe an efficient method, which allows us to capture the topology of an implicit curve, when this curve is smooth[1], but also to localize the singular points if they exist.

The problem of computing the topology of curves has been approach in different ways. A first family of methods is based on a sweeping approach. For 2D planar algebraic curves, such approach has been studied in [GK97] and [GVN02]. It was later extended by Gatellier et al. in [GLMT05] to the 3D spatial curves resulting from the intersection of two algebraic surfaces. See also [AS05]. These methods use a conceptual sweeping line/plane perpendicular to some projection axis, and detect the critical topological events, such as tangents to the sweeping planes and singularities. The final output of these methods are a graph of connected vertices complying to the topology of the original curves. A notable problem of aforementioned approaches is that they relies of the computation of sub-resultant sequences, which can be a bottleneck in many examples with large degree and large coefficients (see Section 4.1).

Another family of methods are the subdivision based techniques, which uses a simple criterion to remove domains which do not contain the roots. A crucial problem involved here is how to efficiently and reliably deduce the root information in a given interval (or a bounding box). In these methods, instead of using monomial representation, we represent the equations using Bernstein basis [Far93]. Among early attempts, Sederberg [Sed89] converted an algebraic curve in to piecewise triangular Bernstein basis. See also [KCMK00] combining symbolic and numeric techniques to compute the topology of 2D curves. The approach of [HFH$^+$05] for computing the curves of intersection of two parameterised surfaces is also combining subdivision techniques with regularity criterion, exploiting the properties of the intersection curve in the 2D parameter domains.

The first problem of computing roots of univariate polynomials has been analyzed for instance in [MRR05], where root information tests are by based on *Descartes' Law of Sign* and its variant in the Bernstein basis. This approach has been extended to the approximation of isolated roots of multivariate systems. In [SP93], the author used tensor product version of Bernstein basis and integrated domain reduction techniques to speed up the convergence and reduce the number of subdivisions. In [EK01], the emphasis is put on the subdivision process, and stopping criterion based on the normal cone to the surface patch. In [MP05], this approach has been improved by introducing pre-conditioning and univariate-solver steps. The complexity of the method is also analyzed in terms of intrinsic differential invariants.

[1] The tangent vector space exists at every points

The application of subdivision methods for handling higher dimensional objects is not so well developed. In [JKGMS05] a method which subdivides up to some precision level, and applies dual marching cube approach to connect points on the curve or to mesh a surface is described. The variety is covered by boxes of a given size, and the connectivity of these cells is used to deduce the piecewise linear approximation. In [ACM05], a subdivision approach exploiting the sign variation of the coefficients in the Bernstein basis in order to certify the topology of the surface in a cell, is used for the purpose of polygonalizing an implicit algebraic surface.

The work of this paper is in the spirit of this former approach. We apply a subdivision approach also exploiting the properties of the Bernstein polynomial representation. We describe a simple regularity test extending the criterion of [ACM05] to curves, which allows us to detect easily when the topology of the curve in a cell is uniquely determined from its intersection with the border of the cell. This provides an efficient test for stopping earlier the subdivision process and branching to path following methods if we are interested in a good geometrical approximation of the curve.

We address the same question as in [GLMT05], but with this new methods, we are able to solve the following problems already identified in this paper:

- To achieve higher numerical stability by operating on Bernstein basis instead of monomial basis;
- Through subdivision on three principle directions, i.e. x, y, z (or x, y, z), to isolate the domain containing the singularities from those containing regular curve segments. This divide-and-conquer approach, in principle, should simplify the graph building algorithm adopted in [GLMT05] where the whole domain has to be considered.

However, for the treatment of singular points, we have to introduced a threshold ϵ to stop the subdivision. Contrarily to [GLMT05], we do not certify the topology at singular points, but computed boxes of size ϵ, containing these singularities.

On the contrary, we show that our approach is able to handle implicit curves with large equation (of total degree about 80 with coefficients of bit-size 200), which resultant-based techniques are not able to treat.

This paper is organized as follows: in Section 2, we will review some of the relevant concepts and theorem required by our proposed algorithm; Section 3 is devoted to outlining our proposed algorithms and the details about how the essential steps in our algorithm are handled. We will show the experiment results in Section 4 and conclude in Section 5 with the problems and possible improvements over the currently proposed algorithm.

2 Fundamental ingredients

This section introduces the theoretical background of Bernstein polynomial representation and how it is related to the problem we want to solve. For a domain $D \subset \mathbb{R}^n$, we denote by $\overset{\circ}{D}$ its interior, by $\overline{D}$ its closure. For a box $D =$

$[a_0, b_0] \times [a_1, b_1] \times [a_2, b_2] \subset \mathbb{R}^3$, its x-face (resp. y-face, z-face) are its faces normal to the direction x (resp. y, z).

2.1 Univariate Bernstein Basis

Given an arbitrary univariate polynomial function $f(x) \in \mathbb{K}$, we can convert it into the representation of Bernstein basis of degree d, which is defined by:

$$f(x) = \sum_i b_i B_i^d(x), \text{ and} \tag{1}$$

$$B_i^d(x) = \binom{d}{i} x^i (1-x)^{d-i} \tag{2}$$

where b_i is usually referred as controlling coefficients. Such conversion is done through a basis conversion [Far93]. The above formula can be generalized to an arbitrary interval $[a, b]$ by a variable substitution $x' = (b-a)x + a$. We denote by $B_d^i(x; a, b)\binom{d}{i}(x-a)^i(b-x)^{d-i}(b-a)^{-d}$ the corresponding Bernstein basis on $[a, b]$.

There are several useful properties regarding Bernstein basis given as follows:

- *Convex-Hull Properties*: Since $\sum_i B_d^i(x; a, b) \equiv 1$ and $\forall x \in [a, b], B_d^i(x; a, b) \geq 0$ where $i = 0, ..., d$, the graph of $f(x) = 0$, which is given by $(x, f(x))$, should always lie within the convex-hull defined by the control coefficients [Far90].
- *Subdivision* (de Casteljau): Given $t_0 \in [0, 1]$, $f(x)$ can be represented piecewisely by:

$$f(x) = \sum_{i=0}^{d} b_0^{(i)} B_d^i(x; a, c) = \sum_{i=0}^{d} b_i^{(d-i)} B_d^i(x; c, b), \text{ where} \tag{3}$$

$$b_i^{(k)} = (1-t_0) b_i^{(k-1)} + t_0 b_{i+1}^{(k-1)} \text{ and } c = (1-t_0)a + t_0 b. \tag{4}$$

Another interesting property of this representation is related to Descartes's Law of signs. The definition of Descartes's Law for a sequence of coefficients $\mathbf{b}_k = b_i | i = 1, ..., k$ is defined recursively:

$$V(\mathbf{b}_{k+1}) = V(\mathbf{b}_k) + \begin{cases} 1, \text{if } b_i b_{i+1} < 0 \\ 0, \text{else} \end{cases} \tag{5}$$

With this definition, we have:

Theorem 1. *Given a polynomial $f(x) = \sum_i^n b_i B_i^d(x; a, b)$, the number N of real roots of f on $]a, b[$ is less than or equal to $V(\boldsymbol{b})$, where $\boldsymbol{b} = (b_i), i = 1, ..., n$ and $N \equiv V(\boldsymbol{b}) \mod 2$.*

The theorem 1 enables a simple yet efficient test of the existence of real roots in a given domain. This test is essential to our algorithm, as it serves as a key criterion to classify whether a domain has certified topology, without actually computing the curve. This allow the our algorithm to execute in reasonably short time, as demonstrated in our experiments.

2.2 Generalization to Multivariate Case

The univariate Bernstein basis representation can be generalized to multivariate ones. Briefly speaking, we can rewrite the definition (Eq. (1)) in the form of tensor products. Suppose for $\mathbf{x} = (x_0, ..., x_{n-1}) \in \mathbb{R}^n$, $f = (\mathbf{x}) \in \mathbb{K}[\mathbf{x}]$ having the maximum degree $\mathbf{d} = (d_0, ..., d_{n-1})$ has the form:

$$f(\mathbf{x}) = \sum_{k_0=0}^{d_0} ... \sum_{k_n=0}^{d_n} b_{k_1,...,k_n} B_{k_0}^{d_0}(x_0)...B_{k_n}^{d_0}(x_n) \tag{6}$$

For a polynomial of n variables, the coefficients can be viewed as a tensor of dimension n.

The de Casteljau subdivision for the multivariate case proceeds similarly to the univariate one, since the subdivision can be done independently with regards to a particular variable x_i.

Based on these properties, a subdivision solver which can be seen as an improvement of the *Interval Projected Polyhedron* algorithm in [SP93], is described in [MP05]. It uses the following operations: The multivariate functions to be solved are enclosed in-between two univariate functions, for each variable. For this purpose, the Bernstein control points of the functions are projected in each direction and the upper and lower envelop are used to define these enveloping univariate polynomials. A lower and upper approximation of the roots of these univariate polynomials are used to reduce the domain. If the reduction is not sufficient, the domain is split. These reduction operations are improved by pre-conditioning steps. See [MP05] for more details.

3 Algorithmic ingredients

We consider the problem of computing the topology of the curve, denoted hereafter as $\mathcal{C}$, resulting from the intersection of two known algebraic surfaces, namely, $f(\mathbf{x}) = 0$ and $g(\mathbf{x}) = 0$ defined in $\mathbb{R}^3$, with $f, g \in \mathbb{R}[x, y, z]$. Our discussion is confined to the case where f and g has no common divisor other than 1, so that their intersection has dimension of 1. We assume moreover that (f, g) is radical or equivalently that the resultant of $f(x, y, z), g(x, y, z)$ with respect to z after a generic change of coordinates, is square free.

3.1 Tangent Vector Field

The tangent vector on $\mathcal{C}$ serves as the key to our analysis of topology of the curve. It serves as an important indicator of topological feature of $\mathcal{C}$. While it is computationally prohibitive to compute the tangent vector at each point on $\mathcal{C}$, we can reach some useful conclusion about the topology of the curve by looking into the tangent vector field defined below:

$$\mathbf{t} = \mathbf{t}_x(\mathbf{x})\mathbf{e}_x + \mathbf{t}_y(\mathbf{x})\mathbf{e}_y + \mathbf{t}_z(\mathbf{x})\,\mathbf{e}_z = \bigtriangledown f \wedge \bigtriangledown g = \begin{vmatrix} \mathbf{e}_x & \mathbf{e}_y & \mathbf{e}_z \\ \partial_x f & \partial_y f & \partial_z f \\ \partial_x g & \partial_y g & \partial_z g \end{vmatrix} \tag{7}$$

where $\mathbf{e}_x$, $\mathbf{e}_y$ and $\mathbf{e}_z$ are the unit vectors along the principle axis x, y and z, respectively; $\mathbf{t}_x$, $\mathbf{t}_y$ and $\mathbf{t}_z$ are functions of $\mathbf{x} = (x, y, z)$.

Singularities on the curve can be easily characterized, as $\mathbf{t}$ vanishes at those points. In [GLMT05], the author also tried to localize the point having a tangent parallel to a virtual sweeping plane. They are connected together with the singularities to form the final topological graph. In order to do this, the whole curve is projected onto some principle projection planes. However, the projected planar curve in many cases has a very different topology as $\mathcal{C}$. In our proposed algorithm, we exploit the subdivision along all three principle axes simultaneously and the critical events are either reduced to regular case (such as for tangents) or localized (such as for intersections). The topology graph can be built without explicitly computing the exact position of the singularities.

3.2 Regularity Test

In this section, we are going now to describe how to detect boxes, for which the topology of the curve can be determined. We will use the following notions:

Definition 1. *We say that a curve $\mathcal{C} \in \mathbb{R}^n$ is* regular *in a compact domain $D \subset \mathbb{R}^n$, if its topology is uniquely determined from its intersection with the boundary D.*

The aim of the method is to give a simple criterion for the regularity of a curve in a box.

To form the topological graph for this domain, we only need to compute the intersections between the curve and the boundary of this domain, and there exists a unique graph to link these intersections so that this graph complies to the true topology of the original curve.

2D case: For 2D planar algebraic curve $\mathcal{C}$ defined by a polynomial equation $f(x, y) = 0$, and denoting the partial derivative of f w.r.t x by $\partial_x f$, we have the following direct property:

Proposition 1. *If $\partial_y f(x, y) \neq 0$ (resp. $\partial_x f(x, y) \neq 0$) in a domain $D = [a_0, b_0] \times [a_1, b_1] \subset \mathbb{R}^2$, the curve $\mathcal{C}$ is regular on D.*

Proof. Suppose that $\partial_y f(x, y) \neq 0$ in D. Then $\mathcal{C}$ is smooth, since its normal vector is defined everywhere, and has no vertical tangents in D. By the implicit function theorem, the connected components of $\mathcal{C} \cap \overset{\circ}{D}$ are the graph of functions of the form $y = \phi(x)$. The closure of such a connected component is called hereafter a branch of $\mathcal{C}$ in D. As $\partial_y f(x, y) \neq 0$ in D, for a given $x \in [a_0, b_0]$ there is at most one branch of $\mathcal{C}$ in D above x. Consequently, the connected components of $\mathcal{C} \cap \overset{\circ}{D}$ project bijectively onto non-overlapping open intervals of $[a_0, b_0]$.

Moreover, as there is no vertical tangent, each of these branches starts and ends at a point on the border ∂D of D. Notice that two branches may share a

starting or ending point, when the curve is tangent (with even multiplicity) to ∂D.

Thus, computing the points of $\mathcal{C} \cap \partial D$, repeating a point if its multiplicity is even, sorting them by lexicographic order such that $x > y$ ($(x_0, y_0) > (x_1, y_1)$ if $x_0 > x_1$ or $x_0 = x_1$ and $y_0 > y_1$), we obtain a sequence of points $p_1, p_2, \ldots, p_{2\,s-1}, p_{2\,s}$ such that the curve $\mathcal{C}$ in D is isotopic to the union of the non-intersecting segments $[p_1, p_2], \ldots, [p_{2s-1}, p_{2s}]$. In other words, the topology of $\mathcal{C}$ is uniquely determined from its intersection points with ∂D and $\mathcal{C}$ is regular on D. □

If $\partial_y f \neq 0$ on D (resp. $\partial_x f \neq 0$), we will say that $\mathcal{C}$ is x-regular (resp. y-regular). A sufficient condition for f to be x-regular (respectively y-regular) is that the Bernstein coefficients of the first derivative of f against y (respectively x) maintains a constant sign (see also [ACM05]). By Descartes' law, this statement implies that the sign variation in this direction should be at most 1.

To put it in another way, by solely studying the sign variations of the tangential gradient vector of the curve (represented in Bernstein basis), i.e. $(\partial_y f(\mathbf{x}), -\partial_x f(\mathbf{x}))$, we are able to detect when the curve is regular on D and to determine uniquely the topological graph.

3D case: The 2D approach can be generalized to the 3D case where the tangential gradient vector of the curve C defined by the intersection of two algebraic surfaces, namely $f(x, y, z) = 0$ and $g(x, y, z) = 0$, is given by $\mathbf{t} = \bigtriangledown(f) \wedge \bigtriangledown(g)$ (see Eh. (7)). Similar to the 2D case, we can represent each component of $\mathbf{t}$ in the Bernstein basis for a given domain (in cube shape) $D = [a_0, b_0] \times [a_1, b_1] \times [a_2, b_2]$. The sign change of the resulting Bernstein coefficients enables a simple regularity test with minimal computation effort.

We describe a first and simple regularity criterion:

Proposition 2. *The 3D spatial curve $\mathcal{C}$ defined by $f = 0$ and $g = 0$ is regular on D, if*

- $\boldsymbol{t}_x(\boldsymbol{x}) \neq 0$ *on D, and*
- $\partial_y h \neq 0$ *(or $\partial_z h \neq 0$) on D, for $h = f$ or $h = g$.*

Proof. Suppose that $\mathbf{t}_x(\mathbf{x}) \neq 0$ and $\partial_z(f) \neq 0$ on D. It implies that $\mathcal{C}$ is smooth in D. Consider two branches of $\mathcal{C}$ in D and project them by π_z onto a (x, y)-plane. Their projection cannot intersect at an interior point. Otherwise, there would be two points $p_1, p_2 \in D$, such that $f(p_1) = 0, f(p_2) = 0$ and $\pi_z(p_1) = \pi_z(p_2)$, which implies that $\partial_z f(p)$ vanishes for an intermediate point $\in]p_1, p_2[$ in D. This is impossible by hypothesis. Consequently, the branches of $\mathcal{C}$ project bijectively onto the branches of $\pi_z(\mathcal{C})$. Their tangent vector is the projection $(\mathbf{t}_x(\mathbf{x}), \mathbf{t}_y(\mathbf{x}))$ of the tangent vector of $\mathcal{C}$. By proposition 1, $\pi_z(\mathcal{C})$ is regular, so that the topology of $\pi_z(\mathcal{C})$, and thus of $\mathcal{C}$, is uniquely determined by the intersection points of $\mathcal{C}$ with the border of D. □

A similar criterion applies by symmetry, exchanging the roles of the x, y, z coordinates.

Let us give now a finer regularity criterion, which is computationally less expensive:

Proposition 3. *If $\mathcal{C}$ is smooth in D and if for all $x_0 \in \mathbb{R}$, the plane $\boldsymbol{x} = x_0$ plane has at most one intersection point with the curve $\mathcal{C}$ in D, then $\mathcal{C}$ is regular on D.*

Proof. Consider the projection $\pi_z(\mathcal{C})$ of the curve $\mathcal{C}$ in D along the z direction. Then the components of $\mathcal{C}$ in D projects bijectively on the (y, z) plane. Otherwise, there exist two points p_0 and p_1 lying on $\mathcal{C}$ such that $\pi_z(p_0) = \pi_z(p_1) = (x_0, y_0)$, then p_0 and p_1 belong to $x = x_0$ which are functions of the form $y = \Phi(x)$. Otherwise, there exist two points on $\pi_z(\mathcal{C})$ and (and on $\mathcal{C} \cap D$) with the same x-coordinate. Consequently, for $x \in [a_0, b_0]$ there is at most one branch of $\pi_z(\mathcal{C})$ in D above x, and the connected components of $\mathcal{C} \cap \overset{\circ}{D}$ project bijectively onto non-overlapping open intervals of $[a_0, b_0]$ as $\pi_z(\mathcal{C})$ does. We conclude as in the 2D case (proposition 1), by sorting the points of $\mathcal{C} \cap \partial D$ according to their x-coordinates, and by gathering them by consecutive pairs corresponding to the starting and ending points of branches of $\mathcal{C} \cap D$. □

Proposition 4. *The 3D spatial curve $\mathcal{C}$ defined by $f = 0$ and $g = 0$ is regular on D, if*

- $\boldsymbol{t}_x(\boldsymbol{x}) \neq 0$ *on D, and*
- $\partial_y h \neq 0$ *on z-faces, and $\partial_z h \neq 0$ and its has the same sign on both y-faces of D, for $h = f$ or $h = g$.*

Proof. Let us fix $x_0 \in [a_0, b_0]$ where $D = [a_0, b_0] \times [a_1, b_1] \times [a_2, b_2]$, let $U = \{x_0\} \times [a_1, b_1] \times [a_2, b_2]$ and let $\Phi_{x_0} : (x_0, y, z) \in U \mapsto (f(x_0, y, z), g(x_0, y, z))$. We are going to prove that under our hypothesis, Φ_{x_0} is injective. The Jacobian $\mathbf{t}_x(x_0, y, z)$ of Φ_{x_0} does not vanish on U, so that Φ_{x_0} is locally injective. We consider the level-set $f(\mathbf{x}) = f_0$ for some $f_0 \in f(U)$. It cannot contain a closed loop in U, otherwise we would have $(\partial_y f, \partial_z f) = 0$ (and thus $\mathbf{t}_x = 0$) in $U \subset D$. We deduce that each connected component of $f(\mathbf{x}) = f_0$ in U intersects ∂U in two points.

Now suppose that Φ_{x_0} is not injective on U, so that we have two points $p_1, p_2 \in U$ such that $\Phi_{x_0}(p_1) = \Phi_{x_0}(p_2)$.

If p_1 and p_2 are on the same connected component of the level set $f(\mathbf{x}) = f_0$ (where $f_0 = f(p_1) = f(p_2)$) in U, then g reaches the same value at p_1 and p_2 on this level set, so that by Role's theorem, there exists a point $p \in U$ in-between p_1 and p_2, such that $\mathrm{Jac}(\Phi_{x_0})(p) = \mathbf{t}_x(p) = 0$. By hypothesis, this is impossible.

Thus p_1 and p_2 belongs to two different connected components of $f(\mathbf{x}) = f_0$ in U. Consequently the value f_0 is reached at 4 distinct points of ∂U, which implies that f has at least 4 extrema on ∂U.

Now note that up to a change of variable $z = a_2 - z$, we can assume that $\partial_z f > 0$ on both $y = a_1, y = b_1$ faces. Then if $\partial_y f < 0$ on $z = a_2$, we have $f(x_0, a_1, b_2) > f(x_0, a_1, a_2) > f(x_0, b_1, a_2)$ and (a_2, a_3) is not a local extrema.

Otherwise $\partial_y f > 0$ and (b_1, a_2) is not a local extrema. In both cases, we do not have 4 extrema, which proves that ϕ_{x_0} is injective and that the intersection of $\mathcal{C}$ with the plane $x = x_0$ in D is at most one point. So by proposition 3, we deduce that $\mathcal{C}$ is regular in D. □

For more details on the injectivity properties, see [Pav04]. Here also, a similar criterion applies by symmetry, exchanging the roles of the x, y, z coordinates.

If one of these criteria applies with $\mathbf{t}_i(x) \neq 0$ on D (for $i = x, y, z$), we will say that $\mathcal{C}$ is i-regular on D.

From a practical point of view, the test that $\mathbf{t}_i(x) \neq 0$ or $\partial_i(h)$ for $i = x, y$ or z, $h = f$ or g, is replaced by the stronger condition that their coefficients on the Bernstein basis of D have a constant sign, which is straightforward to check. Similarly, such a property on the faces of D is also direct, since the coefficients of a polynomial, with a minimal (resp. maximal) x-indices (resp. y-indices, z-indices) are its Bernstein coefficients on the corresponding face.

In addition to these tests, we also test whether both surfaces penetrate the cell, since a point on the curve must lie on both surfaces. This test could be done by looking at the sign change of the Bernstein coefficients of the surfaces with regards to that cell. If no sign change occurs, we can rule out the possibility that the cell contains any portion of the curve $\mathcal{C}$, hence terminate the subdivision early. In this case, we will also say that the cell is regular.

The regularity criterion is sufficient for us to uniquely construct the topological graph g of $\mathcal{C}$ within D. Without loss of generality, we suppose that the curve $\mathcal{C}$ is x-regular in D. Hence, there is no singularity of $\mathcal{C}$ in D. Furthermore, this also guarantees that there is no 'turning-back' of the curve tangent along x-direction, so the mapping of $\mathcal{C}$ onto the x axis is injective. Intuitively, the mapped curve should be a series of non-overlapping line segments, of which the ends correspond to the intersections between the curve $\mathcal{C}$ and the cell, and such mapping is injective.

This property leads to a unique way to connect those intersection points, once they are computed (see section 3.3), in order to obtain a graph representing the topology of $\mathcal{C}$. Here is how this graph is computed in practice: suppose $\mathcal{C}$ is i-regular in the domain D, and that we have computed the set of intersection points $V = \{\mathbf{v}_j\}$ of the curve with the boundary of D. First, we sort the elements in V comparing vectors by their i-th coordinates. Assuming the sorted points $\mathbf{v}_j$ are indexed by $j = 0, 1, 2, ...$, we form the edges $\mathbf{v}_k, \mathbf{v}_{k+1}$, for $k = 0, 2, 4, ...$

However, a special case has to be taken into account, that is when $\mathbf{v}_j$ has a multiplicity $m_i > 1$, for instance, when $\mathcal{C}$ is tangent to the bounding domain D at $\mathbf{v}_i$. In this case, we can treat $\mathbf{v}_j$ conceptually as a multiple point which plays the role of m_i points. In this way, we proceed the connecting process in the same manner as we do for the general case. To determine the multiplicity of a point $\mathbf{v}_j$, we only have to evaluate the derivatives of $\mathcal{C}$ at this point.

3.3 Hierarchical subdivision

We adopted a hierarchical octree to partition the $\mathbb{R}^3$ space, for several reasons:

– each cell of the octree is equivalent to a cube-shaped domain D; which stores the coefficients of the polynomials in the Bernstein basis of the corresponding domain.
– we can take cares of faces shared by cells, to minimize the number of calls to solvers;
– the hierarchical structure of octree allows us to terminate (stop further subdivision) early when a cell is deemed regular or irrelevant.

We begin by setting a initial bounding domain D_0 to a root cell. A cell is subdivided if the curve $\mathcal{C}$ defined in the correspondent domain fails the regularity test. For each subdivision, we result in several smaller domains in form of sub-cells. For each of them, we repeat the regularity test and, if necessary, further subdivides. The subdivision of a cell will terminate either when the curve within is deemed regular, or the size of the cell is beyond a predefined precision ϵ.

There are several techniques to save computation efforts. As the sub-cells share certain faces with their parent cell, the earlier computed intersections on the parent cell's faces are inherited directly by the sub-cells. In addition, sub-cells split from the same parent cell do share some faces as well. Once again, the shared faces should be computed exactly once.

Once a new face is introduced in the octree decomposition, the bivariate solver described in section 2.2, is called directly with the Bernstein coefficients of the polynomials on this face. The points we found are shared by neighbor cells, connected to this face in the octree.

3.4 Symbolic-numeric approach

Some geometric operations such as computing the self-intersection curve or the ridge curve of a parameterized surface leads to the computation of implicit curves of high degree with coefficients of large size. This is either due to projection techniques (see [GP05]), or to their definition through composed operations (see [CFPR05]). In order to be able to handle such curve, the main difficulty is to control the result, using approximate computation, since exact computation though possible, would be prohibitive. We describe here the symbolic-numeric approach that we have developed for this purpose.

We assume that the input equations are given with exact (large) rational (or integer) numbers (even if the input is given with floating point numbers, we will consider it as an exact input). In order to compute the topology of $\mathcal{C}$ in a domain D, we convert its representation in the Bernstein basis of D, using exact rational arithmetic.

Once this conversion is done, we normalize the equation, by dividing by the coefficient of maximal norm. For each resulting rational coefficient c, we compute the smallest interval $[\underline{c}, \overline{c}]$ represented with floating point numbers and containing c.

Then, the subdivision process is performed, using interval arithmetic. The regularity criterion, which reduces to sign evaluations, is applied on these interval coefficients. We use the following convention: a interval is < 0 (resp. > 0) if all

its elements are < 0 (resp. > 0). If the interval contains 0, we say that its sign is indeterminate.

If the regularity test fails,

- either the sign of all the coefficients of the polynomial are indeterminate, and we re-convert the exact polynomial to its representation on the corresponding sub-domain and restart the approximation process.
- or we subdivide the domain, as in the usual case.

3.5 Algorithm Outline

The proposed algorithm for 3D curves is outlined as following:

Algorithm 31 *Computing the topology of the curve $\mathcal{C}$:*
INPUT: *$f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ polynomials $\in \mathbb{Q}[x, y, z]$, a tolerance ϵ and a list of bounding domain $D_0 \leftarrow [a_0, b_0] \times [a_1, b_1] \times [a_2, b_2]$ ($a_{i,b_i} \in \mathbb{R}$).*

- *Step 0: (initialization step) domain list $\mathcal{D} \leftarrow D_0$; vertex list $V \leftarrow$ NIL; connectivity list $E \leftarrow$ NIL;*
- *Step 1: compute $\boldsymbol{t} \leftarrow \bigtriangledown(f) \wedge \bigtriangledown(g)$ given by Eq. (7);*
- *Step 2: convert f, g and $\boldsymbol{t}$ into Bernstein basis representation;*
- *Step 3: while $\mathcal{D}$ is not empty, pick a D in $\mathcal{D}$:*
  - *Step 3.1: compute V the set of intersection points between the boundary of the domain D and the curve $\mathcal{C}$;*
  - *Step 3.2: if the size of D is larger than ϵ:*
    - *if the curve $\mathcal{C}$ within the domain D is regular (see section 3.2):*
      - *sort and connect the points $\boldsymbol{v} \in V$; the connectivities are stored in E;*
    - *else if the domain D is not regular:*
      - *subdivide D and append the subdivided domains into the domain list D*
  - *else if the size of D is not larger than ϵ:*
    - *add the domain D as a 'box' vertex into V;*
    - *this vertex is connected with all intersections $\boldsymbol{v} \in V$ of D; these connectivities are also appended to E;*
  - *Step 3.4: remove D from $\mathcal{D}$ and repeat Step 3;*

OUTPUT: *The graph represented by a set of vertices V, which are either 3D points or boxes (with size less than ϵ) bounding the singularities, and a set of connections E that are representing the edges of the resulting graph.*

We do not describe the algorithm for 2D curves, which is basically a specialization of this one.

4 Experiments

Our proposed algorithm is implemented as a part of SYNAPS (SYmbolic Numeric APplicationS) library[2]. The experiments have been carried out on a 3.4GHz PC, under Linux.

[2] `http://www-sop.inria.fr/galaad/software/synaps/`

4.1 Planar curves of high degree with large coefficients

In this section, we report on the application of the 2d algorithm, in the case of large integer coefficients. The first example is about ridge curve. Ridge curves correspond to local extrema of curvature taken in the principal direction of the surface, after some algebraic manipulations they can be obtained as implicit curve (see [CFPR05]). See also [TG92] and [TG95] for other related approaches. In the example it corresponds to a bicubic surface, the input polynomial is of total degree 84, of multidegree $(43, 43)$ with 1907 monomials. The coefficients are integers encoded on at most 65 bits. For the precision $\epsilon = 10^{-3}$ which controls the singularity localization, it takes 30 seconds. The topology is certified except in tiny boxes (which contains the singularity points). Notice that a pure algebraic approach, exploiting the specificity of problem and with a very efficient Gröbner engine takes about 10 minutes to certify the topology (see [CFPR05]).

The second example is a projection of a self-intersection curve of bicubic patch, computed by resultant techniques (see [GP05]). The input polynomial is of total degree 76, of multidegree $(44, 44)$ with 1905 monomials. The coefficients are integers of at most 288 bits. It takes 5 seconds, for this example with the same precision $\epsilon = 10^{-3}$.

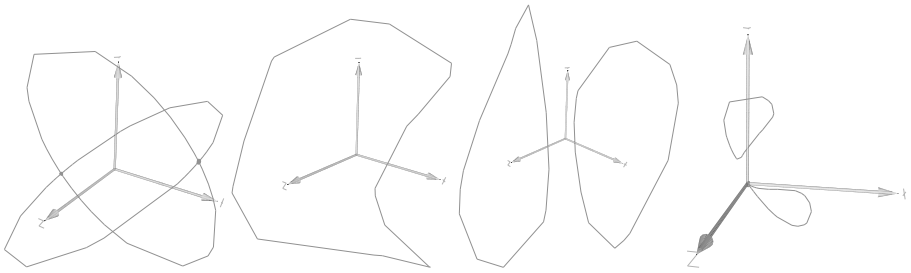
(a) ridge curve (b) self-intersection curve

Fig. 1. Topological descriptions of high complexity curves

4.2 Intersection curves of implicit surfaces

This set of examples are from [GLMT05]. The computational time accompanied is measured up to milliseconds (see Fig. 2):

1) $f(\mathbf{x}) = 0.85934x^2 + 0.259387xy + 0.880419y^2 + 0.524937xz - 0.484008yz + 0.510242z^2 - 1$
   $g(\mathbf{x}) = 0.95309x^2 + 0.303149xy + 0.510242y^2 - 0.200075xz + 0.64647yz + 0.786669z^2 - 1$
   time: 80 msec
2) $f(\mathbf{x}) = -0.125x^2 - 0.0583493xy + 0.493569y^2 + 0.966682xz - 1.5073yz - 0.368569z^2 - 0.865971x - 0.433067y - 0.250095z$
   $g(\mathbf{x}) = x^2 + y^2 + z^2 - 2$
   time: 20 msec
3) $f(\mathbf{x}) = 2x^2 + y^2 + z^2 - 4$
   $g(\mathbf{x}) = x^2 + 2xy + y^2 - 2yz - 2z^2 + 2zx$
   time: 30 msec
4) $f(\mathbf{x}) = x^4 + y^4 + 2x^2y^2 + 2x^2 + 2y^2 - x - y - z$
   $g(\mathbf{x}) = x^4 + 2x^2y^2 + y^4 + 3x^2y - y^3 + z^2$
   time: 130 msec



(a) Example 1 (b) Example 2 (c) Example 3 (d) Example 4

Fig. 2. Topological descriptions of the intersection curve for 4 pairs of low-order algebraic surfaces.

4.3 Silhouette curves of implicit surfaces

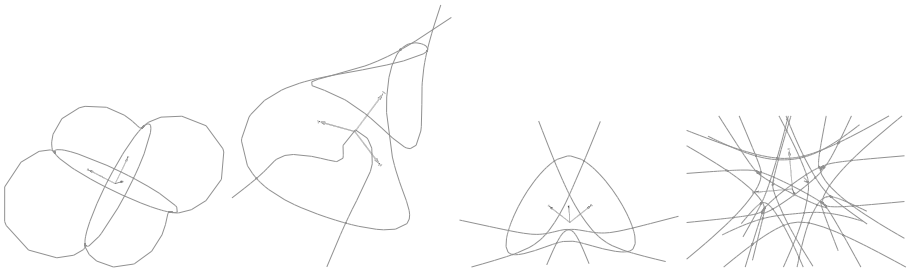
The following samples are taken from *http://www-sop.inria.fr/galaad/surfaces/*. We intersect the surface with its polar variety in one direction (here the x direction). In other words, we intersect the surface with the surface defined by one of its first order derivative (here $\partial_x f$), to extract its silhouette. The surfaces that we used are called respectively *Tetrahedral*, *Q3*, *Q1* and *Barth Sextic* (see Fig.3):

5) $f(\mathbf{x}) = x^4 + 2x^2y^2 + 2x^2z^2 + y^4 + 2y^2z^2 + z^4 + 8xyz - 10x^2 - 10y^2 - 10z^2 + 25$
   $g(\mathbf{x}) = 4x^3 + 4xy^2 + 4xz^2 + 8yz - 20x$
   time: 510 msec

6) $f(\mathbf{x}) = 5.229914547374508y^2z^2 + 3.597883597883598x^2y^2 + y^4 + z^4 - x^4 - 19.49816368932737xyz + 5.229914547374508x^2 - 7.43880040039534y^2$
$g(\mathbf{x}) = -3.597883597883598z^2 + 7.43880040039534z^2x^2 - 110.45982909yz^2 + 7.195767196x^2y + 4y^3 - 19.49816368932737xz - 14.87760080y$
time: 330 msec

7) $f(\mathbf{x}) = x^4 + y^4 + z^4 - 4x^2 - 4y^2z^2 - 4y^2 - 4z^2x^2 - 4z^2 - 4x^2y^2 + 20.7846xyz + 1$
$g(\mathbf{x}) = 4x^3 - 8x - 8xz^2 - 8xy^2 + 20.7846yz$
time: 730 msec

8) $f(\mathbf{x}) = 67.77708776x^2y^2z^2 - 27.41640789x^4y^2 - 27.41640789x^2z^4 + 10.47213596x^4z^2 - 27.41640789y^4z^2 + 10.47213596y^4x^2 + 10.47213596y^2z^4 - 4.236067978x^4 - 8.472135956x^2y^2 - 8.472135956x^2z^2 + 8.472135956x^2 - 4.236067978y^4 - 8.472135956y^2z^2 + 8.472135956y^2 - 4.236067978z^4 + 8.472135956z^2 - 4.236067978$
$g(\mathbf{x}) = 135.5541755xy^2z^2 - 109.6656316x^3y^2 - 54.83281578xz^4 + 41.88854384x^3z^2 + 20.94427192y^4x - 16.94427191x^3 - 16.94427191xy^2 - 16.94427191xz^2 + 16.94427191x$
time: 4010 msec



(a) Example 5 (b) Example 6 (c) Example 7 (d) Example 8

Fig. 3. Topological descriptions of the silhouette curves of algebraic surfaces.

5 Discussion

The algorithm proposed in this paper offers a generic method for computing the topological graph of spatial curves resulting from the intersection of two algebraic surfaces. As demonstrated in the experiments, it is rather robust despite the increase of the complexity of the curve.

The major weakness of this approach, however, is that certain apparently simply situation could result in a lot of subdivisions, such as the curve with parallel structures which are very close to each other.

As specified, in this method, the singular points are only isolated into small boxes of size ϵ, and we do not certify the connection of the branches at these

points. An additional work would be necessary, to certify the singularity type. We are currently investigating on this problem.

Another weakness of this approach is the memory consumption (as the storage requirement is approximately cubic to the depth of the subdivision). This problem however, can be mitigated by a divide-and-conquer approach and pure programming techniques.

Acknowledgements: We would like to thanks J. Wintz, for his very nice software AXEL[3] for the visualisation and manipulation of algebraic objects, that we used to produce the pictures of the curves. We acknowledge the partial support of Aim@Shape (IST NoE 506766) and ACS (IST Fet Open 006413).

References

[ACM05] L. Alberti, G. Comte, and B. Mourrain. Meshing implicit algebraic surfaces: the smooth case. In L.L. Schumaker M. Maehlen, K. Morken, editor, *Mathematical Methods for Curves and Surfaces: Tromso'04*, pages 11–26. Nashboro, 2005.

[AS05] J. Gerardo Alcázar and J. Rafael Sendra. Computing the topology of real algebraic space curves. *J. Symbolic Comput.*, 39:719–744, 2005.

[CFPR05] F. Cazals, J.-C. Faugère, M. Pouget, and F. Rouillier. Topologically certified approximation of umbilics and ridges on polynomial parametric surface. Technical Report 5674, INRIA Sophia-Antipolis, 2005.

[EK01] G. Elber and M.-S Kim. Geometric constraint solver using multivariate rational spline functions. In *Proc. of 6th ACM Symposium on Solid Modelling and Applications*, pages 1–10. ACM Press, 2001.

[Far90] G. Farin. *Curves and surfaces for computer aided geometric design : a practical guide*. Comp. science and sci. computing. Acad. Press, 1990.

[Far93] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide, 3rd Ed.* Academic Press, 1993.

[GK97] T. Grandine and F. Klein. A new approach to the surface intersection problem. *Computer Aided Geometric Design*, 14(2):111–134, 1997.

[GLMT05] G. Gatellier, A. Labrouzy, B. Mourrain, and J.-P. Técourt. *Computing the topology of 3-dimensional algebraic curves*, pages 27–44. Springer-Verlag, 2005.

[GP05] A. Galligo and J.P. Pavone. Self-intersections of a Bézier bicubic surface. In M. Kauers, editor, *Proc. Intern. Symp. on Symbolic and Algebraic Computation*, pages 148–155. New-York, ACM Press., 2005.

[GVN02] L. González-Vega and I. Necula. Efficient topology determination of implicitly defined algebraic plane curves. *Computer Aided Geometric Design*, 19(9):719–743, 2002.

[HFH+05] J. Hass, R. T. Farouki, C. Y. Han, X. Song, and T. W. Sederberg. Guaranteed Consistency of Surface Intersections and Trimmed Surfaces Using a Coupled Topology Resolution and Domain Decomposition Scheme. *Advances in Computational Mathematics*, 2005. To appear.

[JKGMS05] Seong Joon-Kyung, Elber Gershon, and Kim Myung-Soo. Contouring 1- and 2-Manifolds in Arbitrary Dimensions. In *SMI'05*, pages 218–227, 2005.

[3] `http://www-sop.inria.fr/galaad/software/axel/`

[KCMK00] J. Keyser, T. Culver, D. Manocha, and S. Krishnan. Efficient and exact manipulation of algebraic points and curves. *Computer-Aided Design*, 32(11):649–662, 2000.

[MP05] B. Mourrain and J.-P. Pavone. Subdivision methods for solving polynomial equations. Technical Report 5658, INRIA Sophia-Antipolis, 2005.

[MRR05] B. Mourrain, F. Rouillier, and M.-F. Roy. *Bernstein's basis and real root isolation*, pages 459–478. Mathematical Sciences Research Institute Publications. Cambridge University Press, 2005.

[Pav04] J.P. Pavone. *Auto-intersection de surfaces pamatrées réelles*. PhD thesis, Université de Nice Sophia-Antipolis, 2004.

[Sed89] T. Sederberg. Algorithm for algebraic curve intersection. *Computer-Aided Design*, 21:547–554, 1989.

[SP93] E. C. Sherbrooke and N. M. Patrikalakis. Computation of the solutions of nonlinear polynomial systems. *Comput. Aided Geom. Design*, 10(5):379–405, 1993.

[TG92] J.-Ph. Thirion and A. Gourdon. The 3D Marching Lines Algorithm and its Applications to Crest Lines Extraction. Technical Report 1672, INRIA, 1992.

[TG95] J.-Ph. Thirion and A. Gourdon. Computing the Differential Characteristics of Isointensity Surfaces. *Computer Vision and Image Understanding*, 61(2):190–202, 1995.