Topologically certified approximation of umbilics and ridges on polynomial parametric surface

Frédéric Cazals, Jean-Charles Faugère, Marc Pouget, Fabrice Rouillier

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Topologically certified approximation of umbilics and ridges on polynomial parametric surface

Frédéric Cazals — Jean-Charles Faugère — Marc Pouget — Fabrice Rouillier

N° 5674

Septembre 2005

Thème SYM

ISSN 0249-6399 ISRN INRIA/RR--5674--FR+ENG

INRIA
SOPHIA ANTIPOLIS

Topologically certified approximation of umbilics and ridges on polynomial parametric surface

Frédéric Cazals [*] , Jean-Charles Faugère [†] , Marc Pouget [‡] , Fabrice Rouillier [§]

Thème SYM —Systèmes symboliques
Projets Geometrica et Salsa

Rapport de recherche n° 5674 —Septembre 2005 —36 pages

Abstract: Given a smooth surface, a blue (red) ridge is a curve along which the maximum (minimum) principal curvature has an extremum along its curvature line. Ridges are curves of *extremal* curvature and encode important informations used in surface analysis or segmentation. But reporting the ridges of a surface requires manipulating third and fourth order derivatives —whence numerical difficulties. Additionally, ridges have self-intersections and complex interactions with the umbilics of the surface —whence topological difficulties.

In this context, we make two contributions for the computation of ridges of polynomial parametric surfaces. First, by instantiating to the polynomial setting a global structure theorem of ridge curves proved in a companion paper, we develop the first certified algorithm to produce a topological approximation of the curve $\mathscr{P}$ encoding all the ridges of the surface. The algorithm exploits the singular structure of $\mathscr{P}$ —umbilics and purple points, and reduces the problem to solving zero dimensional systems using Gröbner basis. Second, for cases where the zero-dimensional systems cannot be practically solved, we develop a certified plot algorithm at any fixed resolution.
These contributions are respectively illustrated for Bezier surfaces of degree four and five.

Key-words: Ridges, Differential Geometry, Computer Algebra.

[*] INRIA Sophia-Antipolis, Geometrica project
[†] INRIA Rocquencourt, Salsa project
[‡] INRIA Sophia-Antipolis, Geometrica project
[§] INRIA Rocquencourt, Salsa project

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

Approximation certifiée des ombilics et des ridges d'une surface parametrée polynomiale

Résumé : Étant donnée une surface lisse, un *ridge* bleu (rouge) est une courbe le long de laquelle la courbure principale maximum (minimum) a un extremum en suivant sa ligne de courbure. Les ridges sont des lignes d'extrêmes de courbure et codent des informations importantes utilisées en segmentation, recalage, comparaison et analyse de surfaces. Cependant, reporter les ridges d'une surface nécessite l'évaluation de dérivées d'ordre trois et quatre —d'où des difficultés numériques. De plus, les ridges s'auto-intersectent et interagissent de façon complexe avec les ombilics de la surface —d'où des difficultés de nature topologique.

Dans ce contexte, ce papier propose deux contributions. D'une part, en instantiant pour les surfaces paramétrées de façon polynomiale un théorème de structure globale des ridges établi dans un papier joint, nous développons un algorithme produisant une approximation certifiée de la courbe $\mathscr{P}$ codant les ridges de la surface. Cet algorithme exploite la structure singulière de $\mathscr{P}$, et repose sur la résolution de systèmes zéro dimensionnels. D'autre part, pour les cas où les systèmes zéro dimensionnels ne peuvent être effectivement résolus, nous développons un algorithme de tracé certifié à une résolution donné.
Ces deux algorithmes sont illustrés sur des surfaces de Bézier de degrés quatre et cinq respectivement.

Mots-clés : Ridges et Extrêmes de courbure, Géométrie Différentielle, Calcul algébrique.

1 Introduction

1.1 Ridges as curves on smooth surfaces

Convinced the basic principals of esthetic beauty could be revealed by mathematical principles, Felix Klein drew the so-called parabolic curves on the face of the Apollo of Belvedere [HCV52]. Even though such an attempt may appear as desperately optimistic, Klein's intuition paved the alley aiming at bridging the gap between shape representations and shape perception, and several curves drawn on surfaces nowadays participate to the *understanding* of surfaces embedded in $\mathbb{R}^3$. For applications ranging from the design of class A surfaces in the automotive industry to medical imaging, example such curves are reflection lines, curvature lines or *ridges*. An informal introduction to these objects can be found in the treatise [Koe90].

Given a smooth surface, a *ridge* consists of the points where one of the principal curvatures has an extremum along its curvature line. Reporting ridges of a surface is a challenging task stemming from the fact that ridges witness third order differential properties. First, given a surface known analytically or through a mesh, computing or estimating high order derivatives is a numerically a difficult task. Second, curves defined from high order properties are inherently unstable, and classifying their transitions requires delicate singularity analysis [Por01]. Finally, ridges cannot be reported on their own since they have a complex interplay with umbilics and curvature lines, whence topological difficulties.

1.2 Previous work

Given the previous difficulties, no algorithm reporting ridges in a certified fashion has been developed as of today. Most contributions deal with sampled surfaces known through a mesh, and a complete review of these contributions can be found in [CP05]. In the following, we focus on contributions related to parametric surfaces.

Reporting umbilics. Umbilics of a surface are always traversed by ridges, so that reporting ridges faithfully requires reporting umbilics. To do so, Morris [Mor90] minimizes the function $k_1 - k_2$, which vanishes exactly at umbilics. Meakawa et al. [MWP96] define a polynomial system whose roots are the umbilics. This system is solved with the *rounded interval arithmetic projected polyhedron method*. This algorithm uses specific properties of the Bernstein basis of polynomials and interval arithmetic. The domain is recursively subdivided and a set of boxes containing the umbilics is output, but neither existence nor uniqueness of an umbilic in a box is guaranteed.

Reporting ridges. The only method dedicated to parametric surfaces we are aware of is that of Morris [Mor90, Mor96]. The parametric domain is triangulated and zero crossings are sought on edges. Local orientation of the principal directions are needed but only provided with a heuristic. This enables to detect crossings assuming (i)there is at most one such crossing on an edge (ii)the orientation of the principal directions is correct. As this simple algorithm fails near umbilics, these points are located first and crossings are found on a circle around the umbilic.

1.3 Contributions

Consider a parameterized surface $\Phi(u,v)$, the parameterization being *polynomial* with rational coefficients. Let $\mathscr{P}$ be the curve encoding the ridges of $\Phi(u,v)$. We aim at studying $\mathscr{P}$ on the compact box domain $\mathscr{D} = [a,b] \times [c,d]$. We know from [CFPR05] that curve $\mathscr{P}$ is a singular curve defined by a polynomial P with rational coefficients. Ideally, we would like to produce a topologically certified approximation of $\mathscr{P}$. Given a real algebraic curve, the standard way to approximate it consists of resorting to the Cylindrical Algebraic Decomposition (CAD). Running the CAD requires computing singular points and critical points of the curve —points with a horizontal tangent. Theoretically, these points are defined by zero-dimensional systems. Practically, because of the high degree of the polynomials involved, the calculations may not go trough. Replacing the bottlenecks of the CAD by a resolution method adapted to the singular structure of $\mathscr{P}$, we make two contributions:

1. First, we develop an algorithm producing a graph $\mathscr{G}$ embedded in the domain $\mathscr{D}$, which is isotopic to the curve $\mathscr{P}$ of ridges in $\mathscr{D}$. The key points are twofold:

   (a) no generic assumption is required, i.e. several critical or singular points may have the same horizontal projection;

   (b) no computation with algebraic numbers is involved.

   If singular and critical points can be computed —in a reasonable amount of time, the method is effective.

2. Second, if singular or critical points cannot computed in a reasonable amount of time, we develop a certified plot algorithm at any fixed resolution: we subdivide $\mathscr{D}$ into *pixels*, a pixel being lit iff the curve intersects it.

1.4 paper overview

Notations are presented in section 2. The pre-requisites on ridges and on the main tools used by our algorithms are recalled in sections 3 and 4. The main difficulties of CAD based algorithms are discussed in section 5. Certified topological approximation and the certified plot are developed in sections 6 and 7, and illustrated in section 8. Appendix 10 provides the algebraic pre-requisites.

2 Notations

For a bivariate function $f(u,v)$, the partial derivatives are denoted with indices, for example $f_{uuv} = \frac{\partial^3 f}{\partial^2 u \partial v}$. The quadratic form induced by the second derivatives is denoted $f_2(u,v) = f_{uu}u^2 + 2f_{uv}uv + f_{vv}v^2$. The discriminant of this form is denoted $\delta(f_2) = f_{uv}^2 - f_{uu}f_{vv}$. The cubic form induced by the third derivatives in denoted $f_3(u,v) = f_{uuu}u^3 + 3f_{uuv}u^2v + 3f_{uvv}uv^2 + f_{vvv}v^3$. The discriminant of this form is denoted $\delta(f_3) = 4(f_{uuu}f_{uvv} - f_{uuv}^2)(f_{uuv}f_{vvv} - f_{uvv}^2) - (f_{uuu}f_{vvv} - f_{uuv}f_{uvv})^2$.

Let f be a real bivariate polynomial and $\mathscr{F}$ the real algebraic curve defined by f. A point $(u,v) \in \mathbb{C}^2$ is called

- a singular point of $\mathscr{F}$ if $f(u,v)=0,\ f_u(u,v)=0$ and $f_v(u,v)=0$;
- a critical point of $\mathscr{F}$ if $f(u,v)=0$ and $f_u(u,v)=0$ and $f_v(u,v)\neq 0$ (such a point has an horizontal tangent);
- a regular point of $\mathscr{F}$ if $f(u,v)=0$ and it is neither singular nor critical.

If the domain $\mathscr{D}$ of study is a subset of $\mathbb{R}^2$, by *fiber* we refer to a cross section of this domain at a given ordinate or abscissa.

3 The implicit structure of ridges, and study points

3.1 Implicit structure of the ridge curve

As shown in [CFPR05], the ridge curve $\mathscr{P}$ is defined by the bivariate polynomial $P(u,v)$. In the same paper, we also introduce polynomials a, b, a' and b', together with $Sign_{ridge}$, p_2, which are functions of the curvatures of the surface and their first derivatives. These functions characterize the singularities and the colors of $\mathscr{P}$ in the following sense [CFPR05]:

Theorem. 1 *Consider a smooth parametric surface whose parameterization is denoted $\Phi(u,v)$. There exists polynomial functions P, p_2 and $Sign_{ridge}$ so that the set of blue ridges union the set of red ridges union the set of umbilics has equation $P=0$. In addition, for a point of this set $\mathscr{P}$, one has:*

- *If $p_2=0$, the point is an umbilic.*
- *If $p_2\neq 0$ then:*
  - *if $Sign_{ridge}=-1$ then the point is a blue ridge point,*
  - *if $Sign_{ridge}=+1$ then the point is a red ridge point,*
  - *if $Sign_{ridge}=0$ then the point is a purple point.*

Following the theoretical study performed in [CFPR05], the only assumption made is that the surface admits generic ridges in the sense that real singularities of $\mathscr{P}$ satisfy the following conditions:

- Real singularities of $\mathscr{P}$ are of multiplicity at most 3.
- Real singularities of multiplicity 2 are called purple points. They satisfy the system $S_p=\{a=b=a'=b'=0,\ \delta(P_2)>0,\ p_2\neq 0\}$. In addition, two real branches of $\mathscr{P}$ are passing through a purple point.
- Real singularities of multiplicity 3 are called umbilics. They satisfy the system $S_u=\{p_2=0\}=\{p_2=0,P=0,P_u=0,P_v=0\}$. In addition, if $\delta(P_3)$ denote the discriminant of the cubic of the third derivatives of P at an umbilic, one has:

– if $\delta(P_3) > 0$, then the umbilic is called a 3-ridge umbilic and three real branches of $\mathscr{P}$ are passing through the umbilic with three distinct tangents;
– if $\delta(P_3) < 0$, then the umbilic is called a 1-ridge umbilic and one real branch of $\mathscr{P}$ is passing through the umbilic.

As we shall see in section 6, these conditions are checked during the processing of the algorithm.

3.2 Study points and zero dimensional systems

A recalled in introduction, the most demanding task to certify to topology of a real algebraic curve consists of isolating its real singular and critical points. For our problem, the singular and critical points of $\mathscr{P}$ have a well known structure which can be exploited. More precisely: we successively isolate umbilics, purple points and critical points. As a system defining one set of these points also include as solution the points of the previous system, we use a localization method to simplify the calculations (see theorem 2). The points reported at each stage are characterized as roots of a zero-dimensional system —a system with a finite number of complex solutions, together with the number of half-branches of the curve connected to each point. In addition, points on the border of the domain of study need a special care. This setting leads to the definition of *study points*:

Definition. 1 *Study points are points in $\mathscr{D}$ which are*

- *real singularities of $\mathscr{P}$, denoted $S_u \cup S_p$, with $S_u = S_{1R} \cup S_{3R}$ and*
  – $S_{1R} = \{p_2 = P = P_u = P_v = 0, \delta(P_3) < 0\}$ *(1-ridge umbilics)*
  – $S_{3R} = \{p_2 = P = P_u = P_v = 0, \delta(P_3) > 0\}$ *(3-ridges umbilics)*
  – $S_p = \{a = b = a' = b' = 0,\ \delta(P_2) > 0,\ p_2 \neq 0\} = \{a = b = a' = b' = 0,\ \delta(P_2) > 0\} \setminus S_u$ *(purple points)*
- *real critical points of $\mathscr{P}$ in the v-direction (i.e. points with a horizontal tangent which are not singularities of $\mathscr{P}$) defined by the system*
  $S_c = \{P = P_u = 0, P_v \neq 0\}$ *(critical points);*
- *intersections of $\mathscr{P}$ with the left and right sides of the box $\mathscr{D}$ satisfying the system*
  $S_b = \{P(a,v) = 0,\ v \in [c,d]\} \cup \{P(b,v) = 0,\ v \in [c,d]\}$. *Such a point may also be critical or singular.*

4 Some Algebraic tools for our method

In this section, we sketch the two algebraic methods ubiquitously called by our algorithms. More details are provided in appendix 10.

4.1 Zero dimensional systems

In our algorithms, we will need to represent solutions of zero-dimensional systems depending on two or more variables. We use the so called Rational Univariate Representation of the roots [Rou99], which can be viewed as a univariate equivalent to the studied system.

Given a zero-dimensional system $I =< p_1, \ldots, p_s >$ where the $p_i \in \mathbb{Q}[X_1, \ldots, X_n]$, a Rational Univariate Representation of $\mathrm{V}(I)$ has the following shape : $f_t(T) = 0, X_1 = \frac{g_{t,X_1}(T)}{g_{t,1}(T)}, \ldots, X_n = \frac{g_{t,X_n}(T)}{g_{t,1}(T)}$, where $f_t, g_{t,1}, g_{t,X_1}, \ldots, g_{t,X_n} \in \mathbb{Q}[T]$ (T is a new variable). It is uniquely defined w.r.t. a given polynomial t which separates $V(I)$ (injective on $V(I)$), the polynomial f_t being necessarily the characteristic polynomial of m_t (the multiplication operator by the polynomial t) in $\mathbb{Q}[X_1, \ldots, X_n]/I$ [Rou99]. The RUR defines a bijection between the roots of the system I and those of f_t preserving the multiplicities and the real roots :

$$\begin{array}{ccc} \mathrm{V}(I)(\cap\mathbb{R}) & \approx & \mathrm{V}(f_t)(\cap\mathbb{R}) \\ \alpha = (\alpha_1, \ldots, \alpha_n) & \rightarrow & t(\alpha) \\ (X_1(\alpha) = \frac{g_{t,X_1}(t(\alpha))}{g_{t,1}(t(\alpha))}, \ldots, X_n(\alpha) = \frac{g_{t,X_n}(t(\alpha))}{g_{t,1}(t(\alpha))}) & \leftarrow & t(\alpha) \end{array}$$

There exists several ways for computing a RUR. One can use the strategy from [Rou99] which consists of computing a Gröbner basis of I and then to perform linear algebra operations to compute a separating element as well as the full expression of the RUR. The Gröbner basis computation can also be replaced by the generalized normal form from [MT05]. There exists more or less certified alternatives such as the Geometrical resolution [GLS01] (it is probabilistic since the separating element is randomly chosen and its validity is not checked, one also loose the multiplicities of the roots) or resultant based strategies such as [KOR05].

4.2 Univariate root isolation

This second tool is used to analyze fibers i.e. cross sections of $\mathscr{D}$ at a given ordinate or abscissa, and requires isolating roots of univariate polynomials whose coefficients are rational numbers or intervals. The method uses the Descartes rule and is fully explained in [RZ03]. The algorithm is based on a recursive subdivision of the initial interval. If exact computations with rationals are performed, the algorithm is proved to terminate —but such computations may be costly. However, the structure of our problem is such that certifications can be achieved using interval arithmetic rather than an exact arithmetic. To see how, given a polynomial $P = \sum_{i=0}^{n} a_i u^i$ with rational coefficients, assume we are given intervals $[l_i, r_i]$ enclosing the coefficients a_i. Representing a rational number by an interval amounts to approximating the number, and we shall assume the intervals' widths can be made arbitrarily small. The specifications of the algorithm [RZ03] in the case of polynomials with intervals as coefficients are the following :

- Input :
  - $P_{\varepsilon_1} = \sum_{i=0}^{n} [l_i, r_i] u^i$ a univariate polynomial with intervals $[l_i, r_i]$ of width less than ε_1 as coefficients. Notice that P_{ε_1} can be seen as a family of polynomials parameterized by the choice of a $(n+1)$-uple $(a_i)_{i=0\ldots n}$ with $a_i \in [l_i, r_i]$;

  - $[a,b]$ an interval for the variable u;
  - a precision ε_2 for the interval arithmetic computations. (This is the precision used to represent the intervals' boundaries.)

- Output :
  - a list L_i of intervals with rational bounds containing a unique real root of P_{ε_1} (that is any polynomial of the family has a unique real root in each of these intervals);
  - a list L_e of interval with rational bounds where no decision was possible —i.e. the Descartes rule of sign cannot be applied because signs of interval coefficients are not defined;
  - all the elements of L_i and L_e are intervals contained in $]a,b[$;
  - a real root of P_{ε_1} in $]a,b[$ is represented by an interval of L_i or L_e;

According to [RZ03], L_e is not empty only in one of the following situations :

- there exists a polynomial in the family of polynomial P_{ε_1} which has a multiple root in $]a,b[$;
- the precision ε_2 of the interval arithmetic used for the computation is not sufficient;

In the present article, we need to solve two kinds of problems : isolating all the solutions of a polynomial with rational coefficients; and isolating all the simple roots of a polynomial whose coefficients are intervals —of arbitrary length, knowing intervals which separate the multiple roots from the simple ones.

In the first case, the square free part of the polynomial is computed so that it has no multiple root. Hence both problems become equivalent if we use interval arithmetics: we need to isolate all roots of P on the interval $[a,b]$ containing no multiple roots of P. This can straightforwardly be done running the previous algorithm for P_{ε_1}. If the list L_e is empty we are done, otherwise the algorithm is run for $P_{\varepsilon_1/2}$ and $\varepsilon_2/2$. The termination of this process is proved in [RZ03].

4.3 About square-free polynomials

Many computations suppose that the considered polynomials are square-free, replacing them, when needed, by their square-free part. In our algorithms, we use interpolation based algorithms (multi-modular, Hensel lifting, etc.) in the univariate case as well as in the bivariate case (default strategy in most computer algebra systems). One key advantage of such methods is that they detect quickly that a polynomial is square-free by solving a simpler problem (univariate problem in the bivariate case, computation modulo a prime number in the univariate case). In particular, this first part of the computation can be used to test if a polynomial is square-free or not.

In the univariate case for example, if a polynomial is square-free modulo a prime number which do not divide its leading coefficient, then is is square-free.

Thus, using algorithms based on interpolation strategies, the overall cost of the computations for testing if a polynomial is square-free (or to compute its square-free part when it is not) is negligible

compared to the rest. (With a standard variant using the Euclidean algorithm, one would perform, in average, $O(d^2)$ binary operations, d being the degree of the polynomial.)

In the few cases where the computation of a square-free part is non trivial, the computing time for interpolation based methods is, in average, proportional to the size of the result and polynomial in the size (degree and coefficients sizes) of the input. For example, in the univariate case, a naive variant will run Euclid-e's algorithm modulo some prime numbers ($O(d^2)$ binary operations for a polynomial of degree d) until the product of these prime numbers exceeds the size of the coefficients in the result and finally will recover the result using the Chinese remainder theorem). Since the non trivial cases are few, these computations do not represent a blocking step in the whole algorithms of the present paper.

5 On the difficulty of approximating algebraic curves

The standard way to approximate algebraic curves is to resort to the CAD (Cylindrical Algebraic Decomposition). In this section, we recall the main steps of such strategies to approximate $\mathscr{P}$ and discuss the major difficulties of algorithms like [GVN02]. In section 6 we will show how to keep track of the specific implicit structure of ridges [CFPR05] to optimize the process.

The CAD has been introduced in [Col75]. Basically, a CAD adapted to a set of multivariate polynomials is a partition of the ambient space ($\mathbb{R}^n$ if the polynomials depends on n variables) into cells (connected subsets with a trivial topology) where the signs of the considered polynomials are all constant. Such a general method can be adapted to compute a graph reporting the topology of 2-D or 3-D curves —see respectively [GVN02] and [GLMT04]. The following give the basic principles of the method.

Given any implicit curve $P(u,v) = 0$, one consider P as a univariate polynomial in u (or v) and study the values of v (or u) for which the number of roots of P varies. When v varies, a root of P may "go to infinity" or become singular. The first case corresponds to the values of v for which the leading coefficient of P vanishes and the second case to the values of v for which the discriminant wrt u vanishes (v-coordinates of singular points and critical points wrt the projection on the v-axis). Both sets of values can be expressed as the (real) roots of a univariate polynomial $c(v)$ which can be explicitly computed from P. When restricted to a cylinder between the fibers over two consecutive real roots of $c(v)$, $P(u,v) = 0$ has the topology of a trivial covering. Using the CAD to approximate an algebraic curve requires mainly three stages.

In the following, we consider there is no (horizontal) asymptotes: no roots to the leading coefficient wrt the variable u. A change of coordinate is always able to transform the curve in such a configuration and anyways, as we will see later, one can easily avoid such an assumption when studying the curve in a compact box.

First stage. Produce the v-coordinates of the singular points and the critical points wrt the projection on the v-axis. These v-coordinates are the real roots $\alpha_1, .., \alpha_s$ of the discriminant of P w.r.t. u, denoted by $Crit_u$.

Second stage. Consists of building the fiber above each v-coordinate of critical and singular points. This calculation involves polynomials whose coefficients are algebraic numbers, so as to isolate the solutions in each fiber and at least discriminate the multiple points from the simple points. More formally, this requires solving:

- **L-1a**. Report the simple real roots of $P(u,\alpha_i)$, $i = 1..s$;
- **L-1b**. Report the multiple real roots of $P(u,\alpha_i)$, $i = 1..s$ and their multiplicity;

Note that the u-coordinates of the critical and singular points are the multiple roots of the polynomials $P(u,\alpha_i)$.

Third stage. Finally, the connection phase consists of connecting points from fibers. This requires:

- Finding the real roots of $P(u,\beta_i)$, $i = 0..s$, β_i being any arbitrary point in $]\alpha_i,\alpha_{i+1}[$, with the convention $\alpha_0 = -\infty$ and $\alpha_{s+1} = +\infty$.
- Finding the number of half branches which connect to the real roots of $P(u,\alpha_i)$, $i = 1..s$ in order to connect each real root of $P(u,\beta_i)$, $i = 0..s$, to a real root of $P(u,\alpha_i)$, $i = 0..s$ and to a real root of $P(u,\alpha_{i+1})$, $i = 0..s$.

These three stages face four major difficulties.

Computing the discriminant $Crit_u$. For large polynomials, the first difficulty comes from the calculation of the discriminant wrt u, which may be unpractical. In section 6, we shall face this difficulty by sequentially reporting study points independently and thus decreasing strongly the degrees of the involved polynomials.

Isolating the roots of $P(u,\alpha_i)$. The second difficulty comes from the isolation of the roots of $P(u,\alpha_i)$, α_i being a real root of $Crit_u$ and the computation of their multiplicities. Since α_i is an algebraic number, there are three main ways to certify such a computation:

- (i) use an approximated representation of the real algebraic numbers involved such as floating point numbers or intervals. It then becomes possible to isolate some of the simple real roots of $P(u,\alpha_i)$. Managing correctly the precision of the approximations as well as the numerical errors during the computations, one can hope being able to isolate all the simple real roots of $P(u,\alpha_i)$. However, we are not aware of any general implementation of this strategy.
- (ii) use an exact representation of the real algebraic numbers involved – for example an interval and a polynomial to refine it to an arbitrary precision, or Thom's coding of the roots– and apply classical algorithms with the induced arithmetic such as Sturm-habicht sequences and sub-resultants algorithms for computing the roots of $P(u,\alpha_i)$ [BPR03] for details. For large problems, the size of the polynomials involved and the cost of exact arithmetic over real algebraic numbers prevents this solution from being practical.

- (iii) solve the zero-dimensional system $P(u,v)=0, Crit_u(v)=0$. This third strategy computes directly the $2D$ representation of the singular and critical points. On the one hand, the basic computations may be more difficult, but in the other hand, the operations using real algebraic numbers are simple to perform (only evaluates polynomial at real algebraic numbers). In other words, one may increase the number of arithmetic operations but decrease their cost.

Genericity assumption. In order to avoid difficult computations in the cases where it is not possible to compute the multiplicities of the roots at the second stage of the previous algorithm or to perform computations using the strategy (ii), most recent algorithms such as [GVN02] or [GLMT04] suppose that the curve is in *generic position* w.r.t. the coordinate system —which means that each fiber $P(u,\alpha_i)=0$ contains a unique critical or singular point. This implies that each polynomial $P(u,\alpha_i)$ has one and only one multiple root. In such a situation, points of consecutive fibers corresponding to α_i, α_{i+1} are easily connected through points in an intermediate witness fiber [GVN02]. (One-to-one connections are made between regular points, and one is left with connections between the unique critical point of each fiber α_i, α_{i+1} with the remaining points of the intermediate fiber.)

The genericity hypothesis can be met through a linear change of variables. The algorithm described in [GVN02] or [GLMT04] provides such a generic position for the curve and this eases the computation of the roots of $P(u,\alpha_i)$. In our case, the studied curves are usually in generic position and, anyways, it is also true that a randomly chosen linear change of coordinates will put the curve in generic position with a probability one. But checking deterministically that a curve is in generic position may be more costly than all the other operations and is required if we pretend to implement a certified algorithm. For example, in [GVN02] such a test requires the computation of some principal Sturm-Habicht coefficients of P wrt u which is mainly as costly as using strategy (ii) for computing the fibers over the α_i.

Moreover, one can choose not to fully certify the *generic position* (by performing a random linear change of variables) and thus use strategy (i) to compute the fibers $P(u,\alpha_i)=0$. This is suggest by [GVN02] but the way it is done is again not certified since they use a purely numerical function (`fsolve` from MAPLE software), which can not make the distinction between a cluster of 3 simple roots and a triple point.

In addition, replacing a variable by a linear form in a huge bivariate polynomial may be a difficult task. The sizes of the coefficients increases so that all the computations except perhaps those involving real algebraic numbers become more difficult.

All the above problems are illustrated by the example from section 8 :

- we were not able to perform the generic position test;
- we were not able to compute the roots of some $P(u,\alpha_i)$ even under the generic position assumption —in which case $P(u,\alpha_i)$ has exactly one multiple root which allows to use several optimization tricks.

6 Certified topological approximation

In this section, we circumvent the difficulties of the CAD and develop a certified algorithm to compute the topology of $\mathscr{P}$.

6.1 Output specification

Definition. 2 *Let $\mathscr{G}$ be a graph whose vertices are points of $\mathscr{D}$ and edges are non-intersecting straight line-segments between vertices. Let the topology on $\mathscr{G}$ be induced by that of $\mathscr{D}$. We say that $\mathscr{G}$ is a topological approximation of the ridge curve $\mathscr{P}$ on the domain $\mathscr{D}$ if $\mathscr{G}$ is ambient isotopic to $\mathscr{P} \cap \mathscr{D}$ in $\mathscr{D}$.*

More formally, there exists a function $F : \mathscr{D} \times [0,1] \longrightarrow \mathscr{D}$ such that:

- *F is continuous;*
- *$\forall t \in [0,1]$, $F_t = F(.,t)$ is an homeomorphism of $\mathscr{D}$ onto itself;*
- *$F_0 = Id_{\mathscr{D}}$ and $F_1(\mathscr{P} \cap \mathscr{D}) = \mathscr{G}$.*

Note that homeomorphic approximation is weaker and our algorithm using a cylindrical decomposition technique actually gives isotopy. In addition, our construction allows to identify singularities of $\mathscr{P}$ to a subset of vertices of $\mathscr{G}$ while controlling the error on the geometric positions. We can also color edges of $\mathscr{G}$ with the color of the ridge curve it is isotopic to. Once this topological sketch is given, one can easily compute a more accurate geometrical picture.

6.2 Method outline

Taking the square free part of P, we can assume P is square free. We can also assume $\mathscr{P}$ has no part which is a horizontal segment —parallel to the u-axis. Otherwise this means that a whole horizontal line is a component of P. In other words, the content of P wrt u is a polynomial in v and we can study this factor separately and divide P by this factor. Eventually, to get the whole topology of the curve, one has to merge the components.

6.2.1 The algorithm

Our algorithms consists of the following five stages:

1. **Isolating study points.** Study point are isolated in $2D$ with rational univariate representations (RUR). Study points within a common fiber are identified.
2. **Regularization of the study boxes.** The boxes of study points are reduced so as to have the right number of intersections between their border and $\mathscr{P}$. This number is 6 for 3-ridge umbilic, 2 for 1-ridge umbilic, 4 for a purple, 2 for others. Moreover, boxes are reduced so as to have crossings on the top and bottom sides only. Define the number of branches coming from the top and the bottom.

3. **Computing regular points in study fibers.** In each fiber of a study point, the u-coordinates of intersection points with $\mathscr{P}$ other than study points are computed.
4. **Adding intermediate rational fibers.** Add rational fibers between study points fibers and isolate the u-coordinates of intersection points with $\mathscr{P}$.
5. **Performing connections.** This information is enough to perform the connections. Consider the cylinder between two consecutive fibers, the number of branches connected from above the lower fiber is the same than the number of branches connected from below the higher fiber. Hence there is only one way to perform connections with non-intersecting straight segments.

6.2.2 Key points wrt CAD based algorithms

Our algorithm avoids the difficulties of CAD methods, and the following comments are in order.

Zero-dimensional systems versus the discriminant $Crit_u$. Instead of computing the v-coordinates of all critical and singular points at once, as done by the CAD, study points are sequentially computed directly in $2D$, together with the information required to derive the graph $\mathscr{G}$.

Isolating the roots of $P(u,\alpha_i)$. The isolation of roots of polynomials whose coefficients are algebraic numbers does not arise since study points are isolated in $2D$ in the first place. We only use the isolation of simple roots of polynomial whose coefficients are intervals. However, we do need to characterize the presence of multiple study points in the same fibers, an information required by the connection process.

The connection phase without genericity assumption. Algorithms derived from the CAD have problems to perform the proper connection between to consecutive fibers if these fibers contain more than one critical or singular point. We alleviate this limitation using the information on the number of half-branches connected to the point. This number is equal to 6 for a 3-ridge umbilic, 4 for a purple point and 2 otherwise. These informations are sufficient to build the approximation $\mathscr{G}$.

Complexity-wise. The advantage of the strategy is to iteratively split the problem into simpler ones, and to solve the sub-problems directly in $2D$. The main drawback of the method may be the arithmetic asymptotic complexity upper bounds of some of the tools we use to compute and certify the solutions of zero-dimensional systems (see section 10).

Let F be a set of polynomials, $mindeg(F)$ (resp. $maxdeg(F)$) the minimal (resp. maximal) degree of a polynomial which belongs to F. According to [Laz83], in the case of two variables, a Gröbner basis for a Degree ordering has at most $mindeg(F)+1$ polynomials of degree less than $2maxdeg(F)-1$ so that modern strategies like [Fau02] will compute it in a polynomial time. In short, the algorithm is faster than inverting a matrix whose number of columns is bounded by the number of possible monomials while the number of rows is bounded by the number of polynomials. Since all the other algorithms we use are also polynomial (RUR, isolation, etc.) in their input, the full strategies we use are still polynomial.

We would like to point out that we need to compute, certify and provide numerical approximations with an arbitrary precision of the real roots of huge systems. Up to our knowledge, the tools we used are, at least in practice, the most efficient ones with such specifications.

6.3 Step 1. Isolating study points

The method to identify these study points is to compute a RUR of the system defining them. Details of the method are exposed in section 10.

The less the number of solutions of a system, the easier the computation of the RUR. Hence, we use a localization method to decompose the computation of the different types of study points —see section 10.5. More precisely, we sequentially solve the following systems:

1. The system S_u from which the sets S_{1R} and S_{3R} are distinguished by evaluating the sign of $\delta(P_3)$.
2. The system S_p for purple points.
3. The system S_c for critical points.
4. The system S_b for border points, that is intersections of $\mathscr{P}$ with the left and right sides of the box $\mathscr{D}$. Solving this system together with one of the previous identifies border points which are also singular or critical.

Selecting only points belonging to $\mathscr{D}$ reduces to adding inequalities to the systems and is well managed by the RUR. According to 10.5, solving such systems is equivalent to solving zero-dimensional systems without inequalities when the number of inequations remains small compared to the number of variables.

The RUR of the study points provides a way to compute a box around each study point q_i which is a product of two intervals $[u_i^1;u_i^2]\times[v_i^1;v_i^2]$ (see section 10). The intervals can be as small as desired.

The computation of the RUR of one of these systems begins with testing if the polynomial v is separating for the system (see 10 for the definition of a separating element). Note that if it is so, the solution points are in generic position wrt the projection on the v-axis, that is each fiber of a point contains no other point of this system. In any case, we compute the square-free part of the minimal or characteristic polynomial of the multiplication by v modulus the ideal generated by the system (nothing to do when v is separating since it is the first polynomial of the RUR) : its roots are exactly all the v-coordinates of the solutions of the system.

Until now, we only have separate informations on the different systems. In order to identify study points having the same v-coordinate, we need to cross these informations. First we compute isolation intervals for all the v-coordinates of all the study points together, denote I this list of intervals. If two study points with the same v-coordinate are solutions of two different systems, the gcd of polynomials enable to identify them:

- Initialize the list I with all the isolation intervals of all the v-coordinates of the different systems.

- Let A and B be the square free polynomials defining the v-coordinates of two different systems, and I_A, I_B the lists of isolation intervals of their roots. Let $C = gcd(A,B)$ and I_C the list of isolation intervals of its roots. One can refine the elements of I_C until they intersect only one element of I_A and one element of I_B. Then replace these two intervals in I by the single interval which is the intersection of the three intervals. Do the same for every pair of systems.

- I then contains intervals defining different real numbers in one-to-one correspondence with the v-coordinates of the study points. It remains to refine these intervals until they are all disjoint.

Second, we compare the intervals of I and those of the 2d boxes of the study points. Let two study points q_i and q_j be represented by $[u_i^1;u_i^2]\times[v_i^1;v_i^2]$ and $[u_j^1;u_j^2]\times[v_j^1;v_j^2]$ with $[v_i^1;v_i^2]\cap[v_j^1;v_j^2]\neq\varnothing$. One cannot, a priori, decide if these two points have the same v-coordinate or if a refinement of the boxes will end with disjoint v-intervals. On the other hand, with the list I, such a decision is straightforward. The boxes of the study points are refined until each $[v_i^1;v_i^2]$ intersects only one interval $[w_i^1;w_i^2]$ of the list I. Then two study points intersecting the same interval $[w_i^1;w_i^2]$ are in the same fiber.

Finally, one can refine the u-coordinates of the study points with the same v coordinate until they are represented with disjoint intervals since, thanks to localizations, all the computed points are distinct.

Checking genericity conditions of section 3.1.

First, real singularities shall be the union of purple and umbilical points, this reduces to compare the systems for singular points and for purple and umbilical points. Second, showing that $\delta(P_3)\neq 0$ for umbilics and $\delta(P_2)>0$ for purple points reduces to sign evaluation of polynomials at the roots of a system (see section 10.5).

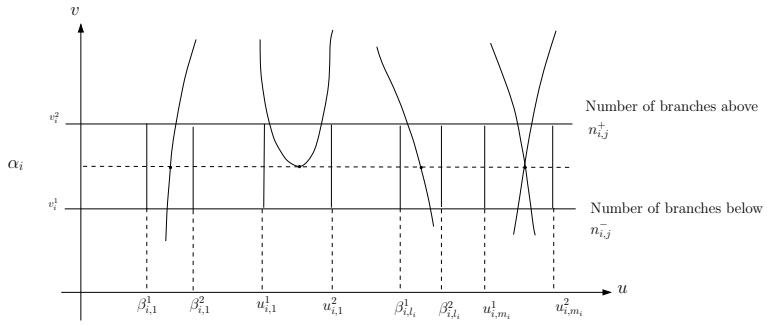

Figure 1: Notations for a fiber involving several critical/singular points: $u_{i,j}^{1(2)}$ are used for study points, $\beta_{i,j}^{1(2)}$ for simple points.

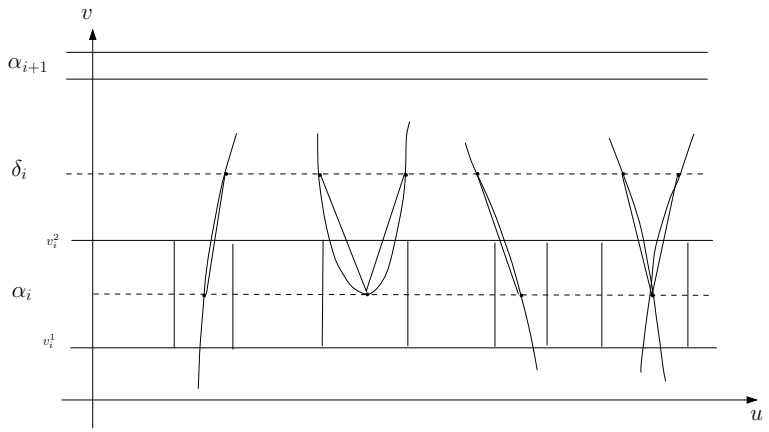

Figure 2: Performing connections

6.4 Step 2. Regularization of the study boxes

At this stage, we have computed all the v-coordinates $\alpha_1, ..., \alpha_s$ of all the study points $\{q_{i,j}, i = 1 \ldots s, j = 1 \ldots m_i\}$ by means of isolating intervals $[v_i^1; v_i^2], i = 1..s$. We have isolated u-coordinates of the m_i study points in each fiber α_i by the intervals $[u_{i,j}^1; u_{i,j}^2], j = 1..m_i$) and we know (section 6.3) the number of branches which should be connected to each of them.

Note that an intersection (in $\mathscr{D}$) of $\mathscr{P}$ with a fiber $v = v_0$ which is not a study point is a regular point of $\mathscr{P}$. Hence, the u-coordinate of this point is a simple root of the univariate polynomial $P(u, v_0)$. The first additional computation we need to do is to ensure that for each fiber $P(u, \alpha_i) = 0$ the isolating boxes of the study points do not contain any regular point. It is sufficient to count the number of intersection points between $\mathscr{P}$ and the border of the isolating box to detect if the box contains or not a regular point : such a computation remains to solve 4 univariate polynomials with rational coefficients and can be done efficiently using the algorithm from section 4.2. Precisely, if the study point is a critical point, the isolating box contains no regular point if and only if the number of points in the computed intersection is 2 (2 for a 1-ridge, 6 for a 3-ridge, 4 for a purple, 2 for a non-critical nor singular border point). If the box contains a regular point, one use the RUR to refine the isolating box and perform again the test : after a finite number of steps, each study point will be represented by an isolating box which do not contain any regular point.

In addition, reducing boxes if necessary, we can also assume the intersections on the border of the isolation boxes only occur on the top or the bottom sides of the boxes (that is the sides parallel to the u-axis). This allows to compute the number of half-branches connected to the top and to the bottom of each study point. For the special case of border points, one has to compute the number of branches inside the domain $\mathscr{D}$ only.

6.5 Step 3. Computing regular points in study fibers

We now compute the regular points in each fiber $P(u,\alpha_i)=0$. Computing the regular points of each fiber is now equivalent to computing the roots of the polynomials $P(u,\alpha_i)$ outside the intervals representing the u-coordinates of the study points (which contain all the multiple roots of $P(u,\alpha_i)$).

Denote $[u^1_{i,j};u^2_{i,j}], j=1..m_i$ the intervals representing the u-coordinates of the study points on the fiber of α_i and $[v_i^1,v_i^2]_\varepsilon$ an interval of length ε containing (strictly) α_i and no other $\alpha_j, j\neq i$. Substituting v by $[v_i^1,v_i^2]_\varepsilon$ in $P(u,v)$ gives a univariate polynomial with intervals as coefficients we denote $P(u,[v_i^1,v_i^2]_\varepsilon)$. We apply the algorithm 4.2 for the polynomial $P(u,[v_i^1,v_i^2]_\varepsilon)$ and on the domain $\cup_{j=1}^{m_i}[u^1_{i,j};u^2_{i,j}]$. We know that for every $v_i\in[v_i^1,v_i^2]_\varepsilon$ the polynomial $P(u,v_i)$ has no multiple roots on this domain. Hence the algorithm will return intervals $[\beta^1_{i,j};\beta^2_{i,j}], j=1\dots l_i$ such that for ε sufficiently small and $\forall v_i\in[\alpha_i]_\varepsilon$, each root of $P(u,v_i)$ belonging to $[a,b]\setminus\cup_{j=1}^{m_i}[u^1_{i,j};u^2_{i,j}]$ is contained in a unique $[\beta^1_{i,j};\beta^2_{i,j}]$.

We have isolated, along each fiber, a collection of points $s_{i,j}$, $i=1\dots s,\ j=1,\dots,m_i+l_i$, which are either study points or regular points of $\mathscr{P}$. Each such point is isolated in a box i.e. a product of intervals and comes with two integers $(n^+_{i,j},n^-_{i,j})$ denoting the number of branches in $\mathscr{D}$ connected from above and from below.

6.6 Step 4. Adding intermediate rational fibers

Consider now an intermediate fiber, i.e. a fiber associated with $v=\delta_i$ $i=1\dots s-1$, with δ_i a rational number in-between the intervals of isolation of two consecutive values α_i and α_{i+1}. If the fibers $v=c$ or $v=d$ are not fibers of study points, then they are added as fibers δ_0 or δ_s.

Getting the structure of such fibers amounts to solving a univariate polynomial with rational coefficients, which is done using the algorithm described in section 4.2. Thus, each such fiber also comes with a collection of points for which ones knows that $n^+_{i,j}=n^-_{i,j}=1$. Again, each such point is isolated in a box.

6.7 Step 5. Performing connections

We thus obtain a full and certified description of the fibers: all the intersection points with $\mathscr{P}$ and their number of branches connected. We know, by construction, that the branches of $\mathscr{P}$ between fibers have empty intersection. The number of branches connected from above a fiber is the same than the number of branches connected from below the next fiber. Hence there is only one way to perform connections with non-intersecting straight segments. More precisely, vertices of the graph are the centers of isolation boxes, and edges are line-segments joining them.

Notice that using the intermediate fibers $v=\delta_i$ is compulsory if one wishes to get a graph $\mathscr{G}$ isotopic to $\mathscr{P}$. If not, whenever two branches have common starting points and endpoints, the embedding of the graph $\mathscr{G}$ obtained is not valid since two arcs are identified.

The algorithm is illustrated on Fig. 2. In addition

- If a singular point box have width δ, then the distance between the singular point and the vertex representing it is less than δ.

- One can compute the sign of the function $Sign_{ridge}$ defined in 3.1 for each regular point of each intermediate fiber. This defines the color of the ridge branch it belongs to. Then one can assign to each edge of the graph the color of its end point which is on an intermediate fiber.

7 Certified plot

In this section, we develop an algorithm to provide information of $\mathscr{P}$ when some of the calculations required to certify the topology do not succeed. The information we provide consists of intersections between rational fibers in the study box and the curve. These intersections are used to define the so-called *certified plot.* Notice that if some calculations succeed –in particular those of umbilics and purple points, then, the isolation boxes of these singularities can be superimposed to the certified plot.

For simplicity, we suppose that P is irreducible, that $P(u,v)=0$ has no isolated points, which is true under our genericity conditions of section 3.1, and that the studied domain is $[0,1]\times[0,1]$. In addition, we denote by n' an integer such that there does not exists a connected component of $P(u,v)=0$ which is embedded in a product of intervals of length less than $1/n'$.

Certified plot with n' known. To specify the plot, we make the following:

Definition. 3 *Given an integer n with $n>n'$, consider the $n\times n$ decomposition of the bounded domain $[0,1]\times[0,1]$ into* pixels*, i.e. products of intervals of length $1/n$: $\{[\frac{i}{n};\frac{i+1}{n}]\times[\frac{j}{n};\frac{j+1}{n}]\ i=0\ldots n-1,\ j=0\ldots n-1\}$. Also assume each pixel has a binary attribute: lit or not.*

This $n\times n$ decomposition is called the certified plot of $\mathscr{P}$ *at resolution n if a pixel is lit if and only if $\mathscr{P}$ intersects the pixel. (Note that the intersection may occur only on the border or even on a corner of the pixel).*

The method reduces to finding intersections between the curve $\mathscr{P}$ and horizontal and vertical fibers defining the pixels. The algorithm processing horizontal fibers is presented on Fig. 3. The same algorithm is used to process the vertical lines, i.e. computations are applied to polynomials $P(i/n,v)$.

For each $i = 0 \dots n$:

1. Compute the square free part of $P(u, i/n)$.
2. Isolate its roots with intervals of length less than $1/n$. (As recalled in section 4.2, the algorithm terminates with $L_e = \emptyset$ since the square free part has been taken.) Let $L_i = \{I_j \ ; \ j = 1 \dots m_i\}$ be the set of these intervals.
3. For each $j = 1 \dots m_i$
   (a) If I_j does not intersect a fiber in the other direction, there exists $k = 0 \dots n-1$ such that $I_j \subset [\frac{k}{n}; \frac{k+1}{n}]$. The pixels containing this segment are lit (if $0 < i < n$ these are $[\frac{k}{n}; \frac{k+1}{n}] \times [\frac{i-1}{n}; \frac{i}{n}]$ and $[\frac{k}{n}; \frac{k+1}{n}] \times [\frac{i}{n}; \frac{i+1}{n}]$, else only one of these is in the studied domain).
   (b) If I_j intersects the fiber $u = k/n$ ($k = 0 \dots n-1$) in the other direction then
   - if $P(k/n, i/n) \neq 0$ then its sign enable to refine I_j such that it does not intersect the fiber and one lights pixels as in step 3a —that is, we light the pixel to the right or the left of $u = k/n$.
   - if $P(k/n, i/n) = 0$ then one lights the pixels containing the point $(k/n, i/n)$ (if this point is not on the border of the domain, there are 4 such pixels).

Figure 3: Processing horizontal fibers $P(u, i/n)$

Certified plot with n' unknown. If n' is not known, one can choose an arbitrary n and apply the same strategy : only components which are embedded into products of intervals of length $< 1/n$ may not be represented.

8 Illustrations

We illustrate our algorithms for Bezier surfaces of degree four and five respectively. In both cases, the study domain is the domain $\mathscr{D} = [0,1] \times [0,1]$.

8.1 Certified topology

Given a parameterized surface, recall that certifying the topology of ridges requires going through three stages. First, the polynomial P is computed using Maple. Second, the zero-dimensional systems defining the study points are solved, together with the intersections between $\mathscr{P}$ and the fibers of study points and intermediate fibers. Third, the connections between all these points are performed so as to produce the embedded graph $\mathscr{G}$ isotopic to $\mathscr{P}$.

We illustrate this process on the Bezier surface whose control points are

$$\begin{pmatrix} [0,0,0] & [1/4,0,0] & [2/4,0,0] & [3/4,0,0] & [4/4,0,0] \\ [0,1/4,0] & [1/4,1/4,1] & [2/4,1/4,-1] & [3/4,1/4,-1] & [4/4,1/4,0] \\ [0,2/4,0] & [1/4,2/4,-1] & [2/4,2/4,1] & [3/4,2/4,1] & [4/4,2/4,0] \\ [0,3/4,0] & [1/4,3/4,1] & [2/4,3/4,-1] & [3/4,3/4,1] & [4/4,3/4,0] \\ [0,4/4,0] & [1/4,4/4,0] & [2/4,4/4,0] & [3/4,4/4,0] & [4/4,4/4,0] \end{pmatrix}$$

Alternatively, this surface can be expressed as the graph of the total degree 8 polynomial $h(u,v)$ for $(u,v) \in [0,1]^2$:

$$\begin{aligned} h(u,v) =& 116u^4v^4 - 200u^4v^3 + 108u^4v^2 - 24u^4v - 312u^3v^4 + 592u^3v^3 - 360u^3v^2 + 80u^3v + 252u^2v^4 - 504u^2v^3 \\ &+ 324u^2v^2 - 72u^2v - 56uv^4 + 112uv^3 - 72uv^2 + 16uv. \end{aligned}$$

The computation of the implicit curve has been performed using Maple 9.5 —see Maple worksheet accompanying [CFPR05], and requires less than one minute. It is a bivariate polynomial $P(u,v)$ of total degree 84, of degree 43 in u, degree 43 in v with 1907 terms and coefficients with up to 53 digits.

The study points S_u, S_p and S_c were computed using the softwares FGB and RS (`http://fgbrs.lip6.fr`). These systems being in shape position —cf appendix 10.3, the RUR can be computed as shown in [Rou99]. Alternatively, Gröbner basis can be computed first using [Fau99] or [Fau02]. We tested both methods and the computation time for the largest system S_c does not exceed 10 minutes. Table 1 gives the main characteristics of these systems.

Figure 4 displays the topological approximation graph of the ridge curve in the parametric domain $\mathscr{D}$ computed with the algorithm of section 6. The surface and its ridges are displayed on Fig. 5. To avoid occlusion problems of lifted ridge segments by the surface, we lifted on the surface the points lit by the algorithm from section 7 rather than the ridge segments of Fig. 4.

There are 19 critical points (black dots), 17 purple points (pink dots) and 8 umbilics, 3 of which are 3-ridge (green) and 5 are 1-ridge (yellow). Figure 6 displays on the left two close-ups of the bottom left 3-ridge umbilic, and on the right a more readable sketch. One can recognize an unsymmetric umbilic, that is a 3-ridge umbilic where the three blue branches are followed by the three red ones round the umbilic. The other 3-ridge umbilics are symmetric, that is branches alternate colors round the umbilic.

System	# of roots $\in \mathbb{C}$	# of roots $\in \mathbb{R}$	# of real roots $\in \mathscr{D}$
S_u	160	16	8
S_p	1068	31	17
S_c	1432	44	19

Table 1: characteristics of zero dimensional systems

On this example, the discriminant with respect to u has degree 3594 in v and coefficients with up to 3418 digits. CAD based strategies require solving polynomials of degree 43 with coefficients in a

field extension defined by an irreducible polynomial of degree 1431 with coefficient of 1071 digits. Up to our knowledge, none of the best existing software or libraries can perform such a computation in a reasonable time.

8.2 Certified plot

We provide a certified plot of the ridges for the degree 5 Bezier surface defined by the height function

$$\begin{aligned} h'(u,v) = &-587\,u^4v^3 - 0.15\,u - 0.5\,v - 469.5\,u^4v^4 + 1835\,u^3v^3 + 353.3\,u^5v^4 - 213.55\,u^5v^5 - 1627\,u^3v^2 + 407.5\,u^3v \\ &- 1642\,u^2v^3 + 1222\,u^2v^2 - 304\,u^2v + 0.5\,u^2 + 1.8\,v^2 - 1.4\,u^3 - 2.5\,v^3 + 465\,uv^3 - 308\,uv^2 + 76.5\,uv \\ &+ 1.5\,u^4 + 808\,u^4v^2 - 205.5\,u^4v - 401.5\,u^3v^4 + 826\,u^2v^4 + 1.1\,v^4 - 309.75\,uv^4 - 68.4\,u^5v^3 - 0.51\,u^5 \\ &+ 0.09\,v^5 - 96.9\,u^5v^2 + 26.1\,u^5v + 452.75\,u^4v^5 - 213.3\,u^3v^5 - 101.9\,u^2v^5 + 76.2\,uv^5 + 0.05\,. \end{aligned}$$

The ridge curve has total degree 110 and 3245 terms, it requires about 15 minutes to be computed with Maple. The system for umbilics has been computed and there are 9 solutions in the domain $\mathscr{D}$. The informations to build a topological approximation cannot be obtained in reasonable time, hence we only provide a certified plot. Figure 7 displays this certified plot and umbilics on the domain $\mathscr{D}$, this 512×512 resolution plot is computed in less than one minute. Figure 8 displays the certified plot lifted on the surface.

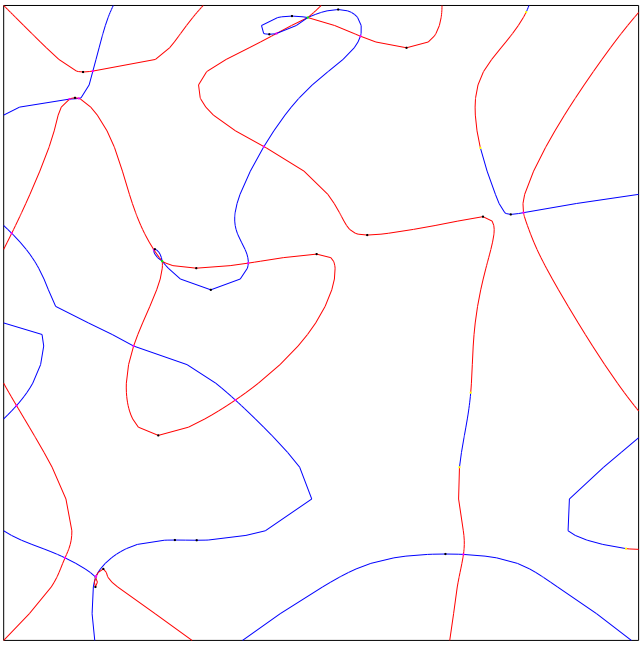

Figure 4: Isotopic approximation of the ridge curve with 3-ridge umbilics (green), 1-ridge umbilics (yellow), purple points (pink) and critical points (black).

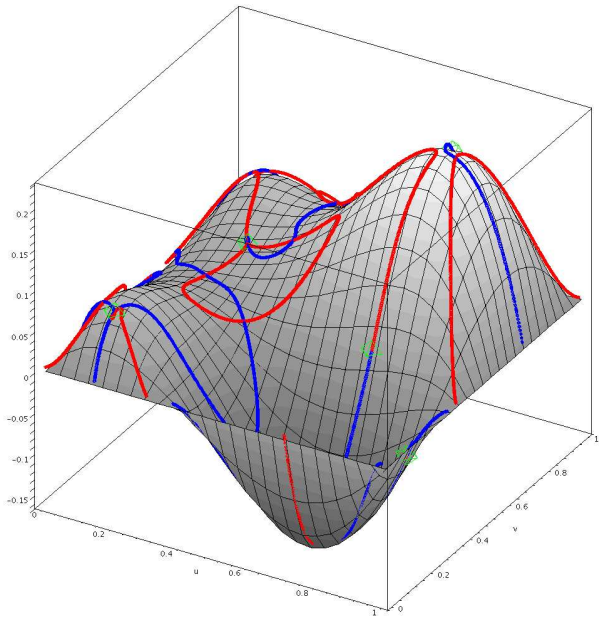

Figure 5: Plot of the degree 4 bivariate Bezier surface with ridges and umbilics

9 Conclusion

This paper develops two algorithms to investigate the ridges of parametric algebraic surfaces. The first one reports a topologically certified approximation of the ridges, and is the first one to achieve such a guarantee. For the practical cases where the resolution of the systems characterizing singular and critical points cannot be performed, the second one computes a certified plot at any fixed resolution. These algorithms are computationally demanding in terms of algebra. They are in a sense complementary to the heuristic ones developed in a companion paper, which are working directly on a triangulation of the surface, and provide a fast way to report non certified results.

The method developed for the computation of the topology of the ridges can be generalized for other algebraic curves. It gives an alternative to usual algorithms based on the CAD.

Acknowledgments. F. Cazals and M. Pouget acknowledge the support of the AIM@Shape and ACS European projects. Jean-Pierre Merlet is acknowledged for fruitful discussions.

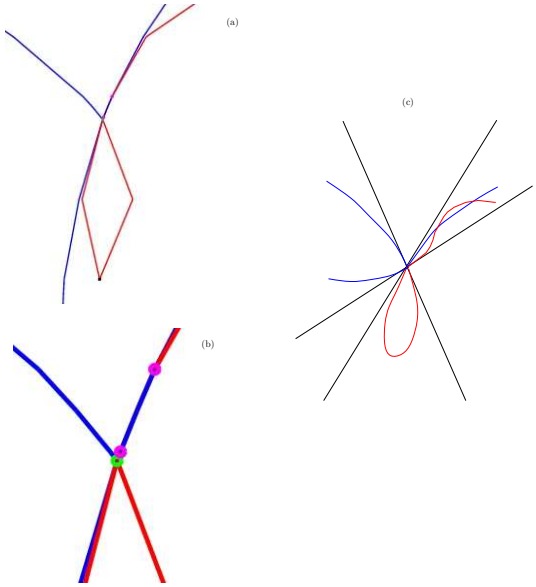

Figure 6: Close-up on the unsymmetrical umbilic of Fig. 4 —bottom left umbilic: (a,b)zooms of the isotopic approximation (c)corresponding arrangement of ridges

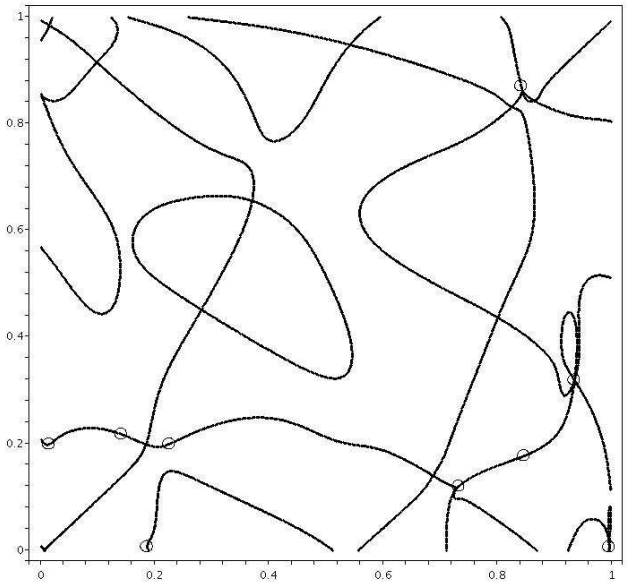

Figure 7: Certified plot of the degree 5 Bezier surface ridges with umbilics (circles) in the parametric domain.

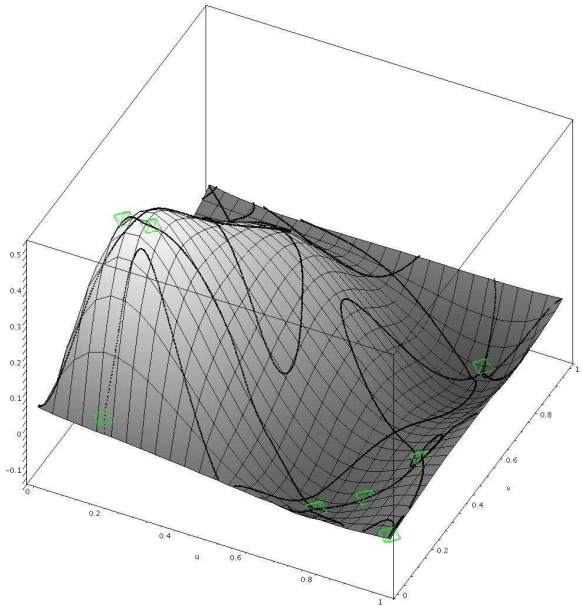

Figure 8: Certified plot of the degree 5 Bezier surface ridges with umbilics lifted on the surface.

References

[AS98] Auzinger and Stetter. An elimination algorithm for the computation of all zeros of a system of multivariate polynomial equations. *Int. Series of Numerical Math.*, 86:11–30, 1998.

[BCL82] B. Buchberger, G.-E. Collins, and R. Loos. *Computer Algebra Symbolic and Algebraic Computation*. Springer-Verlag, second edition edition, 1982.

[BPR03] S. Basu, R. Pollack, and M.-F. Roy. *Algorithms in real algebraic geometry*, volume 10 of *Algorithms and Computations in Mathematics*. Springer-Verlag, 2003.

[Buc85] B. Buchberger. *Gröbner bases : an algorithmic method in polynomial ideal theory*. Recent trends in multidimensional systems theory. Reider ed. Bose, 1985.

[CFPR05] F. Cazals, J.-C. Faugere, M. Pouget, and F. Rouillier. The implicit structure of ridges of a smooth parametric surface. Technical Report 5608, INRIA, 2005.

[CLO92] 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-Berlin-Paris, 1992.

[Col75] G.E. Collins. Quantifier elimination for real closed fields by cylindrical algebraic decomposition. *Springer Lecture Notes in Computer Science 33*, 33:515–532, 1975.

[CP05] F. Cazals and M. Pouget. Topology driven algorithms for ridge extraction on meshes. Technical Report RR-5526, INRIA, 2005.

[Fau99] J.-C. Faugère. A new efficient algorithm for computing gröbner bases (f_4). *Journal of Pure and Applied Algebra*, 139(1-3):61–88, June 1999.

[Fau02] Jean-Charles Faugère. A new efficient algorithm for computing gröbner bases without reduction to zero f_5. In *International Symposium on Symbolic and Algebraic Computation Symposium - ISSAC 2002, Villeneuve d'Ascq, France*, Jul 2002.

[FGLM93] J.C. Faugère, P. Gianni, D. Lazard, and T. Mora. Efficient computation of zero-dimensional gröbner basis by change of ordering. *Journal of Symbolic Computation*, 16(4):329–344, Oct. 1993.

[GLMT04] G. Gatellier, A. Labrouzy, B. Mourrain, and J.-P. Técourt. Computing the topology of 3-dimensional algebraic curves. In *Computational Methods for Algebraic Spline Surfaces*, pages 27–44. Springer-Verlag, 2004.

[GLS01] M. Giusti, G. Lecerf, and B. Salvy. A gröbner free alternative for solving polynomial systems. *Journal of Complexity*, 17(1):154–211, 2001.

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

[HCV52] D. Hilbert and S. Cohn-Vossen. *Geometry and the Imagination*. Chelsea, 1952.

[Koe90] J.J. Koenderink. *Solid Shape*. MIT, 1990.

[KOR05] J. Keyser, K Ouchi, and M. Rojas. The exact rational univariate representation for detecting degeneracies. In *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*. AMS Press, 2005. To appear.

[Laz83] D. Lazard. Gröbner bases, gaussian elimination, and resolution of systems of algebraic equations. In *EUROCAL' 83 European Computer Algebra Conference*, volume 162 of *LNCS*, pages 146–156. Springer, 1983.

[Mor90] R. Morris. *Symmetry of Curves and the Geometry of Surfaces: two Explorations with the aid of Computer Graphics*. Phd Thesis, 1990.

[Mor96] R. Morris. The sub-parabolic lines of a surface. In Glen Mullineux, editor, *Mathematics of Surfaces VI, IMA new series 58*, pages 79–102. Clarendon Press, Oxford, 1996.

[MT05] B. Mourrain and P. Trébuchet. Generalized normal forms and polynomial system solving. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, 2005. To appear.

[MWP96] T. Maekawa, F. Wolter, and N. Patrikalakis. Umbilics and lines of curvature for shape interrogation. *Computer Aided Geometric Design*, 13:133–161, 1996.

[Por01] I. Porteous. *Geometric Differentiation (2nd Edition)*. Cambridge University Press, 2001.

[Rou99] F. Rouillier. Solving zero-dimensional systems through the rational univariate representation. *Journal of Applicable Algebra in Engineering, Communication and Computing*, 9(5):433–461, 1999.

[RR05] N. Revol and F. Rouillier. Motivations for an arbitrary precision interval arithmetic and the mpfi library. *Reliable Computing*, 11:1–16, 2005.

[RZ03] F. Rouillier and P. Zimmermann. Efficient isolation of polynomial real roots. *Journal of Computational and Applied Mathematics*, 162(1):33–50, 2003.

10 Appendix: Algebraic pre-requisites

In this section, we summarize some basic results about Gröbner bases and their applications to solving zero-dimensional systems (systems with a finite number of complex roots). The reader may refer to [CLO92],[BPR03].

Lets denote by $\mathbb{Q}[X_1,\ldots,X_n]$ the ring of polynomials with rational coefficients and unknowns $X_1,\ldots,X_n$ and $S=\{P_1,\ldots,P_s\}$ any subset of $\mathbb{Q}[X_1,\ldots,X_n]$. A point $x\in\mathbb{C}^n$ is a zero of S if $P_i(x)=0 \quad \forall i=1\ldots s$. The ideal $I=\langle P_1,\ldots,P_s\rangle$ generated by $P_1,\ldots,P_s$ is the set of polynomials in $\mathbb{Q}[X_1,\ldots,X_n]$ constituted by all the combinations $\sum_{k=1}^{R} P_k U_k$ with $U_k\in\mathbb{Q}[X_1,\ldots,X_n]$. Since every element of I vanishes at each zero of S, we denote by $\mathrm{V}(S)=\mathrm{V}(I)=\{x\in\mathbb{C}^n \mid p(x)=0 \quad \forall p\in I\}$ (resp. $\mathrm{V}_{\mathbb{R}}(S)=\mathrm{V}_{\mathbb{R}}(I)=\mathrm{V}(I)\bigcap\mathbb{R}^n$) the set of complex (resp. real) zeroes of S.

10.1 Gröbner bases

A Gröbner basis of I is a computable generator set of I with good algorithmical properties (as described below) and defined with respect to a monomial ordering. In this paper, one will use the following orderings:

- *lexicographic order* : (Lex)

$$X_1^{\alpha_1}\cdot\ldots\cdot X_n^{\alpha_n} <_{\mathrm{Lex}} X_1^{\beta_1}\cdot\ldots\cdot X_n^{\beta_n} \Leftrightarrow \exists i_0\leq n \quad , \quad \left\{\begin{array}{l}\alpha_i=\beta_i, \quad \text{for } i=1,\ldots,i_0-1,\\ \alpha_{i_0}<\beta_{i_0}\end{array}\right. \tag{1}$$

- *degree reverse lexicographic order (DRL)* :

$$X_1^{\alpha_1}\cdot\ldots\cdot X_n^{\alpha_n} <_{\mathrm{DRL}} X_1^{\beta_1}\cdot\ldots\cdot X_n^{\beta_N} \Leftrightarrow X^{\left((\Sigma_k\alpha_k),-\alpha_n,\ldots,-\alpha_1\right)} <_{\mathrm{Lex}} X^{\left((\Sigma_k\beta_k),-\beta_n,\ldots,-\beta_1\right)} \tag{2}$$

Lets define the mathematical object “Gröbner”:

Definition. 4 *For any n-uple $\mu=(\mu_1,\ldots,\mu_n)\in\mathbb{N}^n$, let denote by X^μ the monomial $X_1^{\mu_1}\cdot\ldots\cdot X_n^{\mu_n}$. If $<$ is an admissible (compatible with the multiplication) monomial ordering and $P=\sum_{i=0}^{r} a_i X^{\mu(i)}$ any polynomial in $\mathbb{Q}[X_1,\ldots,X_n]$, we define:* $\mathrm{LM}(P,<)=\max_{i=0\ldots r \quad , \quad <} X^{\mu(i)}$ *(leading monomial of P w.r.t. $<$),* $\mathrm{LC}(P,<)=a_i$ *with* $\mathrm{LM}(P,<)=X^{\mu(i)}$ *(leading coefficient of P w.r.t. $<$) and* $\mathrm{LT}(P,<)=\mathrm{LC}(P,<)\mathrm{LM}(P,<)$ *(leading term of P wrt $<$).*

Definition. 5 *A set of polynomials G is a Gröbner basis of an ideal I wrt to a monomial ordering $<$ if for all $f\in I$ there exists $g\in G$ such that $LM(g,<)$ divides $LM(f,<)$.*

Given any admissible monomial ordering $<$ one can easily extend the classical Euclidean division to *reduce* a polynomial p by another one or, more generally, by a set of polynomials F, performing the reduction wrt to each polynomial of F until getting an expression which can not be reduced anymore. Lets denote such a function by $\mathtt{Reduce}(p,F,<)$ (reduction of the polynomial p wrt F). Unlike in the univariate case, the result of such a process is not canonical except if $F=G$ is a Gröbner basis:

Theorem. 2 *Let G be a Gröbner basis of an ideal $I \subset \mathbb{Q}[X_1,\ldots,X_n]$ for a fixed ordering $<$.*

(i) a polynomial $p \in \mathbb{Q}[X_1,\ldots,X_n]$ belongs to I if and only if Reduce$(p,G,<) = 0$,

(ii) Reduce(p,G,<) does not depend on the order of the polynomials in the list G, thus, this is a canonical reduced expression modulo I.

Gröbner bases are computable objects. The most popular method for computing them is Buchberger's algorithm ([BCL82, Buc85]). It has several variants and it is implemented in most of general computer algebra systems like Maple or Mathematica. The computation of Gröbner bases using Buchberger's original strategies has to face to two kind of problems :

- (A) arbitrary choices : the order in which are done the computations has a dramatic influence on the computation time; Precisely, one compute G by increasing the set of initial polynomials, adding so called S-polynomials ($\mathrm{spol}_<(p,q) = \frac{\mathrm{COF}_<(p,q)}{\mathrm{LT}(p)}p - \frac{\mathrm{COF}_<(p,q)}{\mathrm{LT}(q)}q, \mathrm{COF}_<(p,q) = \mathrm{LCM}(\mathrm{LT}_<(p),\mathrm{LT}_<(q))$ where LCM stands for "least common multiple") until all the possible S-polynomials of polynomials of G reduce to 0 modulo G ($\mathrm{Reduce}(\mathrm{spol}_<(p,q),G,<) = 0$).
- (B) useless computations : the original algorithm spends most of its time in computing 0 (at each step, most of the new S-polynomials do reduce to zero in a naive algorithm).

For problem (A), J.C. Faugère proposed ([Fau99] - algorithm F_4) a new generation of powerful algorithms ([Fau99]) based on the intensive use of linear algebra techniques. In short, the arbitrary choices are left to computational strategies related to classical linear algebra problems (matrix inversions, linear systems, etc.). For problem (B), J.C. Faugère proposed ([Fau02]) a new criterion for detecting useless computations

We pay a particular attention to Gröbner bases computed for elimination orderings since they provide a way of "simplifying" the system (an equivalent system with a structured shape). For example, a lexicographic Gröbner basis of a zero dimensional system (when the number of complex solutions is finite) has always the following shape (if we suppose that $X_1 < X_2 \ldots < X_n$):

$$\left\{\begin{array}{l} f(X_1) = 0 \\ f_2(X_1,X_2) = 0 \\ \vdots \\ f_{k_2}(X_1,X_2) = 0 \\ f_{k_2+1}(X_1,X_2,X_3) = 0 \\ \vdots \\ f_{k_{n-1}+1}(X_1,\ldots,X_n) = 0 \\ \vdots \\ f_{k_n}(X_1,\ldots,X_n) = 0 \end{array}\right. .$$

(when the system is not zero dimensional some of the polynomials may be identically null). A well known property is that the zeros of the smallest (w.r.t. $<$) non null polynomial define the Zariski

closure (classical closure in the case of complex coefficients) of the projection on the coordinate's space associated with the smallest variables.

More generally, an admissible ordering $<$ on the monomials depending on variables $[X_1,\ldots,X_n]$ is an ordering which eliminates $X_{d+1},\ldots,X_n$ if $X_i < X_j\ \forall i = 1\ldots d, j = d+1\ldots n$. The lexicographic ordering is a particular elimination ordering.

Definition. 6 *Given two monomial orderings $<_U$(w.r.t. the variables $U_1,\ldots,U_d$) and $<_X$(w.r.t. the variables $X_{d+1},\ldots,X_n$) one can define an ordering which "eliminates" $X_{d+1},\ldots,X_n$ by setting the so called block ordering $<_{U,X}$ as follows : given two monomials m and m', $m <_{U,X} m'$ if and only if $m_{|_{U_1=1,\ldots,U_d=1}} <_X m'_{|_{U_1=1,\ldots,U_d=1}}$ or $(m_{|_{U_1=1,\ldots,U_d=1}} = m'_{|_{U_1=1,\ldots,U_d=1}}$ and $m_{|_{X_{d+1}=1,\ldots,X_n=1}} <_U m'_{|_{X_{d+1}=1,\ldots,X_n=1}})$.*

Two important applications of elimination theory are the "projections" and "localizations". In the following, given any subset $\mathscr{V}$ of $\mathbb{C}^d$ (d is an arbitrary positive integer), $\overline{\mathscr{V}}$ is its Zariski closure, say the smallest subset of $\mathbb{C}^d$ containing $\mathscr{V}$ which is the zero set of a system of polynomial equations.

Proposition. 1 *Let G be a Gröbner basis of an ideal $I \subset \mathbb{Q}[U,X]$ w.r.t. any ordering $<$ which eliminates X. Then $G\bigcap\mathbb{Q}[U]$ is a Gröbner basis of $I\bigcap\mathbb{Q}[U]$ w.r.t. to the ordering induced by $<$ on the variables U; Moreover, if $\Pi_U : \mathbb{C}^n \longrightarrow \mathbb{C}^d$ denotes the canonical projection on the coordinates U, $V(I\cap\mathbb{Q}[U]) = V(G\cap\mathbb{Q}[U]) = \overline{\Pi_U(V(I))}$.*

Proposition. 2 *Let $I \subset \mathbb{Q}[X]$, $f \in \mathbb{Q}[X]$ and T be a new indeterminate, then $\overline{V(I)\setminus V(f)} = V((I + \langle Tf-1\rangle)\bigcap\mathbb{Q}[X])$. If $G' \subset \mathbb{Q}[X,T]$ is a Gröbner basis of $I+\langle Tf-1\rangle$ with respect to $<_{X,T}$ then $G'\bigcap\mathbb{Q}[X]$ is a Gröbner basis of $I : f^\infty := (I+\langle Tf-1\rangle)\bigcap\mathbb{Q}[X]$ w.r.t. $<_X$. The variety $\overline{V(I)\setminus V(f)}$ and the ideal $I : f^\infty$ are usually called the localization of $V(I)$ and I by f.*

10.2 Zero-dimensional systems

Zero-dimensional systems are polynomial systems with a finite number of complex solutions. This specific case is fundamental for many engineering applications. The following theorem shows that we can detect easily that a system is zero dimensional or not by computing a Gröbner basis for any monomial ordering :

Theorem. 3 *Let $G = \{g_1,\ldots,g_l\}$ be a Gröbner basis for any ordering $<$ of any system $S = \{P_1,\ldots,P_s\} \in \mathbb{Q}[X_1,\ldots,X_n]^s$. The two following properties are equivalent :*

- *For all index $i = 1\ldots n$, there exists a polynomial $g_j \in G$ and a positive integer n_j such that $X_i^{n_j} = LM(g_j,<)$;*
- *The system $\{P_1 = 0,\ldots,P_s = 0\}$ has a finite number of solutions in $\mathbb{C}^n$.*

If S is zero-dimensional, then, according to theorem 3, only a finite number of monomials $m \in \mathbb{Q}[X_1,\ldots,X_n]$ are not reducible modulo G, meaning that Reduce(m,G,$<$)$= m$. Mathematically, a system is zero-dimensional if and only if $\mathbb{Q}[X_1,\ldots,X_n]/I$ is a $\mathbb{Q}$-vector space of finite dimension. This vector space can fully be characterized when knowing a Gröbner basis:

Theorem. 4 *Let $S = \{p_1, \ldots, p_s\}$ be a set of polynomials with $p_i \in \mathbb{Q}[X_1, \ldots, X_n]$; $\forall i = 1 \ldots s$, and suppose that G is a Gröbner basis of $\langle S \rangle$ with respect to any monomial ordering $<$. Then :*

- $\mathbb{Q}[X_1, \ldots, X_n]/I = \{\text{Reduce}(p, G, <) \quad , \; p \in \mathbb{Q}[X_1, \ldots, X_n]\}$ *is a vector space of finite dimension;*
- $\mathscr{B} = \{t = X_1^{e_1} \cdots X_n^{e_n} \quad , \quad (e_1, \ldots, e_n) \in \mathbb{N}^n | \quad \text{Reduce}(t, G, <) = t\} = \{w_1, \ldots, w_D\}$ *is a basis of $\mathbb{Q}[X_1, \ldots, X_n]/I$ as a $\mathbb{Q}$-vector space;*
- $D = \sharp\mathscr{B}$ *is exactly the number of complex zeroes of the system $\{P = 0, \quad \forall P \in S\}$ counted with multiplicities.*

Thus, when a polynomial system is known to be zero-dimensional, one can switch to linear algebra methods to get informations about its roots. Once a Gröbner basis is known, a basis of $\mathbb{Q}[X_1, \ldots, X_n]/I$ can easily be computed (Theorem 4) so that linear algebra methods can be applied for doing several computations.

For any polynomial $q \in \mathbb{Q}[X_1, \ldots, X_n]$ the decomposition $\overline{q}$ =Reduce(q,G,$<$)=$\sum_{i=1}^{D} a_i w_i$ is unique (theorem 2) and we denote by $\vec{q} = [a_1, \ldots, a_D]$ the representation of $\overline{q}$ in the basis $\mathscr{B}$. For example, the matrix w.r.t. $\mathscr{B}$ of the linear map m_q: $\begin{pmatrix} \mathbb{Q}[X_1, \ldots, X_n]/I & \longrightarrow & \mathbb{Q}[X_1, \ldots, X_n]/I \\ \vec{p} & \longmapsto & \overrightarrow{pq} \end{pmatrix}$ can explicitly be computed (its columns are the vectors $\overrightarrow{qw_i}$) and one can then apply the following well-known theorem:

Theorem. 5 *(Stickelberger) The eigenvalues of m_q are exactly the $q(\alpha)$ where $\alpha \in V_{\mathbb{C}}(S)$.*

According to Theorem 5, the i-th coordinate of all $\alpha \in V_{\mathbb{C}}(S)$ can be obtained from m_{X_i} eigenvalues but the issue of finding all the coordinates of all the $\alpha \in V_{\mathbb{C}}(S)$ from $m_{X_1}, \ldots, m_{X_n}$ eigenvalues is not explicit nor straightforward (see [AS98] for example) and difficult to certify.

10.3 The Rational Univariate Representation

The Rational Univariate Representation [Rou99] is, with the end-user point of view, the simplest way for representing symbolically the roots of a zero-dimensional system without loosing information (multiplicities or real roots) since one can get all the information on the roots of the system by solving univariate polynomials.

Given a zero-dimensional system $I =< p_1, \ldots, p_s >$ where the $p_i \in \mathbb{Q}[X_1, \ldots, X_n]$, a Rational Univariate Representation of $\mathrm{V}(I)$ has the following shape : $f_t(T) = 0, X_1 = \frac{g_{t,X_1}(T)}{g_{t,1}(T)}, \ldots, X_n = \frac{g_{t,X_n}(T)}{g_{t,1}(T)}$, where $f_t, g_{t,1}, g_{t,X_1}, \ldots, g_{t,X_n} \in \mathbb{Q}[T]$ (T is a new variable). It is uniquely defined w.r.t. a given polynomial t which separates $V(I)$ (injective on $V(I)$), the polynomial f_t being necessarily the characteristic polynomial of m_t (see above section) in $\mathbb{Q}[X_1, \ldots, X_n]/I$ [Rou99]. The RUR defines a bijection between the roots of I and those of f_t preserving the multiplicities and the real roots :

$$\begin{array}{ccc} \mathrm{V}(I)(\cap\mathbb{R}) & \approx & \mathrm{V}(f_t)(\cap\mathbb{R}) \\ \alpha=(\alpha_1,\ldots,\alpha_n) & \rightarrow & t(\alpha) \\ (\frac{g_{t,X_1}(t(\alpha))}{g_{t,1}(t(\alpha))},\ldots,\frac{g_{t,X_n}(t(\alpha))}{g_{t,1}(t(\alpha))}) & \leftarrow & t(\alpha) \end{array}$$

For computing a RUR one has to solve two problems :

- finding a separating element t;
- given any polynomial t, compute a RUR-Candidate $f_t, g_{t,1}, g_{t,X_1}, \ldots, g_{t,X_n}$ such that if t is a separating polynomial, then the RUR-Candidate is a RUR.

According to [Rou99], a RUR-Candidate can explicitly be computed when knowing a suitable representation of $\mathbb{Q}[X_1,\ldots,X_n]/I$:

- $f_t=\sum_{i=0}^{D} a_i T^i$ is the characteristic polynomial of m_t. Lets denote by $\overline{f_t}$ its square-free part.
- for any $v\in\mathbb{Q}[X_1,\ldots,X_n]$, $g_{t,v}=g_{t,v}(T)=\sum_{i=0}^{d-1} Tr\,\mathrm{ace}(m_{vt^i})H_{d-i-1}(T)$, $d=\deg(\overline{f_t})$ and $H_j(T)=\sum_{i=0}^{j} a_i T^{i-j}$

In [Rou99], a strategy is proposed for computing a RUR for any system (a RUR-Candidate and a separating element), but there are special cases where it can be computed differently. When X_1 is separating $\mathrm{V}(I)$ and when I is a radical ideal, the system is said to be in *shape position*. In such cases, the shape of the lexicographic Gröbner basis is always the following :

$$\left\{\begin{array}{l} f(X_1)=0 \\ X_2=f_2(X_1) \\ \vdots \\ X_n=f_n(X_1) \end{array}\right. . \qquad (3)$$

As shown in [Rou99], if the system is in shape position, $g_{X_1,1}=f'_{X_1}$ and we have $f_{X_1}=f$ and $f_i(X_1)=g_{X_1,X_i}(X_1)/g_{X_1,1}(X_1) \bmod f$. Thus the RUR associated with X_1 and the lexicographic Gröbner basis are equivalent up to the inversion of $g_{X_1,1}=f'_{X_1}$ modulo f_{X_1}. In the rest of the paper we call this object a RR-Form of the corresponding lexicographic Gröbner basis. The RUR is well known to be much smaller than the lexicographic Gröbner basis in general (this may be explained by the inversion of the denominator) and thus will be our priviligied object. Note that it is easy to check that a system is in shape position once knowing a RUR-Candidate (and so to check that X_1 separates $\mathrm{V}(I)$): it is necessary and sufficient that f_{X_1} is square-free.

We thus can multiply the strategies for computing a symbolic solution : one can compute the RR-Form Gröbner directly using [Fau99] or [Fau02] for example or by change of ordering like in [FGLM93] or a RUR using the algorithm from [Rou99]. Choosing the right strategy will be part of our experimental section.

10.4 From formal to numerical solutions

Computing a RUR reduces the resolution of a zero-dimensional system to solving one polynomial in one variable (f_t) and to evaluating n rational fractions ($\frac{g_{t,X_i}(T)}{g_{t,1}(T)}, i = 1 \dots n$) at its roots (note that if one simply want to compute the number of real roots of the system there is no need to consider the rational coordinates). The next task is thus to compute all the real roots of the system (and only the real roots), providing a numerical approximation with an arbitrary precision (set by the user) of the coordinates.

The isolation of the real roots of f_t can be done using the algorithm proposed in [RZ03] : the output will be a list l_{f_t} of intervals with rational bounds such that for each real root α of f_t, there exists a unique interval in l_{f_t} which contains α. The second step consists in refining each interval in order to ensure that it does not contain any real root of $g_{t,1}$. Since f_t and $g_{t,1}$ are co-prime this computation is easy and we then can ensure that the rational functions can be evaluated using interval arithmetics without any cancellation of the denominator. This last evaluation is performed using multi-precision arithmetics (MPFI package - [RR05]). As we will see in the experiments, the precision needed for the computations is poor and, moreover, the rational functions defined by the RUR are stable under numerical evaluation, even if their coefficients are huge (rational numbers), and thus this part of the computation is still efficient. For increasing the precision of the result, it is only necessary to decrease the length of the intervals in l_{f_t} which can easily be done by bisection or using a certified Newton's algorithm. Note that it is quite simple to certify the sign of the coordinates : one simply have to compute some gcds and split, when necessary the RUR.

10.5 Signs of polynomials at the roots of a system

Computing the sign of given multivariate polynomials $\{q_1, \dots, q_l\}$ at the real roots of a zero-dimensional system may be important for many applications and this problem is not solved by the above method. Instead of "plugging" straightforwardly the formal coordinates provided by the RUR into the q_i, we better extend the RUR by computing rational functions which coincide with the q_i at the roots of I. This can theoretically simply be done by using the general formula from [Rou99] : $h_{t,j} = \sum_{i=0}^{D-1} \mathrm{Trace}(m_{q_j t^i}) H_{D-i-1}(T)$. One can directly compute the $\mathrm{Trace}(m_{q_j t^i})$ reusing the computations already done if the (classical) RUR (without additional constraints) has already been computed and show that as soon as l is small (at least smaller than the number of variables), it is not more costly to compute the extended RUR than the classical one.

Contents

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 scientifi que
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