Geometric Analysis of Algebraic Surfaces Based on Planar Arrangements

Eric Berberich* Michael Kerber* Michael Sagraloff*

Abstract

We present a method to compute the exact topology of a real algebraic surface S, implicitly given by a polynomial $f \in \mathbb{Q}[x, y, z]$ of arbitrary degree N. Additionally, our analysis provides geometric information as it supports the computation of arbitrary precise samples of S including critical points. We use a projection approach, similar to Collins' cylindrical algebraic decomposition (cad). In comparison we reduce the number of output cells to $O(N^5)$ by constructing a special planar arrangement instead of a full cad in the projection plane. Furthermore, our approach applies numerical and combinatorial methods to minimize costly symbolic computations. The algorithm handles all sorts of degeneracies without transforming the surface into a generic position. We provide a complete C++-implementation of the algorithm that shows good performance for many well-known examples from algebraic geometry.

1 Introduction

Problem and results: The topological analysis of algebraic curves and surfaces has received a lot of attention in algebraic geometry, computer graphics and CAGD. Beside the theoretical interest of the problem, accurate topological and geometric information of algebraic objects is crucial for a good visualization and for a meaningful approximation by simpler objects, such as splines or polygons.

We present an algorithm that provides topological information about an arbitrary algebraic surface S, given by an implicit equation in $\mathbb{Q}[x, y, z]$ of degree N. We compute a cell decomposition, where each cell is a smooth subvariety of S of dimension 0, 1, or 2, and determine how these cells are connected. Our cell decomposition has the *boundary property*, i.e., the boundary of a cell is given by a union of other cells (compare the similar notion of a CW-complex from algebraic topology). The result is similar to a *cylindrical algebraic decomposition* of $\mathbb{R}^3$, but our decomposition represents the topology using only $O(N^5)$ cells whereas the worst case complexity of a cad is $\Omega(N^7)$.

Our algorithm consists of three steps: First, we *project* the z-critical points of S to compute an arrangement $\mathcal{A}_S$, see Section 2. Second, we *lift* the components of $\mathcal{A}_S$ to $\mathbb{R}^3$, obtaining the cell decomposition Ω_S. It suffices to lift over one sample point of each component. Details are in Section 3. Third, we compute the *adjacencies* between the cells of Ω_S, as explained in Section 4.

We describe new methods for all three steps with the goal to replace costly symbolic computations by certified approximation methods as much as possible. Our toolbox for approximate methods contains, for instance, a numerical method for univariate root isolation (Bitstream Descartes [8]), an extension for the non-square-free case (m-k-Bitstream Descartes [7]), and interval arithmetic. Still, we guarantee to reflect the mathematical correct topology of the surface in all cases, as expected from the *exact geometric computation* (EGC) paradigm.

Our approach does not make any assumptions about the input surface and does never transform the coordinate system to prevent degeneracies. This allows to accurately sample the surface in arbitrary resolution by lifting points of a fine granulation of the xy-plane. On the other hand, we have to deal with degenerate situations, in particular with vertical lines that are part of the surface. Such lines are decomposed into vertical segments, and vertices in-between, to satisfy the boundary property.

We also provide an exact and complete implementation of the presented algorithm in C++. To our knowledge, this is the first EGC-implementation for the topological analysis of algebraic surfaces, including singular ones. It relies on an EGC-algorithm to produce arrangements of arbitrary algebraic plane curves, which has been presented recently in [6]. Our experiments show good performance for reference surfaces from algebraic geometry. Essentially needed in the projection step of our approach is the analysis of planar curves of degree up to $N(N-1)$ which limits its practical applicability for high-degree surfaces.

Related work: The problem of topology computation for algebraic plane curves has been extensively studied (see [7], [5] and the references therein). Recently, also exact methods for the case of space curves and surfaces came under consideration. Mourrain and Técourt [10] compute the topology of a surface by an isotopic piecewise linear mesh, using a plane-sweep approach. Cheng et al. [4] use a projection approach to produce a curvilinear wireframe that represents the surface topology. Both methods require a generic position of the surface and apply a linear change of coor-

*Max-Planck-Institut für Informatik, 66123 Saarbrücken, Germany, email: {eric,mkerber,msagralo}@mpi-inf.mpg.de

dinates otherwise. None of them reports on practical performance of their techniques.

Arnon et al. [1] compute cads in $\mathbb{R}^n$. In [2], they also compute the adjacencies between cells in a three-dimensional cad. Similar to us, they do not switch to generic position, and partition vertical lines into several cells to satisfy the boundary property. Our algorithm uses a more suitable cell decomposition for topology information, and applies approximate methods in the adjacency computation.

A more detailed version of this work appears in [3].

2 (n,k)-Arrangements

Throughout the article, the surface S is implicitly given by the polynomial $f \in \mathbb{Q}[x, y, z]$ of degree N. We require f to be square-free and primitive, i.e., S contains no component twice, and has no two-dimensional vertical component. For simplicity, we first assume that S contains no vertical line. The end of Section 4 sketches how to handle vertical lines.

For a fixed (algebraic) point $p = (p_x, p_y) \in \mathbb{R}^2$ we consider the *local polynomial* $f_p := f(p_x, p_y, z) \in \mathbb{R}[z]$.

Definition 1 *The* local degree n_p *is the degree of* f_p *in* z *and* $k_p := \deg \gcd(f_p, f_p')$ *the* local gcd degree.

We partition the (x, y)-plane into connected regions where the local degree and the local gcd degree remain invariant. To represent this partition, we use a planar arrangement. Thus we define:

Definition 2 *A connected set* $C \subset \mathbb{R}^2$ *is called (n,k)-invariant with respect to* S *if the local degree* $n = n_C$ *and the local gcd degree* $k = k_C$ *of* f *are invariant for all* $p \in C$. *An (n,k)-arrangement* $\mathcal{A}_S$ *for* S *is a planar arrangement whose vertices, edges, and faces are (n,k)-invariant with respect to* S.

Theorem 1 *There exists an (n,k)-arrangement* $\mathcal{A}_S$.

Proof. We give a constructive proof. Let p be an arbitrary point in the plane, and $f = \sum_{i=0}^{N} a_i(x, y)z^i$. Then n_p depends on the coefficients $a_N, \ldots, a_0$ by

$$n_p = \deg f_p = \max_{i=0,\ldots,N} \{i \mid a_i(p) \neq 0\}.$$

The same way, the local degree depends on the *principal Sturm-Habicht coefficients* $\mathrm{stha}_i(f_{n_p})$ (compare [9]) by

$$k_p = \deg \gcd(f_p, f_p') = \min_{i=0,\ldots,N} \{i \mid \mathrm{stha}_i(f_{n_p})(p) \neq 0\}.$$

The coefficients a_i's and $\mathrm{stha}_i(f_{n_p})$ define plane curves $\alpha_i = V(a_i)$ and $\sigma_{n_p,i} = V(\mathrm{stha}_i(f_{n_p}))$, respectively, of degree at most $N(N-1)$. Then n_p and k_p are determined by the curves p is part of. Thus, the arrangement $\mathcal{A}_S$ induced by α_i and, for all $n = 1, \ldots, N$, $\sigma_0(f_n), \ldots, \sigma_n(f_n)$, has only (n,k)-invariant cells. □

The constructed arrangement has (n,k)-invariant cells, but it contains far too many cells. To reduce the number of cells, consider the *silhouette* Γ_S of S, defined by $\mathrm{stha}_0(f) = \mathrm{res}_z(f, \frac{\partial f}{\partial z})$.

Lemma 2 *For any point,* $(n_p, k_p) = (N, 0)$ *if and only if* p *is not on* Γ_S. *As a consequence, all edges and vertices of an (n,k)-arrangement* $\mathcal{A}_S$ *away from* Γ_S *can be merged with their adjacent faces to an (n,k)-invariant face.*

Proof. We have that $\mathrm{res}_z(f, \frac{\partial f}{\partial z}) = a_N \mathrm{Disc}(f)$ where $\mathrm{Disc}(f)$ denotes the discriminant of f. Clearly, $n_p = N$ for a point p if and only if $a_N(p) \neq 0$. From the definition of the discriminant, $k_p = 0$ for a point p if and only if $\mathrm{Disc}(f)(p) \neq 0$. □

Consequently, any (n,k)-arrangement $\mathcal{A}_S$ can be turned into the minimal (n,k)-arrangement by a post-processing step (each feature $C \in \mathcal{A}_S$ stores (n_C, k_C) as data): Remove all edges and vertices away from Γ_S, and remove vertices on Γ_S that have exactly two adjacent edges, and both edges have the same local degree and local gcd degree as the vertex (and merge the adjacent edges). In our implementation, we integrated this post-processing step in the arrangement computation of the curves defined in the proof of Theorem 1, i.e., we add the curves into the arrangement one by one, and throw away unnecessary features immediately. This lowers the size of the intermediate values in the algorithm. One can prove that the size of the computed minimal $\mathcal{A}_S$ is $O(N^4)$, i.e., of same magnitude as the size of the silhouette arrangement.

3 The cell decomposition

We now fix an (n,k)-invariant cell C, and consider the surface lifted over C. We define the *local real degree* m_p to be the number of distinct real roots of the local polynomial f_p.

Theorem 3 *Each* $p \in C$ *has the same local real degree* m_C. *Moreover, for each* $i \in \{1, \ldots, m_C\}$, *the* i*-th lift* $C^{(i)}$ *over* C *is connected, where*

$$C^{(i)} := \{(p_x, p_y, z_i) \in \mathbb{R}^3 \mid (p_x, p_y) \in C \text{ and } z_i \text{ is the } i\text{-th root of } f_p\}.$$

Proof idea. Over an (n,k)-invariant set, the number of complex roots is constantly $n - k$. The roots of $f(p, z)$ continuously depend on p, thus in an open neighborhood of any point on C the imaginary roots stay imaginary. As the number of roots is preserved and imaginary roots only appear with their complex conjugate, the real roots also remain real. □

Theorem 3 shows that over C, the surface simply consists of m_C covertical copies of C. We can define:

Definition 3 *Let $\mathcal{A}_S$ be the minimal (n,k)-arrangement for S and m_C the local real degree of a cell $C \in \mathcal{A}_S$. The* cell decomposition *is defined as*

$$\Omega_S := \bigcup_{C \in \mathcal{A}_s} \left(\bigcup_{i=1,\ldots,m_C} \{C^{(i)}\} \right)$$

Theorem 4 *Ω_S consists of $O(N^5)$ cells.*

By computing the adjacencies between these cells as presented in Section 4, we thus can compute the topology of the surface using $O(N^5)$ many sample points which improves the $O(N^6)$ bound from [10]. A cylindrical algebraic decomposition consists of $\Omega(N^7)$ cells in the worst case, due to its vertical decomposition strategy in the plane.

The question remains how we compute the number m_C for each feature of $\mathcal{A}_S$. As we need also geometric information over C for the adjacency computation, we consider a more general problem: given $p \in C$, isolate the real roots of the local polynomial f_p. The number of isolating intervals for a sample point immediately reveals m_C.

For the isolation, we first consider the local gcd degree $k_p = k_C$ (Definition 1): if it is zero, the local polynomial is square-free, and we apply the *Bitstream Descartes method* [8], an exact root solver with adaptive precision, on f_p. Otherwise, we compute m_p, the number of real roots, using the Sturm-Habicht sequence of f_p [9], and apply the *m-k-Bitstream Descartes method* [7], an extension of the Bitstream Descartes for multiple roots, on f_p. If this steps fails (in this case, the m-k-Bitstream Descartes quits with a failure), the square-free part of f_p is computed, again using the Sturm-Habicht sequence, and the real roots of the square-free part are computed using the Bistream Descartes method.

4 Adjacency

The last step is to compute how the cells of Ω_S are connected. We first state without proof:

Theorem 5 *Ω_S has the boundary property, i.e., the boundary of each cell is the union of other cells.*

Equivalently, for any two cells M_1, M_2 with $\dim M_1 < \dim M_2$, either M_1 does not intersect the boundary of M_2, or it is completely contained in the boundary. In the latter case we call M_1 and M_2 *adjacent*. The adjacency relation of such a pair can be checked at an arbitrary point $p \in M_1$, i.e., the two cells are adjacent if and only if $p \in \overline{M_2}$.

Our strategy to compute the adjacencies is to consider all pairs of adjacent features C_1, C_2 of $\mathcal{A}_S$ in the (x, y)-plane, and to find the adjacencies between the lifts $C_1^{(i)}$ and $C_2^{(j)}$. Assume $\dim C_1 < \dim C_2$. There are two cases to consider:

C_1 has dimension 1: This means that $E := C_1$ is an edge, and $F := C_2$ is a face. As a filter, if E has at most one multiple real root, we adopt the combinatorial adjacency algorithm for plane curves from [7].

If the filter does not apply, the treatment is the same as in [2]. We choose sample points p for E and q for F with $q_x = p_x \in \mathbb{Q}$ (for vertical segments, we choose $p_y = q_y \in \mathbb{Q}$), and consider the planar curve $f|_{x=p_x} := f(p_x, y, z) \in \mathbb{Q}[y, z]$. The i-th lift $F^{(i)}$ of F is adjacent to the j-th lift $E^{(j)}$ of E if and only if there is an arc of the curve $V(f|_{x=p_x})$ connecting the i-th point over q_y with the j-th point over p_y. In our implementation, we use the algorithm presented in [7] to compute the adjacency information for $V(f|_{x=p_x})$.

C_1 has dimension 0: Then, C_1 is a vertex at point p, and C_2 is either an edge or a face. As above, we can filter the case that f_p has at most one multiple real root.

For the general method, let $z_1, \ldots, z_m$ denote the real roots of f_p. We choose (rational) intermediate values $q_0, \ldots, q_m$ such that $q_{i-1} < z_i < q_i$ for all $i = 1, \ldots, m$. The planes $z = q_i$ divide the real space in $m + 2$ *buckets* that separate the stack points z_i.

Definition 4 *Let $C \in \mathcal{A}_S$ (edge or face) be adjacent to p. A point p' on C is* bucket-faithful *if there exists a path from p' to p on C such that on that path, each lift $C^{(i)} \in \Omega_S$ over C remains in the same bucket.*

With a bucket-faithful point p' on C, the adjacencies of cells over C with cells over p follows by considering the real roots of $f_{p'}$: if the i-th root of $f_{p'}$ lies in the bucket of z_j, then the cells $C^{(i)}$ and $p^{(j)}$ are adjacent. Furthermore, points over p' that lie in either the bottom- or the top-most bucket belong to asymptotic components, i.e., they are unbounded in z-direction.

To compute a bucket-faithful point for C, we first compute a box B containing p such that no intersection with any plane $z = q_i$ takes place over B. In other words, each continuous path on S over B remains in the same bucket. We shrink B further until all features of $\mathcal{A}_S$ adjacent to p intersect the boundary of B. To find a bucket-faithful point of an edge adjacent to p, we start at p, and follow the edge until it crosses B for the first time. This intersection point is bucket faithful. For a bucket-faithful point of a face F, consider the edge $E \in \mathcal{A}_S$ that precedes F in counterclockwise order around p. Let q_E be the first intersection of E with the box boundary. Choose a point q_F on the box boundary between q_E and the next intersection of the box boundary with Γ_S in clockwise order. q_F is bucket-faithful for F.

Vertical lines In case where S contains a vertical line ℓ_p at a point $p \in \mathbb{R}^2$, the lift $p^{(i)}$, and thus the cell decomposition Ω_S as defined in Definition 3, is

Instance	$\deg_{x,y,z}$	(#V,#E,#F)	$\|\Omega_S\|$	**t** (in s)
steiner-roman	2,2,2	(5,12,8)	28	**0.73**
cayley-cubic	2,2,2	(3,10,8)	31	**0.74**
tangle-cube	4,4,4	(0,6,7)	28	**0.61**
bohemian-dome	4,4,4	(7,20,14)	61	**0.75**
chair	4,4,4	(4,9,7)	31	**3.05**
hunt	6,6,6	(3,2,3)	15	**1.21**
spiky	6,9,6	(1,8,8)	13	**1.43**
C8	8,8,8	(40,48,26)	496	**30.95**
random-3	3,3,3	(2,3,3)	15	**0.17**
random-4	4,4,4	(7,14,8)	64	**4.50**
random-5	5,5,5	(16,24,10)	154	**236.40**
interpolated-3	3,3,3	(4,6,3)	23	**0.34**
interpolated-4	4,4,4	(12,18,9)	82	**31.41**
projection-4d	4,4,4	(4,12,9)	34	**10.33**

no longer well-defined. At such points, ℓ_p is added to the cell decomposition. However, in order to fulfill the boundary property, ℓ_p is decomposed into vertical segments, and separating points in-between, according to the following theorem.

Theorem 6 *Let S contain the vertical line ℓ_p and $F \in \mathcal{A}_S$ be a face, which is adjacent to p. Then for any surface patch $F^{(i)}$ (the j-th lift of F) there exists an interval $I(F^{(i)}) \subset \mathbb{R}$, such that $p \times I(F^{(i)}) = \overline{F^{(i)}} \cap \ell_p$.*

The separating points are given by algebraic equations. The adjacency for cells over p is computed similarly to the case of non-vertical vertices, as described above. Because of space limitations, details are omitted here but are discussed in [3].

5 Implementation, conclusions, and outlook

We implemented the analysis in C++, taking from Exacus the surface representation and the analyses of algebraic curves [6, 7] and combined them with Cgal's `Arrangement_2` package to construct the (n,k)-arrangement. The possibility to attach data to Dcel-features allows to efficiently access (n,k)-relevant data for the lifting step. All computations follow the lazy-evaluation scheme, i.e., they are only triggered on demand and cached, e.g., the lifting. Following the generic programming paradigm, we decoupled combinatorial tasks from surface-specific ones.

We run experiments on well-known examples from algebraic geometry, interpolated instances, and also a generic projection of two quadrics in 4D; executed on a AMD Opteron(tm) 8218 (1 GHz) multiprocessor (1 MB cache) platform (32 GB RAM) running Debian Etch, compiled with `g++-4.1.2` using flags `-O2 -DNDEBUG` and the exact number types of Core. The table presents example surfaces along with their structural data and the obtained running times. About 90% of the time is spent to construct $\mathcal{A}_S$. Some surfaces do not show any (n,k)-vertex (e.g., `tangle-cube`) or -edge (e.g., xy-functional surfaces) at all. Due to our approximative and combinatorial methods, not more than the remaining 10% are spent to compute liftings and adjacencies.

Our work demonstrates that surface analysis is practically feasible for moderate degrees without switching to a generic position. The experiments show promising results thanks to our minimalistic cell decomposition and the consequent application of approximate methods. We are currently investigating how to enhance the cell decomposition to produce exact triangulations of arbitrary surfaces. An extension to multiple surfaces enables to analyze space curves and to realize boolean operations for surfaces.

References

[1] D. S. Arnon, G. E. Collins, and S. McCallum. Cylindrical algebraic decomposition I: The basic algorithm. *SIAM Journal on Computing*, 13:865–877, 1984.

[2] D. S. Arnon, G. E. Collins, and S. McCallum. An adjacency algorithm for cylindrical algebraic decompositions of three-dimensional space. *Journal of Symbolic Computation*, 5:163–187, 1988.

[3] E. Berberich, M. Kerber, M. Sagraloff. Exact Geometric-Topological Analysis of Algebraic Surfaces. In *Proc. of the 24th Ann. Symp. on Computational Geometry (SCG'08)*, 2008. To appear.

[4] J.-S. Cheng, X.-S. Gao, and M. Li. Determining the topology of real algebraic surfaces. In R. Martin, H. Bez, and M. Sabin, editors, *11. IMA Conference on the Mathematics of Surfaces*, volume 3604 of *LNCS*, pages 121–146, 2005.

[5] D. Diochnos, I. Z. Emiris, and E. P. Tsigaridas. On the complexity of real solving bivariate systems. In C. W. Brown, editor, *Proc. of the 2007 International Symp. on Symbolic and Algebraic Computation (ISSAC 2007)*, pages 127–134, 2007.

[6] A. Eigenwillig and M. Kerber. Exact and efficient 2d-arrangements of arbitrary algebraic curves. In *Proc. of the Nineteenth Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA08)*, pages 122–131, 2008.

[7] A. Eigenwillig, M. Kerber, and N. Wolpert. Fast and exact geometric analysis of real algebraic plane curves. In C. W. Brown, editor, *Proc. of the 2007 International Symp. on Symbolic and Algebraic Computation (ISSAC 2007)*, pages 151–158, 2007.

[8] A. Eigenwillig, L. Kettner, W. Krandick, K. Mehlhorn, S. Schmitt, and N. Wolpert. A Descartes algorithm for polynomials with bit-stream coefficients. In *8th International Workshop on Computer Algebra in Scientific Computing (CASC 2005)*, volume 3718 of *LNCS*, pages 138–149, 2005.

[9] L. Gonzalez-Vega, T. Recio, H. Lombardi, and M.-F. Roy. Sturm-Habicht sequences, determinants and real roots of univariate polynomials. In B. Caviness and J. Johnson, editors, *Quantifier Elimination and Cylindrical Algebraic Decomposition*, Texts and Monographs in Symbolic Computation, pages 300–316. Springer, 1998.

[10] B. Mourrain and J.-P. Técourt. Isotopic meshing of a real algebraic surface. Technical Report 5508, INRIA Sophia-Antipolis, 2005.