Exact Geometric-Topological Analysis of Algebraic Surfaces

Eric Berberich
Max-Planck-Institut für Informatik
66123 Saarbrücken, Germany
eric@mpi-inf.mpg.de

Michael Kerber
Max-Planck-Institut für Informatik
66123 Saarbrücken, Germany
mkerber@mpi-inf.mpg.de

Michael Sagraloff
Max-Planck-Institut für Informatik
66123 Saarbrücken, Germany
msagralo@mpi-inf.mpg.de

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 implementation of the algorithm, written in C++. It shows good performance for many well known examples from algebraic geometry.

Categories and Subject Descriptors:
I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling—*Curve, surface, solid and object representations*; G.4 [**Mathematical Software**]—*Algorithm design and analysis*; G.1.5 [**Numerical Analysis**]: Roots of Nonlinear Equations — *Polynomials, methods for*

General Terms: Algorithms, Performance

Keywords: Algebraic surfaces, exact geometric computation, topology computation, cylindrical algebraic decomposition

1. INTRODUCTION

Problem and results: The topological analysis of real algebraic curves and surfaces has received a lot of attention in algebraic geometry, computer graphics and computer aided geometric design. 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 [8], [40].

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 [34], [12]). The result is similar to a *cylindrical algebraic decomposition* (cad) [17], [15] 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 $\Omega(N^7)$. It is possible to refine the decomposition into simply connected cells without compromising the final complexity.

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

We describe new methods for all three steps with the goal to replace costly symbolic computations by certified approximation approaches as much as possible. Our toolbox for approximate methods contains, for instance, a numerical method for univariate root isolation (Bitstream Descartes [20], [23]), an extension for the non-square-free case (m-k-Bitstream Descartes [22]), 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 [44].

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 [21]. Our experiments show good performance for many 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 [22], [19], [29], [39], [31], [28]. Recently, also exact methods for the case of space curves [24], [2], [27] came under consideration.

For topology computation of algebraic surfaces, two principle approaches can be distinguished: one is to consider *level-curves* of the surface for certain critical values and to connect the components of these levels in order to obtain a topological description of the surface; see the recent works of Mourrain and Técourt [37] (also in [11]), Fortuna et al. [25], [26] (for non-singular curves) and Alcázar et al. [1] (where the connection step is missing). The other approach is to project the critical points of the surface to the plane, obtaining the *silhouette curve*. The topology is then deduced by lifting the features induced by the silhouette. We are following this approach; see also Cheng et al. [16] and the articles about cad below.

The tools to compute a surface's topology are similar in all mentioned approaches: each one needs to compute the topology of algebraic plane curves, either to analyze the level curves or the silhouette. Additionally, *critical points* of the surface, or at least their projections, must be identified, which is usually done by resultant-calculus or Groebner bases. Most algorithms, e.g., [37], [25], [26], [16], apply a linear (topology-preserving) transformation to obtain a generic position that simplifies the computation. As already said, we decided not to allow such a transformation in our algorithm to preserve also geometric properties of the surface.[1]

None of the articles [37], [25], [26], [1], [16] report on the practical performance of their algorithms; if implementations are mentioned at all,[2] they mainly propose to carry out the calculations symbolically, or leave the concrete implementation of certain substeps open. We tried to profit from numerical methods as much as possible and we experienced that this accelerates the algorithm significantly. We take this as the main reason of the overall good practical performance of our algorithm.

Cylindrical algebraic decomposition is a more general problem and constitutes its own research area [15]: Arnon et al. [4] presented an algorithm to compute a cad in $\mathbb{R}^n$. Their algorithm has been improved in several ways: Numerical methods have been used to speed-up the lifting step [41], [18], [14], improvements of the projection step reduce the number of considered polynomials in the cad [13], [35], cells in the cad are combined into clusters to reduce the complexity [3], and algorithms have been proposed to compute which cells are adjacent [5], [6], [36]. Some ideas of our algorithm already appeared in those articles; for other problems, we propose novel alternatives. We discuss the similarities and differences with the appropriate references when we discuss the algorithm in detail.

Outline: We describe the lifting step of the algorithm in Section 2 and introduce the planar arrangement and our cell decomposition in Section 3. Section 4 deals with the adjacency computation, and how to treat the special case of vertical line components. Section 5 reports on our implementation and experiments.

2. Z-FIBERS

In what follows, S always denotes a surface of degree N, and $f \in \mathbb{Q}[x,y,z]$ denotes its implicit equation. We henceforth assume that f is a square-free and primitive polynomial, i.e., S contains no irreducible component twice, and has no two-dimensional vertical component. The treatment of non-primitive polynomials consists of a separate analysis of the primitive part and the vertical part. We skip details for brevity.

[1]The idea from [22] to transform the curve back into its original coordinate system seems not to extend easily to the surface case.

[2]Complete implementations have been presented for subclasses of surfaces, such as intersections of quadrics [10] and meshes of non-singular surfaces [38].

DEFINITION 2.1. *The z-fiber of a point $p := (p_x, p_y) \in \mathbb{R}^2$ is*

$$Z_p := \{\gamma \in \mathbb{R} \mid f(p_x, p_y, \gamma) = 0\}.$$

Note that the fiber can be equal to $\mathbb{R}$, in case S contains the whole vertical line $\ell_p := p \times \mathbb{R}$. We aim for a method to compute the z-fiber for an arbitrary point p with algebraic coordinates in the plane, i.e., isolate the real roots of the polynomial $f_p := f(p_x, p_y, z) \in \mathbb{R}[z]$. Computational difficulties arise because f_p has algebraic coefficients for many z-fibers computed by our method, and because f_p might have multiple roots. We use some exact information about f_p to overcome such problems:

DEFINITION 2.2. *Let p and f be as above. The* local degree n_p *is the degree of f_p in z. The* local gcd degree k_p *is the degree of* $\gcd(f_p, f_p')$*. The* local real degree m_p *is the number of distinct real roots of f_p.*

Assuming that n_p, k_p and m_p are known, the z-fiber computation for p works as follows. If $k_p = 0$, then f_p is square-free; in that case, we apply the *Bitstream Descartes method* [20], [23] on f_p. The method computes the real roots of an exact polynomial only by numerically approximating the coefficients, i.e., in our case by evaluating f at x and y with iterated and coherent refinements of interval approximations for p_x and p_y. Otherwise, if $k_p > 0$, we try to use the *m-k-Bitstream Descartes method* [22, Sec. 5]; it exploits knowledge about the local real degree and the local gcd degree, and isolates the real roots using numerical approximations even if f_p has at most one multiple root. Unfavorable cases are detected by the method, it simply reports a failure in this case. If this happens, we compute the square-free part f_p^* of f_p and apply the Bitstream Descartes method on f_p^*.

Why did we choose the Bitstream Descartes method for the lifting step? First of all, the Descartes method is considered to be a practically efficient root isolation method, and using numerical approximations of the coefficients is experienced to speed up the computation further [41], [18], [14]. Thus, our choice for the Bitstream Descartes aims for practical efficiency, but it has another advantage: By a randomized choice of subdivision points, and by its adaptive precision management, the algorithm gives a success guarantee for the square-free case, regardless of the polynomial's root separation. Thus, a fall back to a symbolic root isolator is never necessary. The m-k-variant also gives a success guarantee except for the case that the polynomial is algebraically difficult, i.e., it has several multiple roots. Then, the polynomial has to be made square-free by symbolic computation, but the square-free part can again be tackled with the square-free version of the Bitstream Descartes method.

The remainder of this section deals with the computation of m_p, k_p and the square-free part f_p^*. They are computed using an algebraic tool called *Sturm-Habicht sequence* (cf. [30], the equivalent term of *signed subresultant sequence* appears in [9]):

DEFINITION 2.3. *Let $\mathbb{D}$ be any domain, $g \in \mathbb{D}[t]$ with* $\deg g = n$*, and $\delta_k := (-1)^{k(k+1)/2}$. For $k \in \{0,\dots,n\}$, the k-th Sturm-Habicht polynomial of g is defined as*

$$\begin{aligned} \mathrm{StHa}_n(g) &:= g \\ \mathrm{StHa}_{n-1}(g) &:= g' \\ \mathrm{StHa}_k(g) &:= \delta_{n-k-1}\mathrm{Sres}_k(g,g'), \quad k = 0,\dots,n-2 \end{aligned}$$

where $\mathrm{Sres}_k(g,g')$ *is the k-th subresultant of g and g'. We define* $\mathrm{stha}_k(g)$*, the k-th principal Sturm-Habicht coefficient of g, as the coefficient of t^k in* $\mathrm{StHa}_k(g)$.

The principal Sturm-Habicht coefficients can be represented as determinants of the Sylvester submatrices, possibly multiplied by -1. The signs of the principal Sturm-Habicht coefficients determine $m = \#\{z \in \mathbb{R} \mid g(z) = 0\}$. The degree of $gcd(g, g')$ is given as the minimal index k for which $\mathrm{stha}_k(g) \neq 0$ (for more details, see [29], [22]). Thus, the Sturm-Habicht sequence for f_p reveals the numbers m_p and k_p.

For the square-free part, we consider the cofactors of the Sturm-Habicht polynomials [9, Prop. 8.38].

PROPOSITION 2.4. *For $j < n$, there exist polynomials u_j, v_j with $\deg(u_j) \leq n - j - 2$, $\deg(v_j) \leq n - j - 1$ such that $\mathrm{StHa}_j(g) = u_j g + v_j g'$.*

All cofactors u_j and v_j can be written as determinants of "Sylvester-like" matrices. The square-free part g^* of g is given by one of the v_j's [9, Prop. 10.14, Cor. 10.15]:

LEMMA 2.5. *Let $k = \deg \gcd(g, g') > 0$. Then, $g^* = v_{k-1}$.*

For the computation of the Sturm-Habicht sequence for f_p, we exploit that they are well-behaved under specializing parameters. We restrict to the three-dimensional case here.

PROPOSITION 2.6 (SPECIALIZATION PROPERTY). *Let $f \in \mathbb{R}[x,y,z]$, $p := (p_x, p_y) \in \mathbb{R}^2$. If $\deg_z f = \deg f_p$, then for all $j = 0, \ldots, n$ it holds that $\mathrm{StHa}_j(f_p) = \mathrm{StHa}_j(f)\mid_{x=p_x, y=p_y}$*

In other words, the Sturm-Habicht sequence for f (with z as outer variable) reveals the specialized Sturm-Habicht sequence for all f_p with $\deg_z F = \deg f_p$. Such points p are also called *regular*. We generalize this idea to obtain a Sturm-Habicht sequence also for non-regular points through specialization.

DEFINITION 2.7. *For $f = \sum_{i=0}^{N} a_i(x,y) z^i$, we define* the reductum *$f_n := \sum_{i=0}^{n} a_i(x,y) z^i$ for $n = 0, \ldots, N$.*

LEMMA 2.8. *For all $j = 0, \ldots, n_p$, it holds that $\mathrm{StHa}_j(f_p) = \mathrm{StHa}_j(f_{n_p})\mid_{x=p_x, y=p_y}$.*

In our implementation, we use the algorithm presented in [9, Alg. 8.22] to compute Sturm-Habicht sequences with cofactors. They are computed using a polynomial remainder sequence [33] which is more efficient than computing the Sturm-Habicht sequence via determinantal expressions.

3. (N,K)-ARRANGEMENTS AND THE CELL DECOMPOSITION

The z-fiber computation for p is based on the computation of the integers n_p, k_p and m_p (Definition 2.2). In this section, we compute an arrangement in the (x,y)-plane such that all points of an arrangement feature have invariant n_p and k_p. As we will see, also m_p is invariant for such a feature. This allows to efficiently compute the z-fiber over any point in the plane, since all algebraic information is determined by the arrangement feature the point belongs to. Also, we show that the lift of such a feature is the union of disjoint function graphs which form the basis for our cell decomposition of the surface.

DEFINITION 3.1. *We call a connected set $C \subset \mathbb{R}^2$* (n,k)-invariant *with respect to a surface $S = V(f)$ if the local degree n_C and the local gcd degree k_C of f are invariant for all $p \in C$. An (n,k)-arrangement for S is a planar arrangement whose vertices, edges, and faces are (n,k)-invariant with respect to S.*

In his seminal paper about cylindrical algebraic decomposition, Collins [17] has proved that an (n,k)-invariant set is *delineable*, i.e., that the (real) lift over the set is the union of m disjoint function graphs. We state a slightly weaker version of his theorem:

THEOREM 3.2. *Let C be an (n,k)-invariant set. Then, each $p \in C$ has the same local real degree m_C. Moreover, for each $i = 0, \ldots, m_C$, the i-th lift $C^{(i)}$ over C (defined below) is connected.*

$$C^{(i)} := \{(p_x, p_y, z_i) \in C \times \mathbb{R} \mid z_i \text{ is the i-th root of } p\text{'s z-fiber}\}.$$

PROOF. Over an (n,k)-invariant set, the number of distinct 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 total number of roots is preserved and imaginary roots only appear together with its complex conjugate, the real roots also remain real. See [17, Thm. 1] for more details. □

The next construction also appears in Collins' work [17, Thm. 4]:

THEOREM 3.3. *For each algebraic surface S, there exists an (n,k)-arrangement.*

PROOF. We give a constructive proof. Let p be an arbitrary point in the plane, and $f = \sum_{i=0}^{N} a_i(x,y) z^i$. The local degree of f at p simply 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 gcd degree depends on the principal Sturm-Habicht coefficients $\mathrm{stha}_i(f_{n_p})$ by

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

The coefficients a_i's and $\mathrm{stha}_i(f_n)$ define plane curves $\alpha_i = V(a_i)$ and $\sigma_{n,i} = V(\mathrm{stha}_i(f_n))$, 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 induced by $\alpha_N, \ldots, \alpha_0$ and, for all $n = 1, \ldots, N$, $\sigma_{n,0}, \ldots, \sigma_{n,n}$ has only (n,k)-invariant cells. □

The proof presents a way to compute an (n,k)-arrangement for a surface. However, the resulting arrangement consists of much more features than actually necessary – we aim for an (n,k)-arrangement with large components.

DEFINITION 3.4. *The* silhouette *Γ_S of S is defined by* $\mathrm{stha}_0(f) = \mathrm{res}_z(f, \frac{\partial f}{\partial z})$

LEMMA 3.5. *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 away from Γ_S can be merged with their adjacent faces to a (n,k)-invariant face.*

PROOF. Using [9, Prop. 4.27], we have $\mathrm{res}_z(f, \frac{\partial f}{\partial z}) = a_N \mathrm{Disc}(f)$ 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 regular point p if and only if $\mathrm{Disc}(f)(p) \neq 0$. □

Consequently, having any (n,k)-arrangement, we can turn it into a minimal (n,k)-arrangement by a post-processing step (we assume that each arrangement feature C stores the numbers n_C and 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). A similar idea of merging of a cad into clusters has been proposed by Arnon [3].

We next present an algorithm that integrates this post-processing step in the arrangement computation, to lower the size of the intermediate arrangements in the algorithm. The main tool is the computation of *overlays*. Given arrangements A_1 and A_2, the overlay is the union A_3 of both arrangements; also, each feature of A_3 knows which feature of A_1 and A_2 it comes from.

We start by computing the arrangement A defined by the silhouette Γ_S only. Each face gets the values $(N,0)$ according to Lemma 3.5. We first decompose A such that each feature has invariant local degree. To do so repeat the following steps for $n = N,\ldots,0$: Overlay A with the arrangement of the curve α_n, the result is A'. Remove all features of A' that lie on a face of A. Also, remove all vertices of A' that lie on an edge of A whose local degree has already been set. For each feature that lies on a face of α_n, and whose degree is not set yet, set its local degree to n. Set $A \leftarrow A'$ and proceed with the next iteration. At the end, set the local degree of all features which are not yet set to $-\infty$, as above these features S is vertical.

Next, we further decompose A into (n,k)-invariant cells. For that, we iterate over the degrees and overlay with the corresponding principal Sturm-Habicht coefficient curves $\sigma_{n,i}$.

Repeat for $n = N,\ldots,1$: Repeat for $k = 0,\ldots,n-1$: Overlay A with the arrangement of $\sigma_{n,k}$, the result is A'. Remove all features of A' that lie on a face of A. Remove all vertices of A' that lie on an edge of A whose local gcd degree has already been set, or whose local degree does not equal n. For each feature of A that lies on a face of $\sigma_{n,k}$, whose local degree is n, and whose local gcd degree is not yet set, set the local gcd degree to k. Set $A \leftarrow A'$ and proceed with the next iteration.

We remark the obvious optimization that for the local gcd degree, one has only to consider those degrees n that appear as the local degree of at least one feature. Also, one can stop the inner iteration over the k's as soon as all features of degree n know their local gcd degree.

The (n,k)-arrangement computed by the above algorithm is called $\mathscr{A}_S$ from now on. It basically consists of the overlay of the leading coefficient curve and the discriminant curve of f (compare Lemma 3.5), but introduces some subdivision points on edges to ensure (n,k)-invariance. This approach is in a similar spirit as the improved projection operators in cad computation; see the work of McCallum [35] and the slight improvement by Brown [13]. They show that considering the leading coefficient and the discriminant is sufficient to ensure delineability, if all cells are made *order-invariant* (compare the definition in [35]). So, the consideration of the non-leading coefficients and the principal Sturm-Habicht coefficients is not necessary to ensure the statement of Theorem 3.2. We point out that our arrangement $\mathscr{A}_S$ holds information about the local degree and local gcd degree of each feature by construction. As we have exposed in Section 2, this allows to apply fast methods in the lifting step. Also, to compute an order-invariant cad, one has to deal with the partial derivatives of the discriminant instead, which are usually of higher degree than the principal Sturm-Habicht polynomials used in our approach.

The complexity of our (n,k)-arrangement $\mathscr{A}_S$ is not greater than that for Γ_S.

THEOREM 3.6. *The number of cells of $\mathscr{A}_S$ is $O(N^4)$.*

PROOF. Since arrangements induce planar graphs, it is enough to count vertices. The silhouette Γ_S is of degree $O(N^2)$, so it has, by Bézout's theorem $O(N^4)$ critical points. We have to show that the segmentation with respect to the remaining curves in the algorithm does not introduce more than $O(N^4)$ new vertices.

Consider the decomposition of Γ_S into irreducible components $\Gamma_{S,i}$ with degree ν_i, and fix one $\gamma = \Gamma_{S,i}$ of degree ν. During the algorithm, new vertices for γ (that are not removed in the same iteration) are only introduced in two iteration steps:

First, when a coefficient curve α_n does not contain the whole curve γ. This introduces at most $\nu \cdot N$ many vertices. All further coefficient curves $\alpha_{n-1},\ldots,\alpha_0$ do not introduce new vertices on γ, since the local degree of all edges for γ is set to n.

Second, new vertices are introduced when a Sturm-Habicht polynomial $\mathrm{stha}_k(f_n)$ does not contain the whole curve γ. This introduces at most $\nu \cdot N^2$ many new vertices. All further Sturm-Habicht curves $\mathrm{stha}_{k-1}(f_n),\ldots,\mathrm{stha}_0(f_n)$ do not introduce new vertices on γ, since the local gcd degree of all edges for γ is set to k.

After all, each $\Gamma_{S,i}$ gets at most $O(\nu_i \cdot N^2)$ many new vertices, and the ν_i sum up to N^2. $\square$

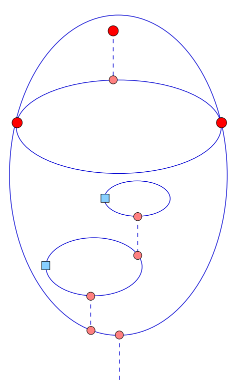
DEFINITION 3.7. *Let S be a surface, without vertical component, $\mathscr{A}_S$ as above and m_C the local real degree of a cell $C \in \mathscr{A}_S$. The* cell decomposition *Ω_S is defined as*

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

COROLLARY 3.8. *For a surface of degree N without vertical line, the number of cells in Ω_S is $O(N^5)$.*

This means that we achieve a topological description of the surface using $O(N^5)$ many sample points. This is less compared to cad which consists of $\Omega(N^7)$ cells in the worst case, due to its vertical decomposition strategy in the plane. An alternative decomposition into $O(N^6)$ cells has been proposed by Mourrain and Técourt [37]. In Section 4.3, we extend Ω_S to surfaces with vertical lines, and show that the extension still keeps the same worst-case complexity of $O(N^5)$.

Extracting simply connected cells Sometimes it might be advantageous to achieve a decomposition into simply connected cells (i.e., each path in a cell is contractible to a point). Our decomposition Ω_S does not have this property. We next propose an algorithm that transforms Ω_S into a decomposition of simply connected cells.



The first step is to compute a simply connected refinement $\mathscr{A}'_S$ of $\mathscr{A}_S$. Only one- and two-dimensional cells of $\mathscr{A}_S$ can be non-simply connected. Consider the planar graph G induced by $\mathscr{A}_S$, by mapping its 0-dimensional cells to nodes, and its 1-dimensional connected cells to edges. Simply connectivity for 1-dimensional cells is achieved by adding an additional vertex for each cyclic edge; see the squared vertices in the picture to the left.

To prevent non-simply connected faces, we apply the following algorithm: while G contains a bounded connected component, choose such a component, and connect its y-minimal point downwards using a vertical arc (dashed) until it reaches another component of G (or if this does not happen, the arc goes to $-\infty$). Observe that each such arc either merges two connected components, or turns one of them unbounded. Thus, it is clear that the algorithm terminates, and produces a graph without bounded connected components. The computed graph induces a refined arrangement $\mathscr{A}'_S$ of $\mathscr{A}_S$. The newly added cells inherit the (n,k)-properties of the cell they are included. For the such refined $\mathscr{A}'_S$, we claim:

PROPOSITION 3.9. *Each cell of $\mathscr{A}_S'$ is simply connected, and its number of cells is $O(N^4)$.*

PROOF. Assume for a contradiction that there is a cell C of $\mathscr{A}_S'$ which is not simply connected. Clearly, C cannot be 1-dimensional as we split all cycles. So assume that C is a face. Since it is not simply connected, there is a cycle P that is not contractible. Hence, its interior contains a connected component, which must be bounded. That contradicts the fact that there is no bounded connected component .

For the complexity statement, observe that we introduce at most one edge and two vertices, and split at most one face for each connected component. Since the number of connected components is not greater than the number of faces, we add at most 4 cells for each face of $\mathscr{A}_S$. This proves that we do not increase the complexity. $\square$

The arrangement $\mathscr{A}_S'$ implies a cell decomposition Ω_S' by lifting the components (compare Definition 3.7).

PROPOSITION 3.10. *Each cell of Ω_S' is simply connected, and its number of cells is $O(N^5)$.*

PROOF. Each cell C of Ω_S' is the diffeomorphic image of a (simply connected) cell of $\mathscr{A}_S'$, it follows that C is simply connected as well. The complexity statement is clear, as each cell can have up to N lifts. $\square$

We mention that this refinement into simply connected cells has not yet been integrated into our implementation that we present in Section 5.

4. ADJACENCY

As already mentioned we aim for a decomposition of S which also fulfills the *boundary property*, i.e., the boundary of each cell should be the union of other cells. Equivalently, for any two cells M_1, M_2 with $\dim M_1 < \dim M_2$, we must have $M_1 \cap \overline{M_2} = \varnothing$ or $M_1 \subset \overline{M_2}$. In the latter case we call M_1 and M_2 *adjacent*. Then 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 $p \in \overline{M_2}$. Theorem 4.1 shows that in case of a surface S which contains no vertical line ℓ_p, the decomposition Ω_S defined in Definition 3.7 already has this boundary property.

THEOREM 4.1. *Let $M_1, M_2 \in \Omega_S$ with $\dim M_1 < \dim M_2$ and $C_1, C_2 \in \mathscr{A}_S$ their corresponding adjacent projections onto the plane. If C_1 has local degree $n_{C_1} \neq -\infty$ and $M_1 \cap \overline{M_2} \neq \varnothing$, then $M_1 = \overline{M_2} \cap (C_1 \times \mathbb{R})$.*

PROOF. Let M_2 be the j_0-th lift of C_2 and $p = (p^*, z_0) \in \overline{M_2} \cap (C_1 \times \mathbb{R})$ an arbitrary point, contained in a lift $C_1^{(i_0)}$ of C_1. For the lifts $p^{*(i)}$ of p^* we choose a box neighborhood B_{p^*} of p^* and also disjoint boxes $B_1, \ldots, B_{m_{C_1}}$ lying above B_{p^*} with $B_i = B_{p^*} \times \left[p^{*(i)} - \delta, p^{*(i)} + \delta\right]$ and a $\delta > 0$. We can assume that B_{p^*} and δ are chosen such that the i-th lift of $C_1 \cap B_{p^*}$ is contained in B_i. For B_{p^*} small enough, it follows that the j_0-th lift of $B_{p^*} \cap C_2$ is also contained in B_{i_0} as $p \in B_{i_0} \cap \overline{M_2}$. As a direct consequence $((B_{p^*} \cap C_1) \times \mathbb{R}) \cap \overline{M_2}$ is the i_0-th lift of $(B_{p^*} \cap C_1)$. Now for any two points p_1^* and p_2^* on C_1 there exists a compact path Σ on C_1, which connects them. Then we consider an open covering of Σ with local neighborhoods $B_{p'}$, $p' \in \Gamma$, such that $((B_{p'} \cap C_1) \times \mathbb{R}) \cap \overline{M_2}$ is the $i_{p'}$-th lift of C_1. Then from restricting to a finite partial covering it follows that $i_{p'} = i_0$ for all p', thus $C_1^{(i_0)} = \overline{M_2} \cap (C_1 \times \mathbb{R})$. Now $M_1 \cap \overline{M_2} \neq \varnothing$ exactly if $M_1 = C_1^{(i_0)}$. $\square$

In case where S contains a vertical line ℓ_p, we also get a decomposition of S into non-singular cells: It consists of lifted elements of $\mathscr{A}_S$ with local degree $n \neq -\infty$, and finitely many vertical lines. However, in general, the boundary property is no longer fulfilled for this decomposition. For a patch M which projects onto a face, adjacent to p, its closure $\overline{M}$ may only contain a single point of ℓ_p, a line segment, a ray or ℓ_p :

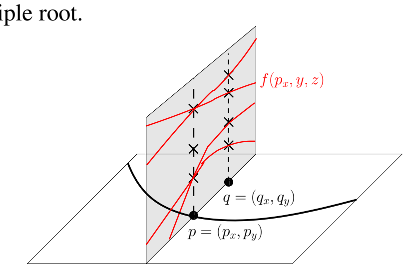
THEOREM 4.2. *Let S contain the vertical line ℓ_p and $F \in \mathscr{A}_S$ be a face, which is adjacent to p. Then for any surface patch $F^{(j)}$ (the j-th lift of F) there exists an interval $I(F^{(j)}) \subset \mathbb{R}$, such that $p \times I(F^{(j)}) = \overline{F^{(j)}} \cap \ell_p$.*

PROOF. Given two points $(p, z_0), (p, z_1) \in \overline{F^{(j)}} \cap \ell_p$, $z_0 < z_1$, on the vertical line, there exist corresponding continuous paths $\Sigma_l \subset F^{(j)}$ with $(p, z_l) \in \overline{\Sigma_l}$ for $l = 0, 1$. Now let (p, z^*) be an arbitrary point in between (p, z_0) and (p, z_1). If we restrict to (end-)parts of Σ_l we can assume that for every point $(q_l, z_{q_l}) \in \Sigma_l$ we have $z_{q_0} < z^*$ and $z_{q_1} > z^*$. We now consider the projection $\Sigma_l^* \subset F$ of Σ_l onto the plane. We further denote B_ε the open disc with radius ε and center p. Then from the definition of F it follows the existence of an $\varepsilon_0 > 0$ such that $\Sigma_\varepsilon := \partial B_\varepsilon \cap \overline{F}$ is connected for all $\varepsilon < \varepsilon_0$. Then Σ_ε intersects Σ_0^* as well as Σ_1^*, thus because of continuity the j-th lift $\Sigma_\varepsilon^{(j)} \subset \overline{F^{(j)}}$ of Σ_ε contains a point s_ε with z-coordinate z^*. It follows that $F^{(j)}$ contains an arc of the z^*-level curve of S, which passes the point (p, z^*). Hence, we must have $(p, z^*) \in \overline{F^{(j)}} \cap \ell_p$. $\square$

Theorem 4.2 shows that in case of a vertical line we still have to decompose the vertical lines into segments to obtain a decomposition Ω_S of S which fulfills the boundary property. In Section 4.3 we show how to determine the intervals $I(F^{(j)})$ and thus, how to decompose the vertical lines.

4.1 Edge-face adjacencies

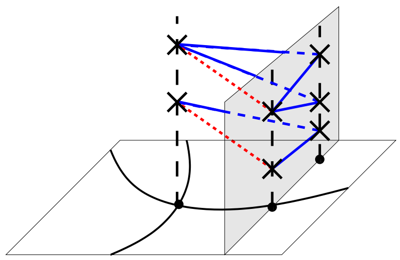
Let E be an edge of $\mathscr{A}_S$, and let F denote an adjacent face in the arrangement $\mathscr{A}_S$. We want to compute the adjacencies between cells above E and cells above F. From the boundary property it suffices to check for an arbitrary point $p = (p_x, p_y) \in E$ if p is adjacent to the lifted surface patch. Therefore, we choose such a sample point with rational x-coordinate p_x (in the case of a vertical edge, we choose a rational y-coordinate and proceed analogously). If the local degree over p is N, and the z-fiber over p has been computed using the m-k-Bitstream Descartes method (compare Section 2), adjacencies are computed similarly to the planar adjacency methods described in [22], [29]. All roots but one of f_p are simple and the cells over E to which they belong have precisely one adjacent surface patch over F. The remaining surface patches must be adjacent to the possibly multiple root.



If f_p was not isolated using the m-k-Bitstream Descartes method, the treatment is the same as in [6]. We choose a rational sample point $q = (q_x, q_y)$ for F with $q_x = p_x$, 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 a segment 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 [22] to compute the adjacency information for $V(f|_{x=p_x})$.

4.2 Adjacencies of a vertex

We consider a vertex point p whose z-fiber is finite, thus there exist finitely many zero-dimensional cells $(p_x, p_y, z_1), \ldots, (p_x, p_y, z_m)$.



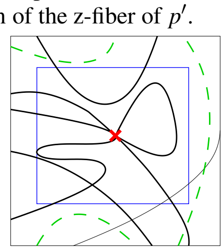
We apply several filters. If $n_p = N$, and p's z-fiber has been constructed using the m-k-Bitstream Descartes method, the adjacencies are computed as described in Section 4.1. Second, adjacencies between p and an edge E can often be derived by a transitivity argument from the combination of adjacencies of E with its adjacent faces F_1 and F_2, and the adjacencies of F_1 and F_2 to p (compare the picture on the right). We skip further details of this simple argument.

If no filter applies, 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 fiber points z_i.

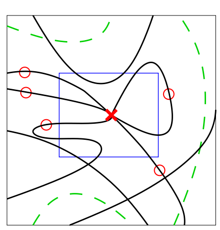
DEFINITION 4.3. *Let $C \in \mathscr{A}_S$ be adjacent to p. A point p' on C is* bucket-faithful *if there exists a path from p' to p on C such that on that path, each cell $C^{(i)} \in \Omega_S$ over C remains in the same bucket.*

With a bucket-faithful point p' on C, the adjacencies of cells over C with cells over p follows by considering the z-fiber of p': if the i-th point over 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-most or the top-most bucket belong to asymptotic components, i.e., they are unbounded in z-direction.

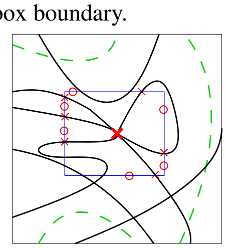
It is easy to prove by an ε-argument that a bucket-faithful point p' exists for each cell C adjacent to p. However, we want to prevent p' being too close to p, as this results in a bad separation of the roots of $f_{p'}$ and thus complicates the computation of the z-fiber of p'.



Observe that p' on C is bucket-faithful if and only if there is a path from p' to p on C that does not intersect any of the *bucket curves* defined by $f(x,y,q_i) \in \mathbb{Q}[x,y]$. We first compute a *bucket box* around p that contains no point of any of the bucket curves (depicted on the right, the bucket curves are drawn as dashed lines), This is easily done with interval arithmetic: Use approximations of p to evaluate $f(p_x, p_y, q_i)$ for all $i = 0, \ldots, m$ until no resulting interval contains zero. The final approximation of p defines the bucket box.



In the second step, we compute bucket-faithful points inside the bucket box for each adjacent cell (note that not each point inside the bucket box is also bucket-faithful). For each adjacent edge, choose an arbitrary sample point, and shrink the box until all these points are outside the box (depicted on the right). After that, each cell has a bucket-faithful point on the box boundary. Compute all intersection points of $\mathscr{A}_S$ with the box boundary.



Follow each edge E starting in p, until it crosses the box boundary. The intersection point is bucket-faithful for E. For a face F, consider the edge $E \in \mathscr{A}_S$ that precedes F in counterclockwise order around p. Let p'' be the bucket-faithful point of E at the box boundary. Let p' be a point on the box boundary between p'' and the next intersection of the box's boundary with $\mathscr{A}_S$ in clockwise order. p' is a bucket-faithful point for F.

The described method does not cover the special case of an isolated vertex p yet. In this case, we compute the intersections of $\mathscr{A}_S$ with the vertical line $x = p_x$, and choose an intermediate value between p_y and the next intersection point above.

Our method for vertex adjacencies has a similar basic idea as the local box algorithm by Collins and McCallum [36] for cads. Still, there are some differences: our construction of the "local box" (which we call bucket box) is more efficient as it only involves interval arithmetic. Also, we have to handle adjacent components that are not x-monotone, which complicates the computation of bucket-faithful points. Moreover, their local box algorithm requires irreducible polynomials as input which implies a preceding factorization step.

4.3 Vertical lines

In the special case where S contains a vertical line ℓ_p, in general, the lift $F^{(i)}$ of a face $F \in \mathscr{A}_S$, adjacent to p in $\mathscr{A}_S$, is no longer adjacent to exactly one lift of p. From Theorem 4.2 it follows that $F^{(i)}$ is adjacent to a connected set $p \times I(F^{(i)})$ on ℓ_p, i.e., a single point, a line segment, a ray or ℓ_p. We define

$$Z'_p := \bigcup_{C \in \mathscr{A}_S \setminus \{p\} : p \in \overline{C}} \left(\bigcup_{i=1,\ldots,m_C} \{z_A | z_A \text{ is an endpoint of } \ell_p \cap \overline{C^{(i)}}\} \right)$$

as the union of all endpoints of intervals $I(F^{(i)})$ and all z-values of endpoints (over p) of lifted arcs in $\mathscr{A}_S$, adjacent to p. In the first step we show how to get a candidate list Z^*_p for Z'_p. Let $I(F^{(i)})$ be an adjacency interval, which consists of more than one point, and $(p, z_0) \in I(F^{(i)})$ be an arbitrary interior point, i.e., $z_0 \notin Z'_p$. Then Theorem 4.2 tells us that the curve $C_{z_0} = \{(x,y) \in \mathbb{R}^2 | f(x,y,z_0) = 0\}$, embedded into the arrangement $\mathscr{A}_S$, contains at least one arc that leaves p and passes the face F. Vice versa, each of these arcs corresponds uniquely to a lifted surface patch above F which is adjacent to (p, z_0). The idea how to get a candidate list of possible endpoints of the intervals $I(F^{(i)})$ is based on the following geometric consideration.

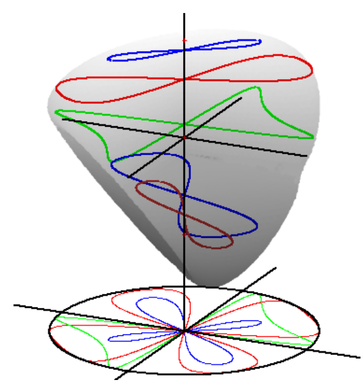


Figure 4.1: Steiner Roman Surface with horizontal intersections at $z = \frac{1}{2}, \frac{2}{5}, \frac{3}{10}, \frac{1}{10}, 0, -\frac{3}{10}, -\frac{2}{5}$

We sweep with a horizontal plane $z = z_0$ along the vertical line and consider the arrangement $\mathscr{A}_{S,z_0}$, denoting the overlay of C_{z_0} and $\mathscr{A}_S$. We are interested in all values z_0 where we detect possible changes of the local topology of $\mathscr{A}_{S,z}$ at p, i.e., we have to detect whenever for any face $F \in \mathscr{A}_S$, the number of arcs of C_z leaving p and passing F changes. For a *generic* z_0 (to be specified), a slight perturbation of z_0 leads to a deformation of C_{z_0} such that the local topology of $\mathscr{A}_{S,z_0}$ at p is preserved. Then arcs $A \subset F$ which correspond to surface patches $F^{(i)}$ are continuously deformed into arcs, that are still contained in F and also correspond to $F^{(i)}$. Hence, $F^{(i)}$ must be adjacent to all points (p,z) in a neighborhood of z_0. In case where z_0 is an endpoint of an interval $I(F^{(i)})$, perturbing z_0 results in either loosing an arc that passes p or in an arc that switches the face.

In the example of Figure 4.1 this happens for $z_0 = \pm\frac{1}{2}$, where we loose arcs, and for $z_0 = 0$, where arcs switch the face.

In the following theorems we specify these ideas and provide an algebraic description for non-generic z_0 with respect to local topology changes of $\mathscr{A}_{S,z_0}$. It turns out that the computed candidate list Z_p^* does not only contain all endpoints of intervals $I(F^{(i)})$, but also the z-values of endpoints (over p) of lifted silhouette arcs, which are adjacent to p. Hence, we obtain a superset Z_p^* of Z_p'.

Let us first state the main result of this section:

THEOREM 4.4. *Let*

$$\begin{aligned} r(x,z) &:= \mathrm{res}_y(f, f_y) = (x-p_x)^{i_0}\tilde{r}(x,z), \\ h(x,z) &:= \mathrm{res}_y(f, \mathrm{res}_z(f, f_z)) = (x-p_x)^{k_0}\tilde{h}(x,z) \end{aligned}$$

with the following definitions of exponents

$$\begin{aligned} i_0 &:= \max\{i : (x-p_x)^i | r(x,z)\}, \\ j_0 &:= \min\{j : \frac{\partial^j f}{\partial y^j}(p_x, p_y, z) \neq 0\} \\ k_0 &:= \max\{k : (x-p_x)^k | h(x,z)\}. \end{aligned}$$

Then for $z_0 \notin Z_p^* := \{z | \tilde{r}(p_x, z) = 0 \vee \frac{\partial^{j_0} f}{\partial y^{j_0}}(p_x, p_y, z) = 0 \vee \tilde{h}(p_x, z) = 0\}$ the local topology of $\mathscr{A}_{S,z_0}$ at p is preserved for any sufficiently small perturbation of z_0 and $Z_p' \subset Z_p^*$.

We assumed S to be square-free and that it does not contain a two-dimensional, vertical component, thus the curve C_z is square-free and does not share a common component with Γ_S for all but finitely many $z \in \mathbb{R}$. As such *degenerate* z-values are exactly given by $\mathrm{res}_y(f, f_y)(x,z) \equiv 0$ or $\mathrm{res}_y(f, \mathrm{res}_z(f, f_z)) \equiv 0$, it follows that the above factorization of $r(x,z)$ and $h(x,z)$ as well as j_0 is well defined. In particular for each $z_0 \in Z_p'$, the curve C_{z_0} is square-free and it neither contains the vertical line $L := V(x-p_x) \subset \mathbb{R}^2$ nor any component of Γ_S.

We split the proof of Theorem 4.4 as follows. Theorem 4.5 shows that Z_p^* contains all z-values of endpoints (over p) of lifted silhouette arcs, adjacent to p, as well as all $z_0 \in I(F^{(i)})$, where $I(F^{(i)})$ consists of only one point. Then, in Theorem 4.6, we prove our claim about preserving the topology which finally leads to a proof that Z_p^* contains the endpoints of intervals $I(F^{(i)})$, which consists of more than one point.

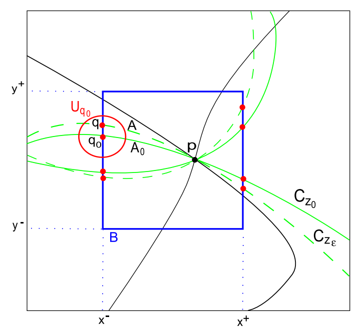
THEOREM 4.5. *Let $L \subset \mathbb{R}^2$ denote the line $x = p_x$ and let $C \subset \Gamma_S$ be a component of the silhouette, adjacent to p. Then the endpoint (over p) of any lift $C^{(i)}, L^{(i)}$ is contained in $p \times Z_p^{**}$ with $Z_p^{**} := \{\frac{\partial^{j_0} f}{\partial y^{j_0}}(p_x, p_y, z) = 0 \vee \tilde{h}(p_x, z) = 0\} \subset Z_p^*$. Furthermore, for each surface patch $F^{(i)}$ that is connected with exactly one point $(p, z_0) \in \ell_p$, it holds that $z_0 \in Z_p^{**}$.*

PROOF. For any sequence $p_n := (x_n, y_n) \in \Gamma_S \backslash L$, $\lim_{n\to\infty} p_n = p$, and any lift $p_n^{(i)} := (x_n, y_n, z_n^{(i)})$ we must have that $f(p_n^{(i)}) = \mathrm{res}_z(f, f_z)(p_n) = 0$, thus $h(x_n, z_n^{(i)}) = (x_n - p_x)^{k_0}\tilde{h}(x_n, z_n^{(i)}) = 0$. It follows that $\tilde{h}(x_n, z_n^{(i)}) = 0$, so if we pass to the limit, we obtain $\tilde{h}(p_x, \lim_{n\to\infty} z_n^{(i)}) = 0$. This shows that the lift of any component of Γ_S, distinct from L, runs into a point $(p, z) \in p \times Z_p^{**}$ on ℓ_p. Now we consider a sequence $p_n := (p_x, y_n) \in L \backslash \{p\}$ of points on L, that converges towards p. Then, for any lift $p_n^{(i)} := (p_x, y_n, z_n^{(i)})$ of p_n we must have $f(p_n^{(i)}) = (y_n - p_y)^{j_0}\tilde{f}(y_n, z_n^{(i)}) = 0$ with $\tilde{f}(y,z) := \frac{f(p_x,y,z)}{(y-p_y)^{j_0}}$. It follows that $\tilde{f}(y_n, z_n^{(i)}) = 0$, thus $\lim_{n\to\infty} \tilde{f}(y_n, z_n^{(i)}) = \tilde{f}(p_y, \lim_{n\to\infty} z_n^{(i)}) = 0$ and $\frac{\partial^{j_0} f}{\partial y^{j_0}}(p_x, p_y, \lim_{n\to\infty} z_n^{(i)}) = 0$. Finally, for a face $F \in \mathscr{A}_S$, that is adjacent to p, its closure either contains a silhouette arc C, adjacent to p or some part of L that contains p. If we assume that the i-th lift of F is connected with exactly one point on ℓ_p, this also holds for either a lift of C or L. As we already have shown that endpoints (over p) of such lifts are contained in $p \times Z_p^{**}$, our last claim follows. $\square$

In Theorem 4.4 we stated that for $z_0 \notin Z_p^*$, any sufficiently small deformation of z_0 does not change the local topology of $\mathscr{A}_{S,z_0}$ at p:

THEOREM 4.6. *Let $z_0 \notin Z_p^*$, then there exists an $\varepsilon > 0$, such that for all ε-approximations z_ε of z_0, the local topology of $\mathscr{A}_{S,z_0}$ at p does not change. Furthermore, if an arc $A_0 \subset F \in \mathscr{A}_S$ of C_{z_0} corresponds to a surface patch $F^{(i)}$ (i.e., it is the projection of a z_0-level curve on $F^{(i)}$ onto F) then A_0 continuously deforms into an arc A of C_{z_ε}, which also corresponds to $F^{(i)}$.*

PROOF: As $\tilde{r}(p_x, z_0), \tilde{h}(p_x, z_0) \neq 0$ and $\frac{\partial^{j_0} f}{\partial y^{j_0}}(p_x, p_y, z_0) \neq 0$ there exists an $\varepsilon > 0$ and $x^- < p_x < x^+$, such that $r(x,z), h(x,z) \neq 0$ and $\frac{\partial^{j_0} f}{\partial y^{j_0}}(p_x, p_y, z_\varepsilon) \neq 0$ for all $x \in [x^-, x^+] \backslash \{p_x\}$ and z_ε an arbitrary ε-approximation of z_0. As for any z_ε the root p_y of $f(p_x, y, z_\varepsilon)$ has multiplicity j_0, it follows the existence of $y^- < p_y < y^+$ such that $[y^-, y^+]$ is an isolating interval for the real root p_y of $f(p_x, y, z_\varepsilon)$. Now if we restrict to the rectangle $B := [x^-, x^+] \times [y^-, y^+]$ we obtain an isolating area for the x-critical point p of C_{z_ε}, i.e., for each point $(x,y) \in C \cap B \backslash \{p\}$ its y-value is an ordinary root of $f(x, y, z_\varepsilon)$.



Furthermore B is also an isolating area for the intersection point p of C_{z_ε} with the silhouette Γ_S. W.l.o.g. we can assume that B has been chosen small enough such that $\mathscr{A}_{S,z_\varepsilon}$ has starshape within B, i.e., all points $\{\Gamma_S \cup C_{z_\varepsilon}\} \cap B$ are connected by arcs of $\Gamma_S \cup C_{z_\varepsilon}$ with p. We can further assume that all intersection points of C_{z_ε} with ∂B are on the left or on the right edge of ∂B. Thus we get a one-to-one correspondence between arcs $A \subset C_{z_\varepsilon}$ and points $(x_A, y_A) \in C_{z_\varepsilon} \cap \partial B$ with $x_A \in \{x^-, x^+\}$. As y_A is an ordinary root of $f(x_A, y, z_\varepsilon)$ it follows that for sufficiently small ε the number of such points (x_A, y_A) as well as the number of arcs that leave p stays the same for all z_ε. This shows that the local topology of $\mathscr{A}_{S,z_\varepsilon}$ at p does not change (for details we refer to [21] and [22]), proving the first part of the theorem.

For the second claim let us consider an arbitrary arc $A_0 \subset C_{z_0}$. Then A_0 corresponds to a surface patch $F^{(i)}$, i.e., the i-th lift of its corresponding point $q_0 := (x_{A_0}, y_{A_0}) \in A_0 \cap \partial B$ lies on the z_0-level curve of S. We choose a neighborhood $U_{q_0} \subset F$ of q_0 which fulfills the following two conditions:

- U_{q_0} contains no point (x_A, y_A) that corresponds to an arc $A \subset C_{z_0}$, different from A_0
- There exist open, isolating intervals $I_1, \ldots, I_{m_F} \subset \mathbb{R}$ for the roots of all polynomials $f(q,z)$ where $q \in U_{q_0}$ (in particular we have $z_0 \in I_i$).

The first condition can trivially be fulfilled as q_0 is an interior point of F. For the second condition we remark that $f(q_0, z_\varepsilon)$ is a square-free polynomial, thus isolating intervals for its real roots remain isolating for $f(q, z_\varepsilon)$ for any sufficiently small approximation q of q_0. Now we can choose ε small enough, such that

- U_{q_0} contains exactly one point $q_{z_\varepsilon} := (x_A, y_A)$ that corresponds to an arc $A \subset C_{z_\varepsilon}$.
- $[z_0 - \varepsilon, z_0 + \varepsilon] \subset I_i$

The preceding conditions are a direct consequence of the fact that the set of points (x_A, y_A) continuously deform with varying z_ε and that we can choose ε sufficiently small.

Now the point q_{z_ε} corresponds to an arc $A \subset F$ of C_{z_ε}, thus there exists a lift $q_{z_\varepsilon}^{(j)} \subset F^{(j)}$ on the z_ε-level curve. From the properties of U_{q_0} it follows that $i = j$, thus $F^{(i)}$ is adjacent to (p, z_ε). This shows that $F^{(i)}$ is adjacent to all points (p, z_ε). $\square$

We can now proof the central result:

PROOF OF THEOREM 4.4. Theorem 4.5 already tells us that z-values of endpoints on ℓ_p of lifted silhouette arcs and lifted surface patches, adjacent to exactly one point on ℓ_p are contained in $Z_p^{**} \subset Z_p^*$. Thus it remains to show that a given $z \notin Z_p^*$ cannot be an endpoint of an interval $I(F^{(i)})$ that consists of more than one point. We prove by contradiction, so assume that z_0 is an endpoint of $I(F^{(i)})$, then, from Theorem 4.6, there exists an ε-neighborhood $U_\varepsilon(z_0)$ of z_0 such that for all $z \in U_\varepsilon(z_0)$ the local topology of $\mathscr{A}_{S,z}$ at p is preserved. Furthermore each arc of C_z, that correspond to a surface patch $F^{(i)}$ continuously deforms into an arc, that also corresponds to the same patch. As we assumed z_0 to be an endpoint of $I(F^{(i)})$ the neighborhood $U_\varepsilon(z_0)$ must contain an interior point $\dot{z}$ of $I(F^i)$. Then, from Theorem 4.2, we know that $C_{\dot{z}}$ contains an arc, which leaves p and passes the face F. It follows that there exists a corresponding arc of C_z for any $z \in U_\varepsilon(z_0)$. But this shows that $I(F^{(i)})$ contains $U_\varepsilon(z_0)$, a contradiction. $\square$

As a consequence of Theorem 4.4, we can define our cell decomposition Ω_S in general.

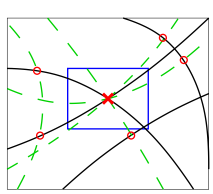
DEFINITION 4.7. *Let S be a surface with (n,k)-arrangement $\mathscr{A}_S$. Let V be the set of vertices in $\mathscr{A}_S$ whose lifts are vertical lines. For $p \in V$, let ω_p denote the partition of ℓ_p into elements of Z_p^* and their induced intervals of $\mathbb{R}$. We define*

$$\Omega_S := \bigcup_{C \in \mathscr{A}_s \setminus V} \left(\bigcup_{i=1,\dots,m_C} \{C^{(i)}\} \right) \cup \bigcup_{p \in V} \omega_p$$

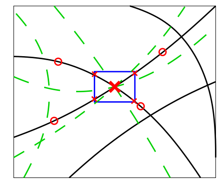
By construction of Z_p^*, Ω_S has the boundary property. We can also show that vertical lines do not increase the complexity.

THEOREM 4.8. *The number of cells of Ω_S is $O(N^5)$.*

PROOF. Using Corollary 3.8, it remains to show that the vertical lines do not introduce more than $O(N^5)$ cells. The number of vertices with vertical lines is in $O(N^2)$. For a fixed p, the set Z_p^* is the union of the roots of three polynomials in z (compare Theorem 4.4), whose degree is at most $O(N^3)$. $\square$



Adjacencies for vertical line cells: Let p denote a vertex in $\mathscr{A}_S$ having a vertical line. We proceed similar to Section 4.2 by defining bucket values q_i and bucket curves $V(f(x,z,q_i))$ for each intermediate value between elements of Z_p^*. There is a complication here, as all bucket curves now are intersecting p, and we cannot build a bucket box as before. Instead, we compute the overlay of $\mathscr{A}_S$ with all bucket curves, and build a box around p that does not contain an intersection of $\mathscr{A}_S$ with any bucket curve, except at p itself.



For the sample points of edges from $\mathscr{A}_S$, we further proceed as in Section 4.2. Choose points at each adjacent cell of $\mathscr{A}_S$ and shrink the box until they are outside. Then traverse the edges starting p and choose the first box intersection as sample point for the edge. This point is bucket-faithful (recall Definition 4.3) and reveals the adjacencies between the lifted cells over the edge with the cells at the vertical line, which is valid due to the construction of Z_p^*.

For an adjacent face F, we first compute which patches $F^{(j)}$ over F are adjacent to whole vertical segments. Each such vertical segment contains one of the bucket values q_i. Thus, a patch over F that is adjacent to an interval causes an arc of the bucket curve for q_i that lies in F and ends in p. We proceed as follows. Iterate over the arcs of all bucket curves in F that leave p. Let q_i be the bucket value of the currently considered bucket curve. Choose a sample point on the bucket curve (inside the bucket box), build the z-fiber over it, and determine which patch $F^{(j)}$ has the z-coordinate q_i. Mark this patch to be adjacent to the vertical segment containing q_i, and also to the two endpoints of the segment.

Finally, when all patches adjacent to an interval are detected, consider the remaining patches. They are adjacent to some zero-dimensional cell over p. Choose a bucket-faithful point for the face (analogous to Section 4.2), and determine the buckets which the remaining patches belong to.

5. IMPLEMENTATION AND RESULTS

Often, implementations of algorithms in this area of research are lacking, or do exclude certain degeneracies, like vertical lines or singularities. Our presented algorithm is transformed into a fully working C++-implementation, based on the projects CGAL and EXACUS.[3] Algebraic surfaces are represented by EXACUS' class template `Algebraic_surface_3`. Algebraic curves are taken from EXACUS' ALCIX library that implements recent work by Eigenwillig et. al. [21, 22]. To construct and refine the (n,k)-arrangement for a surface S using CGAL's `Arrangement_2` package [42], we rely on EXACUS' ability to provide a model of CGAL's ARRANGEMENTTRAITS_2 concept for algebraic curves for arbitrary degree. This feature is essential for the projection step of our algorithm.

Arrangements in CGAL integrate the faces, edges, and vertices by a *doubly-connected-edge-list* (DCEL) that is extended with geometric data. For technical reasons, curves are split into x-monotone subcurves. Our traversal combines them to maximal (n,k)-constant paths. We make extensive usage of advanced operations on arrangements [43]. For example, we attach a collection of information (e.g., n_C and k_C) to each DCEL-component. In combination with CGAL's overlay mechanism, the computation $\mathscr{A}_S$ can be implemented as explained in Section 3. Additionally, the construction of z-fibers as presented in Section 2 benefits from the precomputed parameters n_C and k_C for each cell. This avoids to repeat costly tests, e.g., whether a point lies on some curves. We also follow the scheme of lazy-evaluation, e.g., the sample point for a cell and its z-fiber is only computed on demand, and then cached.

We shortly want to mention, that our design of implementation decouples combinatorial and generic tasks from surface-specific ones using the generic programming paradigm [7]. In particular, three tasks, that follow our algorithmic description, are expected from a supported surface. First, decompose the polynomials $\mathrm{res}_z(f, \frac{\partial f}{\partial z})$, a_i, and $\mathrm{stha}_i(f_n)$ into square-free factors and construct corresponding curve instances. Second, a surface is required

[3] See project homepages at `www.cgal.org` and `www.mpi-inf.mpg.de/EXACUS`

Instance	$\deg_{x,y,z}$	(#V,#E,#F)	$\lvert\Omega_S\rvert$	t
steiner-roman	2,2,2	(5,12,8)	28	**0.73**
cayley-cubic	2,2,2	(3,10,8)	31	**0.74**
dupin-cyclide	4,4,4	(3,4,4)	10	**0.19**
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**
star	6,6,6	(1,1,2)	5	**3.61**
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**

Table 1: Complexity and running times (in seconds) for a selection of surfaces. Some defining polynomials can be found in Appendix A.

to construct a z-fiber for given p, knowing n_p and k_p. Third, for two adjacent cells of $\mathscr{A}_S$, it has to compute their lifted adjacencies (see Section 4 for details). The newly written code consists of about 15,000 lines C++. It will be published with a future release of CGAL.

Experiments: We also run experiments on our implementation on well-known examples from algebraic geometry,[4] and interpolated instances, and also a generic projection of two quadrics in 4D. All experiments are executed on an AMD Dual-Core Opteron(tm) 8218 (1 GHz) multi-processor platform. Each processor has an internal cache of 1 MB and the total memory consists of 32 GB. The system runs Debian Etch. We compiled using `g++-4.1.2` with flags `-O2 -DNDEBUG` and use the exact number types of CORE [32]. Observe that our software currently does not benefit from having several processors, although many steps of the algorithm are well-suited for parallel computations.

Table 1 states for a selection of tested surfaces the size of the (n,k)-arrangement $\mathscr{A}_S$, the total number of cells in Ω_S, and the obtained running times. It is also expected, that (some) surfaces do not show any (n,k)-vertex (e.g., `tangle-cube`), or -edge (e.g., *xy*-functional surfaces) at all. Concerning the running times, we observed that about 90% is spent to construct $\mathscr{A}_S$. This is no surprise, as we have to analyze plane algebraic curves of degree up to $N(N-1)$. The remaining 10% are spent to compute lifts and adjacencies, which allows to conclude that these steps benefit from our approximative and combinatorial methods.

Conclusion and outlook

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 saving cell decomposition and the consequent application of approximate methods. We consider our result to serve as a basis for solving related problems. For instance, 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. For the future, we plan to work on these theoretical tasks, and to augment our implementation towards multiple surfaces.

[4] Subsets of the tested example surfaces are provided courtesy of INRIA by the AIM@SHAPE Shape Repository, by `www.singsurf.org`, by `www.freigeist.cc`, and by [38]

Acknowledgements We thank the referees for their excellent reviews and all EXACUS-developers for their great supporting work.

6. REFERENCES

[1] J. G. Alcázar, J. Schicho, J. R. Sendra: "A delineability-based method for computing critical sets of algebraic surfaces". *Journal of Symbolic Computation* **42** (2007) 678–691.

[2] J. G. Alcázar, R. Sendra: "Computation of the Topology of Real Algebraic Space Curves". *Journal of Symbolic Computation* **39** (2005) 719–744.

[3] D. S. Arnon: "A Cluster-Based Cylindrical Algebraic Decomposition Algorithm". *Journal of Symbolic Computation* **5** (1988) 189–212.

[4] D. S. Arnon, G. E. Collins, S. McCallum: "Cylindrical Algebraic Decomposition I: The Basic Algorithm". *SIAM Journal on Computing* **13** (1984) 865–877. Reprinted in [15], pp. 136–151.

[5] D. S. Arnon, G. E. Collins, S. McCallum: "Cylindrical Algebraic Decomposition II: An Adjacency Algorithm for the Plane". *SIAM Journal on Computing* **13** (1984) 878–889. Reprinted in [15], pp. 152–165.

[6] D. S. Arnon, G. E. Collins, S. McCallum: "An Adjacency Algorithm for Cylindrical Algebraic Decompositions of Three-Dimensional Space". *Journal of Symbolic Computation* **5** (1988) 163–187.

[7] M. H. Austern: *Generic Programming and the STL*. Addison-Wesley, 1999.

[8] C. L. Bajaj, G. Xu: "Spline Approximations of Real Algebraic Surfaces". *Journal of Symbolic Computation* **23** (1997) 315–333.

[9] S. Basu, R. Pollack, M.-F. Roy: *Algorithms in Real Algebraic Geometry*, *Algorithms and Computation in Mathematics*, vol. 10. Springer, 2nd edn., 2006.

[10] E. Berberich, M. Hemmer, L. Kettner, E. Schömer, N. Wolpert: "An Exact, Complete and Efficient Implementation for Computing Planar Maps of Quadric Intersection Curves". In: J. Mitchell, G. Rote, L. Kettner (eds.) *21st Annual Symposium on Computational Geometry (SCG'05)*. Association for Computing Machinery (ACM), ACM, Pisa, Italy, 2005 99–106.

[11] J.-D. Boissonnat, D. Cohen-Steiner, B. Mourrain, G. Rote, G. Vegter: "Meshing of Surfaces". In: J.-D. Boissonnat, M. Teillaud (eds.) *Effective Computational Geometry for Curves and Surfaces*, 181–229. Springer, 2007.

[12] G. E. Bredon: *Topology and Geometry*. Springer, 1993.

[13] C. W. Brown: "Improved projection for cylindrical algebraic decomposition". *Journal of Symbolic Computation* **32** (2001) 447–465.

[14] C. W. Brown: "Constructing Cylindrical Algebraic Decompositions of the Plane Quickly", 2002. URL `http://www.cs.usna.edu/~wcbrown/`. Unpublished.

[15] B. F. Caviness, J. R. Johnson (eds.): *Quantifier Elimination and Cylindrical Algebraic Decomposition, Texts and Monographs in Symbolic Computation*. Springer, 1998.

[16] J.-S. Cheng, X.-S. Gao, M. Li: "Determining the Topology of Real Algebraic Surfaces". In: R. Martin, H. Bez, M. Sabin (eds.) *11. IMA Conference on the Mathematics of Surfaces*, *LNCS*, vol. 3604, 2005 121–146.

[17] G. E. Collins: "Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition". In: *Second GI Conference on Automata Theory and Formal Languages*, *LNCS*, vol. 33, 1975 134–183. Reprinted in [15], pp. 85–121.

[18] G. E. Collins, J. R. Johnson, W. Krandick: "Interval Arithmetic in Cylindrical Algebraic Decomposition". *Journal of Symbolic Computation* **34** (2002) 145–157.

[19] D. Diochnos, I. Z. Emiris, E. P. Tsigaridas: "On the Complexity of Real Solving Bivariate Systems". In: C. W. Brown (ed.) *Proocedings of the 2007 International Symposium on Symbolic and Algebraic Computation (ISSAC 2007)*, 2007 127–134.

[20] A. Eigenwillig: *Real Root Isolation for Exact and Approximate Polynomials Using Descartes' Rule of Signs*. Ph.D. thesis, Universität des Saarlandes, Germany, 2008.

[21] A. Eigenwillig, M. Kerber: "Exact and Efficient 2D-Arrangements of Arbitrary Algebraic Curves". In: *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA08)*, 2008 122–131.

[22] A. Eigenwillig, M. Kerber, N. Wolpert: "Fast and Exact Geometric Analysis of Real Algebraic Plane Curves". In: C. W. Brown (ed.) *Proocedings of the 2007 International Symposium on Symbolic and Algebraic Computation (ISSAC 2007)*, 2007 151–158.

[23] A. Eigenwillig, L. Kettner, W. Krandick, K. Mehlhorn, S. Schmitt, N. Wolpert: "A Descartes Algorithm for Polynomials with Bit-Stream Coefficients". In: *8th International Workshop on Computer Algebra in Scientific Computing (CASC 2005)*, *LNCS*, vol. 3718, 2005 138–149.

[24] M. El Kahoui: "Topology of Real Algebraic Space Curves". *Journal of Symbolic Computation* (2007). In press.

[25] E. Fortuna, P. Gianni, D. Luminati: "Algorithmical determination of the topology of a real algebraic surfaces". *Journal of Symbolic Computation* **38** (2004) 1551–1567.

[26] E. Fortuna, P. Gianni, P. Parenti, C. Traverso: "Algorithms to compute the topology of orientable real algebraic surfaces". *Journal of Symbolic Computation* **36** (2003) 343–364.

[27] G. Gatellier, A. Labrouzy, B. Mourrain, J.-P. Técourt: "Computing the topology of 3-dimensional algebraic curves". In: *Computational Methods for Algebraic Spline Surfaces*. Springer, 2005 27–44.

[28] L. Gonzalez-Vega, M. El Kahoui: "An Improved Upper Complexity Bound for the Topology Computation of a Real Algebraic Plane Curve". *Journal of Complexity* **12** (1996) 527–544.

[29] L. Gonzalez-Vega, I. Necula: "Efficient Topology Determination of Implicitly Defined Algebraic Plane Curves". *Computer Aided Geometric Design* **19** (2002) 719–743.

[30] L. Gonzalez-Vega, T. Recio, H. Lombardi, M.-F. Roy: "Sturm-Habicht Sequences, Determinants and Real Roots of Univariate Polynomials". In [15], pp. 300–316.

[31] H. Hong: "An Efficient Method for Analyzing the Topology of Plane Real Algebraic Curves". *Mathematics and Computers in Simulation* **42** (1996) 571–582.

[32] V. Karamcheti, C. Li, I. Pechtchanski, C. Yap: "A Core Library for Robust Numeric and Geometric Computation". In: *Proceedings of the 15th Annual ACM Symposium of Computational Geometry (SCG)*, 1999 351–359.

[33] R. Loos: "Generalized polynomial remainder sequences". In: B. Buchberger, G. Collins, L. Loos (eds.) *Computer Algebra – Symbolic and Algebraic Computation*, 115–138. Springer, 1982.

[34] W. Massey: *Algebraic Topology: An Introduction*. Springer, 1967.

[35] S. McCallum: "An Improved Projection Operation for Cylindrical Algebraic Decomposition". In [15], pp. 242–268.

[36] S. McCallum, G. E. Collins: "Local Box Adjacency Algorithms for Cylindrical Algebraic Decompositions". *Journal of Symbolic Computation* **33** (2002) 321–342.

[37] B. Mourrain, J.-P. Técourt: *Isotopic Meshing of a real algebraic surface*. Technical Report 5508, INRIA Sophia-Antipolis, 2005.

[38] S. Plantinga, G. Vegter: "Isotopic meshing of implicit surfaces". *The Visual Computer* **23** (2007) 45–58.

[39] R. Seidel, N. Wolpert: "On the Exact Computation of the Topology of Real Algebraic Curves". In: *Proceedings of the 21st Annual ACM Symposium on Computational Geometry (SCG 2005)*, 2005 107–115.

[40] B. T. Stander, J. C. Hart: "Guaranteeing the topology of an implicit surface polygonization for interactive modeling". In: *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 1997 279–286.

[41] A. W. Strzebonski: "Cylindrical Algebraic Decomposition using validated numerics". *Journal of Symbolic Computation* **41** (2006) 1021–1038.

[42] R. Wein, E. Fogel, B. Zukerman, D. Halperin: "2D Arrangements". In: *CGAL-3.3 User and Reference Manual*, 2007. URL `http://www.cgal.org/Manual/3.3/doc_html/cgal_manual/Arrangement_2/Chapter_main.html`.

[43] R. Wein, E. Fogel, B. Zukerman, D. Halperin: "Advanced Programming Techniques Applied to CGAL's Arrangement Package". *Computational Geometry — Theory and Applications* **38** (2007) 37–63.

[44] C. K. Yap: "Robust Geometric Computation". In: J. E. Goodman, J. O'Rourke (eds.) *Handbook of Discrete and Computational Geometry*, chap. 41, 927–952. CRC Press, 2nd edn., 2004.

APPENDIX

A. SURFACES

We finally give the defining polynomials of some example surfaces that we analyzed in Section 5.

steiner-roman $f=(y^2+(x^2))\cdot z^2+(((1)\cdot x)\cdot y)\cdot z+((x^2)\cdot y^2)$

cayley-cubic $f=(5\cdot y+(5\cdot x))\cdot z^2+(5\cdot y^2+(-2)\cdot y+(5\cdot x^2+(-2)\cdot x))\cdot z+((5\cdot x)\cdot y^2+(5\cdot x^2+(-2)\cdot x)\cdot y)$

dupin-cyclic $f=447279\cdot z^4+(894558\cdot y^2+(894558\cdot x^2+(-1155200)\cdot x+1155200))\cdot z^2+(447279\cdot y^4+(894558\cdot x^2+(-1155200)\cdot x+(-1155200))\cdot y^2+(447279\cdot x^4+(-1155200)\cdot x^3+(-1404800)\cdot x^2+5120000\cdot x+(-2560000)))$

tangle-cube $f=z^4+(-5)\cdot z^2+(y^4+(-5)\cdot y^2+(x^4+(-5)\cdot x^2+10))$

bohemian-dome $f=z^4+(2\cdot y^2+((-2)\cdot x^2))\cdot z^2+((-1)\cdot y^4+(2\cdot x^2+(-4))\cdot y^2+(x^4))$

chair $f=16\cdot z^4+(288\cdot y^2+(288\cdot x^2+(-600)))\cdot z^2+((-1280)\cdot y^2+(1280\cdot x^2))\cdot z+(80\cdot y^4+((-96)\cdot x^2+(-600))\cdot y^2+(80\cdot x^4+(-600)\cdot x^2+5125))$

hunt $f=4\cdot z^6+(12\cdot y^2+(12\cdot x^2+276))\cdot z^4+(12\cdot y^4+(24\cdot x^2+(-528))\cdot y^2+(12\cdot x^4+(-960)\cdot x^2+4620))\cdot z^2+(4\cdot y^6+(12\cdot x^2+(-129))\cdot y^4+(12\cdot x^4+(-150)\cdot x^2+1380)\cdot y^2+(4\cdot x^6+87\cdot x^4+84\cdot x^2+(-4900)))$

star $f=100\cdot z^6+(300\cdot y^2+(300\cdot x^2+(-300)))\cdot z^4+(300\cdot y^4+(600\cdot x^2+(-599))\cdot y^2+(300\cdot x^4+(-599)\cdot x^2+300))\cdot z^2+(100\cdot y^6+(300\cdot x^2+(-300))\cdot y^4+(300\cdot x^4+(-599)\cdot x^2+300)\cdot y^2+(100\cdot x^6+(-300)\cdot x^4+300\cdot x^2+(-100)))$

spiky $f=z^6+((-3)\cdot y^3+(3\cdot x^2))\cdot z^4+(3\cdot y^6+(21\cdot x^2)\cdot y^3+(3\cdot x^4))\cdot z^2+((-1)\cdot y^9+(3\cdot x^2)\cdot y^6+((-3)\cdot x^4)\cdot y^3+(x^6))$

C8 $f=32\cdot z^8+(-64)\cdot z^6+40\cdot z^4+(-8)\cdot z^2+(32\cdot y^8+(-64)\cdot y^6+40\cdot y^4+(-8)\cdot y^2+(32\cdot x^8+(-64)\cdot x^6+40\cdot x^4+(-8)\cdot x^2+1))$