# Cell decomposition of almost smooth real algebraic surfaces

Gian Mario Besana* Sandra Di Rocco† Jonathan D. Hauenstein‡
Andrew J. Sommese§ Charles W. Wampler ¶

July 12, 2011

**Abstract**

We present a numerical algorithm to decompose almost smooth real algebraic surfaces into 2-cells. Each 2-cell (a face) has a generic interior point and a boundary consisting of 1-cells (edges). Similarly, the 1-cells in turn have a generic interior point and a vertex at each end. Each 1-cell and each 2-cell has an associated homotopy for moving the generic interior point to any other point in the interior of the cell. This work draws on previous results for the curve case. The algorithm works for surfaces in any dimension, i.e., in $\mathbb{R}^N$. Once the cell decomposition is in hand, one can sample the 2-cells and 1-cells to any resolution, limited only by the computational resources available.

## Contents



*College of Computing and Digital Media, DePaul University, 243 S Wabash, Chicago IL 60604 (gbesana@depaul.edu, `www.depaul.edu/~gbesana`)

†Department of Mathematics, KTH, S-10044 Stockholm, Sweden (dirocco@math.kth.se, `www.math.kth.se/~dirocco`). This author was supported by the Mittag-Leffler Institute, the University of Notre Dame, and VR grant NT:2010-5563.

‡Department of Mathematics, Mailstop 3368, Texas A&M University, College Station, TX 77843 (jhauenst@math.tamu.edu, `www.math.tamu.edu/~jhauenst`). This author was supported by Texas A&M University, the Mittag-Leffler Institute, and NSF grant DMS-0915211.

§Department of Mathematics, University of Notre Dame, Notre Dame, IN 46556 (sommese@nd.edu, `www.nd.edu/~sommese`). This author was supported by the Mittag-Leffler Institute, the Duncan Chair of the University of Notre Dame, and NSF grant DMS-0712910

¶General Motors Research and Development, Mail Code 480-106-224, 30500 Mound Road, Warren, MI 48090 (charles.w.wampler@gm.com, `www.nd.edu/~cwample1`) This author was supported by the Mittag-Leffler Institute and NSF grant DMS-0712910.

# 1 Introduction

Let

$$f(z) := \begin{bmatrix} f_1(z_1, \ldots, z_N) \\ \vdots \\ f_\nu(z_1, \ldots, z_N) \end{bmatrix} = 0 \tag{1}$$

be a polynomial system with each $f_i$ having real coefficients. In this article we give a numerical algorithm to decompose the two-dimensional set of real points of a two-dimensional irreducible complex component $Z \subset \mathbb{C}^N$ of the solution set of $f(z) = 0$ under the assumptions that

1. $Z$ is self-conjugate, i.e., $Z$ is taken to itself under conjugation; and
2. the singularities of the solution set of $f(z) = 0$ meet $Z$ in at most a finite set.

In effect we are assuming that $Z$ is smooth except on a finite set and $Z$ meets other components of $f^{-1}(0)$ in at most a finite set. Though these assumptions are restrictive, they cover many cases in higher dimensional spaces. They avoid the computational expense caused by the multiplicity of sets in the construction being more than one and the difficulties that occur when one and two dimensional are blended together, e.g., the Whitney umbrella. See Remark 3.1 for how these conditions may be checked by using algorithms on the system of equations (1) that are implemented in Bertini [4].

We use projections and critical sets along the general lines starting in the Tarski-Seidenberg elimination of quantifiers (see [2]). We use an approach motivated by the classical application of Morse theory [1, 9, 6] to prove the First Lefschetz Theorem on hyperplane sections. This is in line with the numerical approach to the curve case [8].

# 2 Background

In this section we collect some background material. Given a polynomial system $f(z) = 0$ as in (1), we let $V(f)$ denote the underlying solution set which is an algebraic set and we let $f^{-1}(0)$ denote the set $V(f)$ with its full non-reduced structure.

We use $\mathbb{P}^k$ to refer to $k$-dimensional complex projective space and $\mathbb{P}^k_{\mathbb{R}}$ to refer to $k$-dimensional real projective space.

## 2.1 Witness sets

Let $Z$ be an irreducible component of the solution set of the system $f(z) = 0$ given in (1). Assume for simplicity that $Z$ is a multiplicity one component of the solution set of $f(z) = 0$: this assumption is sufficient for the results of this article.

By a witness set for $Z$ we mean a triple $\{f, L, W\}$ where $L(z)$ is a system of $\dim Z$ random affine linear equations and $W$ is the intersection of the solution set $\mathcal{L} = V(L)$ with $Z$.

By genericity, the affine linear space $\mathcal{L}$ meets $Z$ transversely in a finite set of $d = \deg Z$ points $\{w_1, \ldots, w_d\}$ contained in the smooth points of $Z$.

Corresponding to the decomposition of the solution set $V(f)$ into irreducible components is the numerical irreducible decomposition developed in a series of articles by Sommese, Verschelde, and Wampler of which the major ones are [10, 12]. See [14, 15] for detailed expositions of the numerical irreducible decomposition, which computes witness sets for the components of $V(f)$, and for the main developments of the field of Numerical Algebraic Geometry up to the present. The numerical decomposition contains the degree, dimension, and multiplicity information on the irreducible components of $V(f)$, and may be used to compute other quantities of interest in applications and algebraic geometry, e.g., see [5] for the computation of the genus of a desingularization of a one-dimensional component of $V(f)$.

If we have a witness set $\{f, L, W\}$ for an irreducible component $Z$ of $V(f)$, then we can find a witness set for $Z \cap V(L')$ for any other full rank set of $\dim Z$ affine linear equations $L'(z)$ by a homotopy that moves $L$ to $L'$. In particular, for any linear projection, $p$, onto $\mathbb{C}^k$, $k \leq \dim Z$, we can find a witness set for $Z \cap p^{-1}(\alpha)$ by making a subset of the equations in $L'$ be $p(z) - \alpha$.

Also, we can compute a witness point superset for $(Z \times \mathbb{P}^k) \cap V$, where $V = g^{-1}(0)$ for some $g(z, u) : \mathbb{C}^N \times \mathbb{C}^{k+1} \to \mathbb{C}^k$, with $g(z, u)$ homogeneous in $u$. One approach is to find a witness set for $V$ and use the diagonal intersection homotopy [13]. It is also possible to compute this intersection without pre-computing a witness set for $V$ [7].

### 2.2 Linear projections

An affine linear projection on Euclidean space is a map $p : \mathbb{C}^N \to \mathbb{C}^k$ defined by

$$p(z_1, \ldots, z_N) = A \cdot \begin{bmatrix} z_1 \\ \vdots \\ z_N \end{bmatrix}$$

where $A$ is a $k \times N$ matrix of rank $k$. If $A$ is real, then the linear projection is also a map $p : \mathbb{R}^N \to \mathbb{R}^k$. In this article we only need some very special cases of linear projections, namely

$$\begin{aligned} \pi(z_1, \ldots, z_N) &= (z_1, z_2) \\ \pi_1(z_1, \ldots, z_N) &= z_1 \\ \pi_2(z_1, \ldots, z_N) &= z_2. \end{aligned} \tag{2}$$

We define the *critical points* of a solution component of an algebraic system with respect to a projection as follows.

**Definition 2.1 (Critical point:)** *Let $Z$ be a solution component in $f^{-1}(0)$. A* critical point *of $Z$ with respect to projection $p$ is any point $z^* \in Z$ such that the null space of the Jacobian matrix of $f$ evaluated at $z^*$ includes a vector $v$ that $p$ sends to 0.*

If $Z \subset \mathbb{C}^N$ is $k$-dimensional and projection $p$ is a map $p : \mathbb{C}^N \to \mathbb{C}^k$, then a smooth point $z^*$ of $Z$ is a critical point with respect to $p$ if d$p$ drops rank at $z^*$. Singular points of $Z$ are also critical points.

Using the projections in (2), we have the following lemma.

**Lemma 2.2** *Let $Z \subset \mathbb{C}^N$ be a two-dimensional component of the solution set of $f$ as in (1), let $z^*$ be a point of $Z$, and let $C$ be the component in $Z \cap \pi_1^{-1}(\pi_1(z^*))$ that contains $z^*$. Assume that $Z$ is in general enough position so that $C$ is one dimensional. If $z^*$ is a critical point of $C$ with respect to $\pi_2$, then $z^*$ is also a critical point of $Z$ with respect to $\pi$.*

**Proof.** By assumption, $C$ is a component of $g^{-1}(0)$, where $g = \{f(z), \pi_1(z) - \pi_1(z^*)\}$. Let $Jf$ and $Jg$ be the Jacobian matrices of $f$ and $g$, resp. Recognizing that

$$Jg = \begin{bmatrix} & Jf & \\ 1 & 0 \cdots & 0 \end{bmatrix},$$

the proof is obvious. □

## 2.3 The one-dimensional case

Let $Z$ be a one-dimensional complex irreducible component of the solution set of the polynomial system (1) with each $f_i$ having real coefficients. For this article the case when $Z$ is of multiplicity one suffices. If $Z$ is not self-conjugate, then the set of real points in $Z$ is finite and can be found via intersection with its complex conjugate. Hence, we may assume that $Z$ is self-conjugate. Let $Z_{\mathbb{R}} = Z \cap \mathbb{R}^N$ be the set of real points of $Z$.

We follow [8] with one minor change: we work with an arbitrary real linear projection $p : \mathbb{C}^N \to \mathbb{C}$ instead of a general linear projection. This change means that there might be more than one critical point of the map $p_{Z_{\mathbb{R},1}} : Z_{\mathbb{R},1} \to \mathbb{C}$ over the same image point in $\mathbb{C}$. Theoretically this changes very little about the algorithm. We are forced to make this change to carry out the cell decomposition of a surface, because a random change of coordinates will not guarantee that every curve we deal with is in general position.

The not-necessarily general linear projection has several consequences. First, it is possible that the whole curve could lie in a hyperplane that projects to the same image point in $\mathbb{C}$. This case can be detected inexpensively by sampling enough general points on the curve [11]. Moreover, because of the generic change of coordinates made at the start of the algorithm in §3, this will not happen for the curves that arise in our method for the cell decomposition of a surface.

Second, with an arbitrary linear projection, an inflection point of the curve might be a critical point, with its tangent line projecting to a point. Under a general perturbation of the projection, this critical point would break up into several points (possibly complex). Thus, we see that the inflection point is a singular solution of the equations that model criticality. This is not really a problem, since our numerical methods handle singularities well using singular endgames, but computing such points is numerically more challenging than computing simple critical points.

The set of real points $Z_{\mathbb{R}} = Z \cap \mathbb{R}^N$ of $Z$ breaks up into a finite number of connected components such that

1. a finite set $Z_{\mathbb{R},0} \subset Z_{\mathbb{R}}$ of the components are isolated singular points of $Z$; and
2. a finite set $Z_{\mathbb{R},1} \subset Z_{\mathbb{R}}$ of the components are one-dimensional sets, which are smooth outside the singular set of $Z$.

The goal is to break $Z_{\mathbb{R},1}$ into 1-cells. For us, 1-cells are subsets of $Z_{\mathbb{R},1}$ that are images of the unit interval $[0,1]$ into $Z_{\mathbb{R},1}$ under some homeomorphism that is a diffeomorphism on $(0,1)$. Such a set is specified uniquely by the image of 0, the image of 1, and a random point in the interval. We represent a 1-cell of $Z_{\mathbb{R},1}$ by the data structure of an edge which is a quadruple $(w, c_0, c_1, p)$ such that

1. $p : \mathbb{C}^N \to \mathbb{C}$ is a real linear projection from $\mathbb{R}^N$ to $\mathbb{R}$;
2. $c_0, c_1$ are the boundary points of the 1-cell;
3. $w$ is a point in the 1-cell minus its boundary points; and
4. $p$ maps the 1-cell one-to-one onto its image and is a diffeomorphism on the 1-cell minus the endpoints.

By a degenerate edge, we mean a quadruple $(w, c_0, c_1, p)$ with $w = c_0 = c_1$, which represents a point.

As discussed briefly above, we may assume that

$$p(Z_{\mathbb{R},1}) \quad \textit{is not a point.} \tag{3}$$

Let the projection $p$ be given as $a \cdot z$, where $a \in \mathbb{R}^{1\times N}$ is a row matrix.

The algorithm for computing a cell decomposition of $Z_{\mathbb{R}}$ takes an input argument *merge*, which is specified as either "true" or "false." The role of this input argument will become clear in step 7 of the algorithm below. Later, the merge option is used in the 2-dimensional algorithm to simplify certain cross-sectional curves of a real surface.

1. **Find critical points:** Find the set of critical points $\mathcal{B}$ on $Z_{\mathbb{R}}$ with respect to $p$, that is, find the solutions in $Z$ of the system

$$f_{crit} = \begin{bmatrix} f(z) \\ Jf(z) \cdot v \\ a \cdot v \end{bmatrix} = 0, \qquad z \in \mathbb{C}^N, v \in \mathbb{P}^{N-1}, \tag{4}$$

where $Z$ is in $V(f)$, and $Jf$ is the Jacobian matrix of $f$. As discussed in [8], this must be done with a method that finds solutions at all dimensions, since singular points of $Z$ may have higher-dimensional tangent spaces satisfying (4). Let $\bar{p} : (\mathbb{C}^N \times \mathbb{P}^{N-1}) \to \mathbb{C}^N$ be the natural projection $(z, v) \mapsto (z)$. By the assumption that $Z$ is a multiplicity one curve, for any solution component $V$ of $Z \cap V(f_{crit})$, $\bar{p}(V)$ must consist of isolated points.

Let $\mathcal{B}$ be the set of real solution points in $\bar{p}(Z \cap V(f_{crit}))$.

2. **Add bounding points**

   (a) Let $\hat{T} = \pi_1(\mathcal{B}) = \{t_1, \ldots, t_m\}$ sorted such that $t_1 < t_2 < \cdots < t_{m-1} < t_m$.

   (b) If $\mathcal{B} \neq \emptyset$

      i. Choose $t_0 < t_1$ and $t_{m+1} > t_m$.

      ii. Let $T = \hat{T} \cup \{t_0, t_{m+1}\}$.

   Otherwise, set $T = \{t_0, t_1\}$ for any $t_0 < t_1$.

3. **Cut at $t_i$:**

   (a) Let
   $$\mathcal{E} = \{p^{-1}(t_i) \cap Z_{\mathbb{R}}, i = 0, \ldots, m+1\} \setminus \mathcal{B}.$$
   Solve to find $\mathcal{E}$ by a homotopy moving the linears of the witness set of $Z$ (see Section 2.1).

   (b) For $j = \{0, m+1\}$, if $p^{-1}(t_j) \cap \ Z_{\mathbb{R}} = \emptyset$, delete $t_j$ from $T$.

   (c) Set $m := \#(T) - 2$, and renumber the elements of $T$ as $t_0, \ldots, t_{m+1}$.

   (d) If *merge* is false, append $\mathcal{E}$ to $\mathcal{B}$ and set $\mathcal{E} = \emptyset$.

4. **Find generic points:** For $i = 0, \ldots, m$, let $t_i^*$ be a generic number in $(t_i, t_{i+1})$ and solve to find $W_i^* = p^{-1}(t_i^*) \cap Z_{\mathbb{R},1}$, again by a homotopy on the witness set of $Z$.

5. **Find end points:** For $i = 0, \ldots, m$ and each $w \in W_i^*$,

   (a) track $w$ to the left from $t = t_i^*$ to $t = t_i$ to find a point $c_{i,w} \in (\mathcal{B} \cup \mathcal{E})$, and

   (b) track $w$ to the right from $t = t_i^*$ to $t = t_{i+1}$ to find a point $c_{i+1,w} \in (\mathcal{B} \cup \mathcal{E})$.

   The homotopy for these steps is $\{f(z), \pi(z) - t\} = 0$.

6. **Construct edges:** For $i = 0, 1, \ldots, m$ and each $w \in W_i^*$, create an edge $\gamma = (w, c_{i,w}, c_{i+1,w}, p)$.

7. **Merge:** If *merge* is "true," then replace any two edges that meet at a point in $\mathcal{E}$ with a single edge. That is, if edge $(w_1, c_{i,w_1}, e, p)$ meets edge $(w_2, e, c_{i+1,w_2}, p)$ at $e \in \mathcal{E}$, these are merged to form a single edge $(e, c_{i,w_1}, c_{i+1,w_2}, p)$.

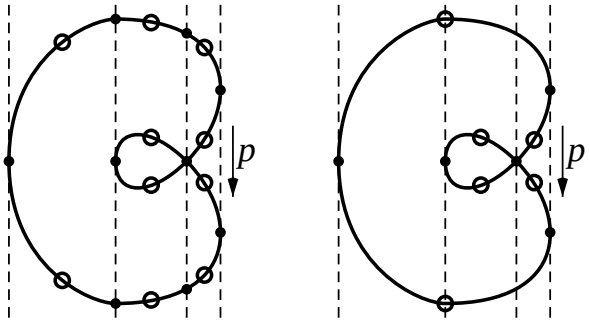


Figure 1: A decomposition of a curve before and after merging edges that meet outside of $\mathcal{B}$. Filled dots are edge endpoints, the centers of the open circles mark general points on the edges.

8. **Detect isolated points:** $Z_{\mathbb{R},0}$ is the subset of points in $\mathcal{B}$ that have no incident edges.

The data structure $D_1(Z, p)$ representing $Z$ consists of the finite list of edges resulting from the algorithm above and the system $f(z) = 0$. The edges will often be referred to as 1-cells. Figure 1 illustrates the decomposition of a curve into 1-cells.

We added bounding points in Step 2 to cut off any arcs of the curve that go to infinity on the left or right. If the curve is bounded in either direction, then this is discovered in Step 3, and the corresponding bounding point is removed from $T$. If the curve is unbounded, then one can continue the bounding point as far as one wishes towards infinity: there are no more critical points to be encountered.

A line has no critical points, in which case $\mathcal{B}$ at Step 1 is empty. Then, at Step 2, we set $T = \{t_0, t_1\}$ with any $t_0 < t_1$. The result of the decomposition algorithm in such a case is a single line segment.

For a union of one-dimensional complex irreducible components, we compute the data structure for each irreducible component.

## 3 Two dimensional case

Let $Z$ be a two-dimensional complex irreducible component of the solution set of the system $f(z) = 0$ in (1). We make two assumptions about $Z$.

1. $Z$ is self-conjugate, that is, if $z$ is in $Z$ then the complex conjugate of $z$, $z'$ is also in $Z$. If not, then $Z$ must have a corresponding conjugate component, say $Z'$, and the real points in $Z$ are found in $Z \cap Z'$, which will be at most a 1-dimensional component.

2. $Z$ is almost smooth in the sense that the intersection of $Z$ with the singular set of $f^{-1}(0)$, i.e., the subset of $Z$ where the rank of the Jacobian is less than $N - 2$, is finite.

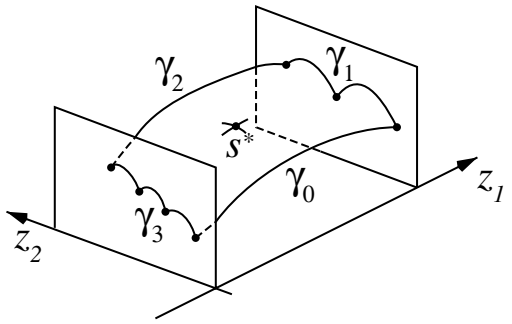


Figure 2: A two cell: generic point $s^*$; simple edges $\gamma_0$ and $\gamma_2$; and compound edges $\gamma_1$ and $\gamma_3$.

**Remark 3.1** The assumptions are easily checked. To check self-conjugacy, take a witness point $w$ of $Z$ and check if the conjugate of $w$ lies on $Z$. The almost smoothness condition is checked by solving a system that expresses the condition that the null space of the Jacobian on the surface be 3 or greater. Following [3], and letting $Jf$ be the $\nu \times N$ Jacobian matrix of $f$, the system to solve is

$$\begin{bmatrix} f(z) \\ Jf(z) \cdot B \cdot \begin{bmatrix} I_3 \\ \Xi \end{bmatrix} \end{bmatrix} = 0, \tag{5}$$

where $B$ is a general (random) $N \times N$ matrix, $I_3$ is the $3 \times 3$ identity matrix, and $\Xi$ is a $(N-3) \times 3$ matrix of unknowns. If this system has no positive-dimensional components, then the almost smoothness condition holds.

**Remark 3.2** The algorithm below for decomposing a surface will still work properly for some exceptions to the almost smoothness condition. For example, any singularity sets of any dimension that do not meet the real surface $Z_{\mathbb{R}}$ in more than a finite number of points will not cause trouble. However, checking almost smoothness over $\mathbb{R}$ is more difficult than checking it over $\mathbb{C}$. We leave it for future work to remove the almost smoothness condition entirely.

When $Z$ is self-conjugate and almost smooth, as we assume, the set of real points $Z_{\mathbb{R}}$ of $Z$ is a union of a finite set of isolated points $Z_{\mathbb{R},0}$ (all of which are in the singular locus of the reduction of $Z$) with a two dimensional set $Z_{\mathbb{R},2}$ whose intersection with the smooth points of $Z$ is smooth.

Our goal is to break $Z_{\mathbb{R},2}$ into 2-cells and to list the points in $Z_{\mathbb{R},0}$. A 2-cell is the image under a homeomorphism of a closed disk in $\mathbb{R}^2$ under a map that is diffeomorphic on the open disk. As illustrated in Figure 2, the data structure representing a 2-cell is a quadruple $(s^*, \gamma_0, \gamma_1, \gamma_2, \gamma_3, \pi_1, \pi_2)$ such that

1. $s^*$ is a general point in the interior of the 2-cell;

2. $\pi, \pi_1, \pi_2$ are linear projections as in (2);

3. $\gamma_0$ and $\gamma_2$ are 1-cells, possibly degenerate, as described in Section 2.3;

4. $\gamma_1$ and $\gamma_3$ are each a collection of 1-cells, possibly degenerate, connected end-to-end with $\pi_1(\gamma_1)$ and $\pi_1(\gamma_3)$ each a single point distinct from the other, and the 1-cells inside each progressing monotonically with respect to $\pi_2$;

5. for $i = 0, 1, 2, 3$, one endpoint of $\gamma_i$ is an endpoint of $\gamma_{i-1}$ and the other is an endpoint of $\gamma_{i+1}$ (using modulo 4 arithmetic).

It may help to think of $\gamma_0$ and $\gamma_2$ as the "up" and "down" edges, and $\gamma_1$ and $\gamma_3$ as "right" and "left" compound edges.

Below in Item 1, we make a random linear change of coordinates and then use $\pi, \pi_1, \pi_2$. Instead of making a random change of coordinates, we could have chosen $\pi$ as a random linear projection with $\pi_1, \pi_2$ chosen as compositions of $\pi$ with generic maps of $\mathbb{C}^2$ to $\mathbb{C}$. The initial random change of coordinates is easier to work with and as far as numerical computations go, e.g., within Bertini [4] (the software we use to find the witness sets), the computational differences are slight. Indeed, evaluations are done using straight-line programs and thus changes of coordinates are just an evaluation step before evaluating polynomials.

Our data structure $D_2(Z, \pi)$ for the real surface is the system of equations $f(z) = 0$, a rotation matrix $A$, and a union of 2-cells $(s^*, \gamma_1, \gamma_2, \gamma_3, \gamma_4, \pi, \pi_1, \pi_2)$ such that every point is either in the interior of exactly one of the 2-cells or is in the interior of exactly one edge, or is a vertex. The interior of each edge is in the boundary of exactly two 2-cells.

The following is an algorithm to compute the proposed data structure for the real points in $Z$. We begin with a witness set $\{f, L, W\}$ for $Z$.

As we step through the algorithm, we illustrate it on the example of a surface in $\mathbb{R}^3$ defined by the equation

$$((x + 0.35)^2(1 - x^2) - y^2)^2 - z^2 - 0.00531441 = 0. \tag{6}$$

This is a compact surface with one hole and a pinch-point singularity at $(x, y, z) = (-0.8, 0, 0)$. Because of its special form, we can solve for $z$ in terms of $(x, y)$. Sampling $(x, y)$ on a fine grid in the box $[-1.1, 1.1] \times [-1.1, 1.1]$, one can make a mesh plot of the surface in Matlab as appears in Figure 3. The mesh is two disconnected pieces, jagged around the boundary, because this crude approach does not solve for the boundary curve.

The algorithm for the two-dimensional case is as follows.

1. **Change coordinates:** Choose a random $A \in SO(\mathbb{R}, N + 1)$, and let $F(z) = f(Az)$. Identify $Z \subset f^{-1}(0)$ with the corresponding component in $F^{-1}(0)$ and

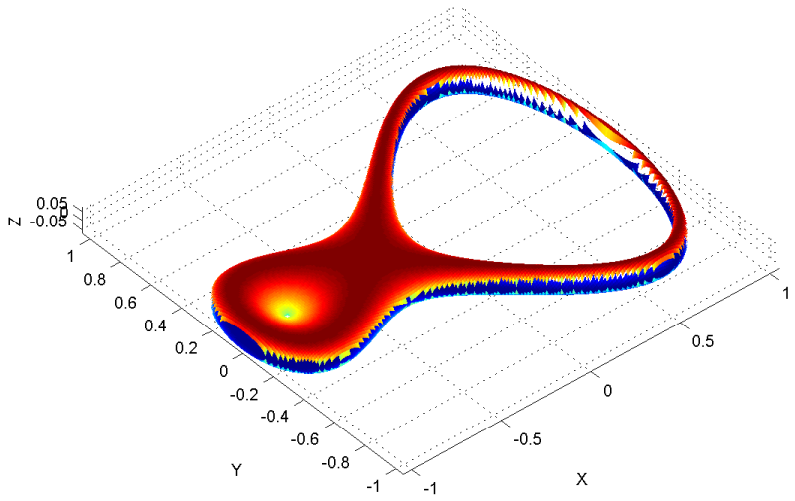


Figure 3: The running example drawn by meshing a grid of samples.

use $A$ to transform the witness set $\{f, L, W\}$ into a witness set for the rotated component. By abuse of notation, we rename the rotated $Z$ and the function $F(z)$ as $Z$ and $f$ from here on.

Figure 4 illustrates the example surface as a rotated point cloud, which we will decompose in the remaining steps.

2. **Choose projections:** Let $\pi, \pi_1, \pi_2$ be as given in (2).

3. **Find the critical set:** Let $Jf$ be the Jacobian matrix for $f$, and let $Jf_{3:N}$ be the submatrix obtained by deleting the first two columns. Starting with the witness set for $Z$, find a witness set for the curve $Crit = Z \cap V(G(z, v))$, where

$$G(z, v) = \begin{bmatrix} f(z) \\ Jf_{3:N}(z) \cdot v \end{bmatrix} = 0, \tag{7}$$

where $v \in \mathbb{P}^{N-3}$. Use this witness set in all calculations on the curve in the following steps. Since the curve, $Crit$, described by (7) lives in $\mathbb{C}^N \times \mathbb{P}^{N-3}$, we introduce the projection

$$\bar{\pi}_1(z, v) \mapsto \pi_1(z) \mapsto z_1.$$

By abuse of notation, we will still refer to this as simply $\pi_1$. Similarly, $\pi_2$ is reinterpreted as the map $(z, v) \mapsto z_2$.

4. **Add up and down boundaries:** Decompose $Crit$ with respect to $\pi_2$ to get $D_1(Crit, \pi_2)$. If $Crit$ is found to be unbounded in either direction, cut it off by adding the appropriate planar slice of $Z$ as follows. Suppose $Crit$ is unbounded

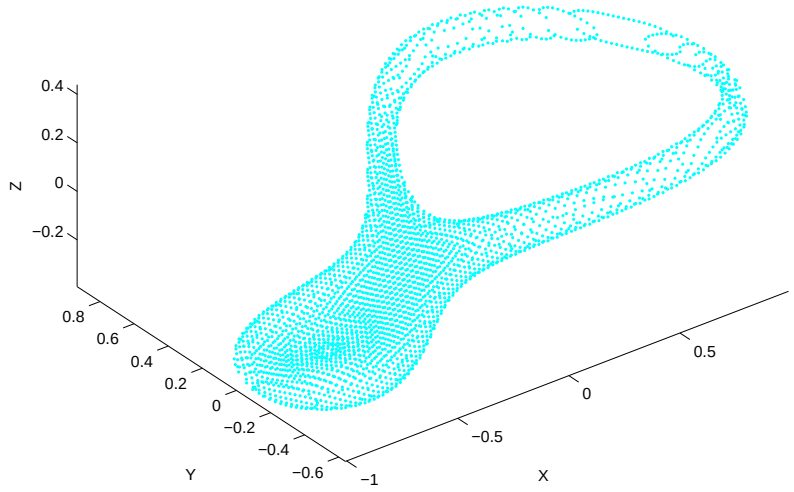


Figure 4: Point cloud on the surface after general rotation (Step 1).

below with respect to $\pi_2$ and that $t_0$ is the lower boundary inserted in Step 2 of the curve algorithm to cut it off. Then add the curve $Z_{\mathbb{R}} \cap \pi_2^{-1}(t_0)$ to $Crit$. Add a similar curve to form the upper boundary, if necessary.

5. **Decompose the critical set:**

   (a) Use the curve algorithm to compute $C = D_1(Crit, \pi_1)$ with option *merge* = false.

   (b) Let $T$ be as developed in the curve algorithm.

6. **Slice $Z_{\mathbb{R}}$ vertically at critical values:** Let the elements of $T$ be $t_0, \ldots, t_{m+1}$. Find the curve decompositions $K_i = D_1(\pi_1^{-1}(t_i), \pi_2)$ for $i = 0, \ldots, m+1$.

7. **Slice $Z_{\mathbb{R}}$ vertically at generic values:** For $i = 0, \ldots, m$, let $t_i^*$ be the same point in $(t_i, t_{i+1})$ that appears in the decomposition $D_1(Crit, \pi_1)$ from Step 5. Compute the curve decompositions $K_i^* = D_1(\pi_1^{-1}(t_i^*), \pi_2)$ with option *merge* = true.

   The generic point, $s_{i,j}$, of the $j$th edge in $K_i^*$ will be the generic point of a 2-cell. Initialize the surface decomposition as a list of incomplete 2-cells of the form

$$\{s_{i,j}, [\,], [\,], [\,], [\,], \pi, \pi_1, \pi_2\},$$

   where each "[ ]" is a placeholder for the edges to be filled in subsequently.

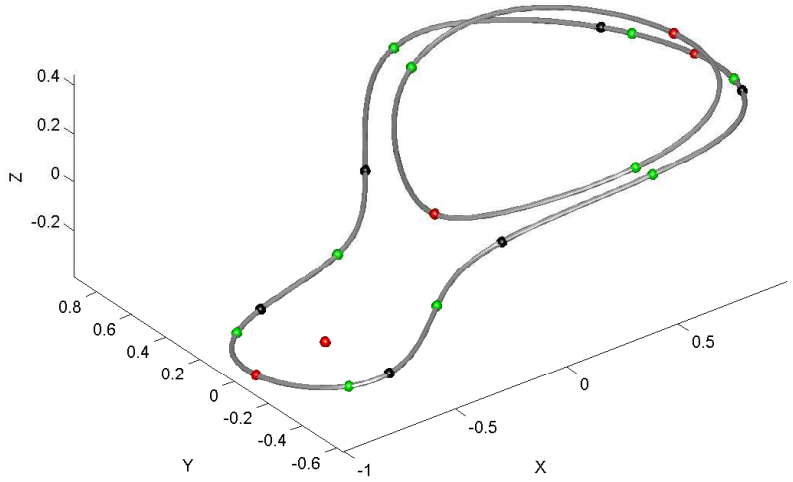


Figure 5: Decomposition of the critical curve at Step 5. Red dots are critical points, black dots are endpoints from slicing at the critical points, green dots are the generic points of the 1-cells. There are eleven 1-cells, one of which, at the pinch point, is degenerate.

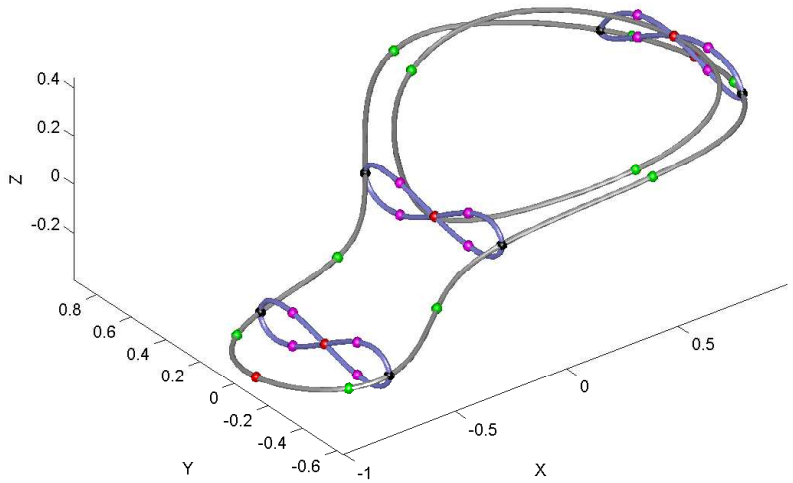


Figure 6: Slice $Z_{\mathbb{R}}$ vertically at critical values, Step 6. There are five slices: three look like figure eights, two on the far ends are just single points. Magenta balls mark the generic points of the 1-cells of the decompositions $K_i$ of these curves.

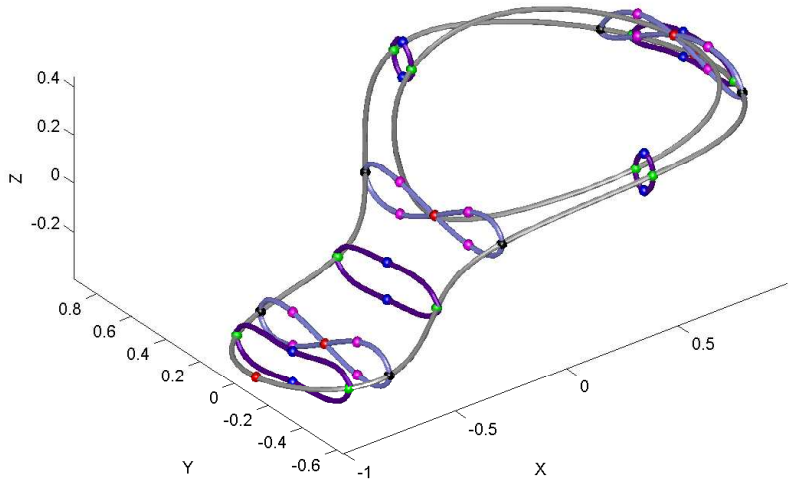


Figure 7: Slice $Z_{\mathbb{R}}$ vertically at generic values, Step 7. Blue points are the generic points of the curve decompositions, $K_i^*$. These points become the generic points of the 2-cells. The critical points for these curves coincide with the green generic points of the decomposition of the critical curve, see Step 8.

8. **Find upper and lower edges:** Since we constructed $K_i^*$ with *merge* = true, Lemma 2.2 implies that the end points of each bounded edge in $K_i^*$ must lie on the critical curve while an unbounded edge ends at the up or down boundaries established in Step 4, both now included in $Crit$. Because in Step 7 we sliced $Z_{\mathbb{R}}$ at the same values $t_i^*$ as were used in the decomposition of $Crit$ in Step 5, each end point of every edge in $K_i^*$ must be the same as a generic point in one of the edges of $Crit$, which thereby identifies the lower and upper edges of the 2-cell. These are the edges $\gamma_0$ and $\gamma_2$ for point $s_{i,j}$.

   After completing this step, each 2-cell in $S$ has the form

$$\{s_{i,j}, \gamma_0, [\,], \gamma_2, [\,], \pi, \pi_1, \pi_2\}.$$

9. **Find left and right edges:** Because we constructed the decomposition of $Crit$ in Step 5 with *merge* = false, the 1-cells in $Crit$ all start and stop at adjacent values in $T$. Moreover, by Lemma 2.2, the end points of these 1-cells must be part of the critical set of the vertical curves $K_i$. That is, for either end point of a 1-cell in $Crit$, at least one 1-cell of some $K_i$ has an end point that matches it. For a 2-cell with generic point $s_{i,j}$, edges $\gamma_0$ and $\gamma_2$ have a left end point that is a critical point of $K_i$ and a right end point that is a critical point of $K_{i+1}$.

   Consider the 1-cell in $K_i^*$ whose generic point is $s_{i,j}$ and whose end points lie on $\gamma_0$

and $\gamma_2$. The interior of the 1-cell is diffeomorphic to its own image under $\pi_2$. As $t$ varies in the interval $(t_i, t_i^*]$, there is a continuous deformation of this 1-cell lying in $\pi_1^{-1}(t)$ that maintains the diffeomorphism. Thus, as $t$ approaches $t_i$, the 1-cell must limit to a connected chain of 1-cells in $K_i$. Similarly, as $t$ approaches $t_{i+1}$, the 1-cell limits to a connected chain of 1-cells in $K_{i+1}$. These are the compound left and right edges $\gamma_3$ and $\gamma_1$, resp., of the 2-cell for $s_{i,j}$.

Let the left end points of $\gamma_0$ and $\gamma_2$ be $w_0$ and $w_2$, resp., and let $w_i^* = \gamma_i \cap \pi^{-1}(t_i^*)$, $i = 0, 2$. (That is, $w_i^*$ is the generic point of 1-cell $\gamma_i$.)

To find the left edge, search $K_i$ to find all sequences of edges that monotonically progress from $\pi_2(w_0)$ to $\pi_2(w_2)$. One of these sequences is the left edge. Let $q$ be the generic point of a 1-cell in one of these sequences. Then, that 1-cell is part of the left edge if and only if there is a continuous path $r(t) \in Z_{\mathbb{R}}$ from $s_{i,j}$ to $q$ such that $\pi_2(\gamma_0 \cap \pi_1^{-1}(t)) < \pi_2(r(t)) < \pi_2(\gamma_2 \cap \pi_1^{-1}(t))$. To test this, we form a homotopy as follows.

Let

$$\rho_{start} = \frac{\pi_2(s_{i,j}) - \pi_2(w_0^*)}{\pi_2(w_2^*) - \pi_2(w_0^*)} \qquad \text{and} \qquad \rho_{end} = \frac{\pi_2(q) - \pi_2(w_0)}{\pi_2(w_2) - \pi_2(w_0)},$$

then define

$$\rho(t) = \frac{(t_i - t)\rho_{start} + (t - t_i^*)\rho_{end}}{t_i - t_i^*}.$$

By this definition, $\rho(t) \in (0, 1)$ for $t \in [t_i, t_i^*]$. The homotopy function is:

$$H(r, u_0, v_0, u_2, v_2, t) = \begin{bmatrix} G(u_0, v_0) \\ \pi_1(u_0) - t \\ G(u_2, v_2) \\ \pi_1(u_2) - t \\ f(r) \\ \pi_1(r) - t \\ (1 - \rho(t))\pi_2(u_0) + \rho(t)\pi_2(u_2) - \pi_2(r) \end{bmatrix} = 0, \qquad (8)$$

where function $G$ is defined in (7). For $k = 0, 2$, let $v_k^*$ be the unique solution for $v$ of $G(w_k^*, v) = 0$. We follow the path of $H(r, u_0, v_0, u_2, v_2, t) = 0$ from

$$(s_{i,j}, w_0^*, v_0^*, w_2^*, v_2^*)$$

at $t = t_i^*$ to $t = t_i$. If and only if this path ends at $q$, the 1-cell with $q$ as its generic point is part of the left edge $\gamma_3$ for the 2-cell identified with $s_{i,j}$. A similar homotopy tests which 1-cells belong to the right edge $\gamma_1$.

The logic is that we are sliding the center point $r(t)$ along the surface $Z_{\mathbb{R}}$ in the same $z_1$-plane as $u_0$ and $u_2$, with $u_0$ and $u_2$ moving along the arcs $\gamma_0$ and $\gamma_2$ of the critical curve. As we slide, $\rho(t)$ determines the fraction of the way $\pi_2(r(t))$

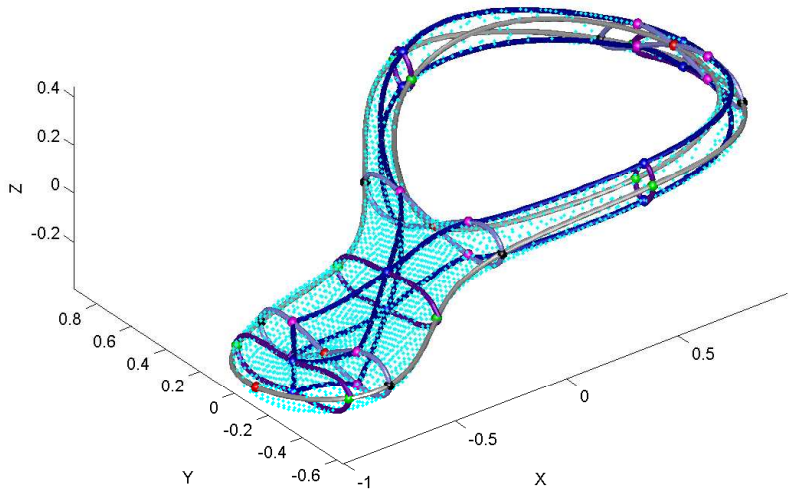


Figure 8: Find the left and right edges by connecting the generic point (blue) of each 2-cell to generic points (magenta) of the neighboring curves $K_i$ and $K_{i+1}$ (Step 9).

lies between $\pi_2(u_0)$ and $\pi_2(u_1)$ on a schedule that puts it at $\pi_2(s_{i,j})$ at the start and arrives at $\pi_2(q)$ at the end.

Because of the way we choose generic points in the curve decomposition (see Figure 1), the generic points of several prospective 1-cells often project to the same $\pi_2(q)$, so the homotopy tests all of these at once.

Since the left and right compound edges are unique, we may stop testing the prospective 1-cells as soon as a complete compound edge has been assembled.

Degenerate 1-cells on the left or right do not have to be tested. Since these have no interior, each one's membership in the edges $\gamma_1$ or $\gamma_3$ is determined by its connectivity to the end points of other 1-cells. In particular, it may happen that $\gamma_0$ and $\gamma_2$ meet at a critical point which therefore forms the entire left or right edge. For example, in Figure 8, the two 2-cells near $x = -1$ have a degenerate left edge consisting of the critical point (red). This edge does not have to be tested, since we already know the up and down edges meet there.

10. **Assemble all two-cells:** The data $(s^*_{ij}, \gamma_0, \gamma_1, \gamma_2, \gamma_3, \pi, \pi_1, \pi_2)$ defines a two-cell. The data structure $D_2(Z, \pi, \pi_1, \pi_2)$ consists of the finite list of two-cells resulting from the algorithm above.

    In the running example, as shown in Figure 8, there are ten 2-cells. At this point, the topology of the surface is fully revealed by the way the 2-cells glue together across their 1-cell boundaries.

11. **Find isolated points $Z_{\mathbb{R},0}$:** The isolated points are the points in $\mathcal{B}$ that are not part of any edge used in the decomposition of $Z_{\mathbb{R},1}$ into 2-cells.

**Remark 3.3 Sampling a 2-cell.** A variant of the homotopy (8) can be used to sample the 2-cell. Just remove the dependence of $\rho$ on $t$ and let it be a new variable. With that change, one can track any path in the open box $(\rho, t) \in (0,1) \times (t_i, t_{i+1})$ to collect values of $r$ that lie in the interior of the cell. Points on the boundary of the cell can be sampled using the curve decompositions of the edges, which are already in our data set.

# 4 Discussion and Conclusions

We have described an algorithm to decompose the real surface inside an irreducible and reduced two-dimensional complex algebraic set. The decomposition consists of 2-cells given by a generic point in the interior and edges given by 1-cells or a chain of connected 1-cells. The cells are established by first placing the surface in general position with a random rotation, and then using projections on the first two coordinate directions.

Although the algorithm has been illustrated on a surface in $\mathbb{R}^3$, the technique also applies to surfaces in any higher dimensional space. The algorithm also applies to a non-compact surface by inserting bounding hyperplanes and computing the arcs where the surface meets these planes. These planes are placed outside the limits of the critical curve of the surface, so none of the finite topology of the surface (holes) is missed. If the surface is compact, the algorithm returns a complete cell decomposition that could be used to compute the geometric genus of the surface.

Although we have not done so here, one could consider finding a cell decomposition of a compactification of a noncompact surface by homogenizing the polynomials defining the surface and casting the problem in real projective space. The decomposition could follow a similar approach to the algorithm given here, but would need to be computed using several patches of projective space with allowance for properly gluing the pieces together.

We have restricted the algorithm to complex algebraic sets that have at most a finite number of singularities, and we provide a test that verifies whether a given surface satisfies this assumption. The algorithm as written already works for a somewhat wider class of algebraic sets. In particular, it is enough for there to be a finite number of real singular points, but we do not provide a test to verify that condition. Rather, we leave it to future work to remove entirely the almost smoothness condition, which we believe is feasible.

# References

[1] A. Andreotti and T. Frankel, *The Lefschetz theorem on hyperplane sections*, Ann. of Math. (2), 69 (1959), pp. 713–717.

[2] S. Basu, R. Pollack, and M.-F. Roy, *Algorithms in real algebraic geometry*, vol. 10 of Algorithms and Computation in Mathematics, Springer-Verlag, Berlin, second ed., 2006.

[3] D. Bates, J. Hauenstein, C. Peterson, and A. Sommese, *Numerical decomposition of the rank-deficiency set of a matrix of multivariate polynomials*, in Approximate Commutative Algebra, vol. 14 of Texts and Monographs in Symbolic Computation, Springer, 2010, pp. 55–77.

[4] D. Bates, J. Hauenstein, A. Sommese, and C. Wampler, *Bertini: Software for numerical algebraic geometry*. Available at `www.nd.edu/~sommese/bertini`.

[5] D. Bates, C. Peterson, A. Sommese, and C. Wampler, *Numerical computation of the genus of an irreducible curve within an algebraic set*, Journal of Pure and Applied Algebra, 215 (2011), pp. 1844–1851.

[6] M. Goresky and R. MacPherson, *Stratified Morse theory*, vol. 14 of Ergebnisse der Mathematik und ihrer Grenzgebiete (3) [Results in Mathematics and Related Areas (3)], Springer-Verlag, Berlin, 1988.

[7] J. Hauenstein and C. Wampler, *Solving structured polynomial systems using regeneration*, (2011). in preparation.

[8] Y. Lu, D. Bates, A. Sommese, and C. Wampler, *Finding all real points of a complex curve*, in Proceedings of the Midwest Algebra, Geometry and Its Interactions Conference, vol. Contemporary Mathematics 448, AMS, 2007, pp. 183–205.

[9] J. Milnor, *Morse theory*, Based on lecture notes by M. Spivak and R. Wells. Annals of Mathematics Studies, No. 51, Princeton University Press, Princeton, N.J., 1963.

[10] A. Sommese, J. Verschelde, and C. Wampler, *Numerical decomposition of the solution sets of polynomial systems into irreducible components*, SIAM J. Numer. Anal., 38 (2001), pp. 2022–2046.

[11] A. Sommese, J. Verschelde, and C. Wampler, *Numerical irreducible decomposition using projections from points on the components*, in Symbolic computation: solving equations in algebra, geometry, and engineering (South Hadley, MA, 2000), vol. 286 of Contemp. Math., Amer. Math. Soc., Providence, RI, 2001, pp. 37–51.

[12] A. Sommese, J. Verschelde, and C. Wampler, *Symmetric functions applied to decomposing solution sets of polynomial systems*, SIAM J. Numer. Anal., 40 (2002), pp. 2026–2046.

[13] ——, *Homotopies for intersecting solution components of polynomial systems*, SIAM J. Numer. Anal., 42 (2004), pp. 1552–1571.

[14] A. Sommese and C. Wampler, *The Numerical Solution of Systems of Polynomials Arising in Engineering and Science*, World Scientific, Singapore, 2005.

[15] C. Wampler and A. Sommese, *Numerical algebraic geometry and algebraic kinematics*, Acta Numerica, 20 (2011), pp. 469–567.