Cell decomposition of almost smooth real algebraic surfaces

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

April 16, 2012

Abstract

Let Z be a two dimensional irreducible complex component of the solution set of a system of polynomial equations with real coefficients in N complex variables. This work presents a new numerical algorithm, based on homotopy continuation methods, to decompose into 2-cells any almost smooth real algebraic surface contained in Z. Each 2-cell (a face) has a generic interior point and a boundary consisting of 1-cells (edges). Similarly, the 1-cells 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, defining an invertible map from the parameter space of the homotopy to the cell. This work draws on previous results for the curve case. 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.

1 Introduction

Let

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

*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 grants DMS-0915211 and DMS-1114336.

§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.

be a polynomial system with each f_i having real coefficients. Let $V(f)$ denote the underlying solution set

$$V(f) := \{x \in \mathbb{C}^N \mid f(x) = 0\},$$

which is an algebraic set, and let $f^{-1}(0)$ denote the set $V(f)$ with its full possibly nonreduced structure.

In this article we give a numerical algorithm to decompose the two-dimensional set of real points, $Z_{\mathbb{R}} = Z \cap \mathbb{R}^N$, of a two-dimensional irreducible complex component $Z \subset V(f)$ into cells 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.

Note that if Z is not self-conjugate, then there must exist another irreducible component conjugate to it, and the real points of Z, which must lie in the intersection of the two components, can be a set of dimension at most one. Hence, the first assumption is not a restriction but rather is a condition for Z to contain a real surface. However, the second assumption is restrictive: in effect we are assuming that Z as an irreducible component of $f^{-1}(0)$ is smooth except on a finite set. In particular, this means that Z meets other components of $V(f)$ in at most a finite set. In this article, we call any Z that meets the second criterion "almost smooth."

The almost smoothness condition eliminates several possibilities that would otherwise introduce complexity in our algorithm. First, if Z were a set of multiplicity greater than one, any path cut out on Z would be singular and would therefore not be conducive to numerical path tracking. This can be overcome by deflation procedures [22, 41], but we do not wish to introduce that complication into this article. Similarly, if Z were to contain a singular curve within an otherwise nonsingular real surface, that curve would define some of the 1-cell boundaries to 2-cells produced by our algorithm. If that curve is not of multiplicity one, numerically tracking along it requires deflation. Also problematic are examples with components of different dimension, e.g. the Whitney umbrella, which consists of a two dimensional real surface and a one dimensional "handle." Part of the handle extends free of the surface and is not part of any 2-cell in a decomposition of the surface. Again, such curves could be tracked using deflation.

While we see no fundamental barrier to adding procedures that would remove the almost smooth restriction, for the sake of simplicity we choose to impose this condition in this first presentation of our approach. The self-conjugacy and almost smoothness conditions can be checked at the outset: see Remark 6.1 below.

The data structure we introduce is a main contribution of this article. Let $S = \mathbb{R}^N \cap Z$ be the set of real points of Z, i.e., S is the real surface we wish to decompose. The decomposition leads to a finite list of 2-cells, a finite list of boundary 1-cells, and a finite list of end points. The boundary of each 2-cell is a subset of the list of 1-cells, and the endpoints of each 1-cell are in the list of points. Each 1-cell and each 2-cell has a generic interior point and an associated homotopy for moving the generic interior point

to any other point in the interior of the cell. The homotopy for each n-cell ($n = 1$ or 2) has n parameters and implicitly defines an invertible mapping from the interior of a patch of parameter space to the interior of the cell. If S is compact, a decomposition in $\mathbb{R}^N$ covers all of S: each point of S is a member of the interior of exactly one 2-cell or the interior of exactly one 1-cell or is in the list of end points. If S is not compact, we compactify it by homogenizing f so that Z becomes an irreducible component in $\mathbb{P}^N$ and $S = \mathbb{P}^N_{\mathbb{R}} \cap Z$. By using multiple patches on $\mathbb{P}^N_{\mathbb{R}}$, we again produce a cell decomposition that covers all of S.

The cell decomposition shares characteristics of the witness-set data structure that is a mainstay of numerical algebraic geometry, and similarly allows straightforward numerical manipulation of the real surface. The cell structure is specified using representative points and projections. Once computed, the data structure may be used to:

1. generate as many points as desired spread out over a given cell or group of cells;
2. subdivide the cells into smaller cells;
3. decide if a real point satisfying the original polynomial system lies on the surface being decomposed; and
4. if the point does lie on the surface, the data-structure facilitates a decision of which cell or boundary curve the point lies on.

A cell decomposition can also be used in several ways to analyze the surface. Chief among these is visualization. Implicit curves and surfaces are in general difficult to graph. The 2-cells in our cell decomposition can be subdivided into smaller cells to provide a triangulation of the surface to any desired resolution. Using standard computer graphics, these triangles can be used to visualize projections of the surface into $\mathbb{R}^3$. Alternatively, the cells could be sampled as points and visualized as a point cloud.

A second use is for optimization. Approximate maxima of a general real continuous function on the surface could be found by examining its values on a fine enough subdivision of the surface.

A third use is in topological analysis of the surface. In particular, the list of cells and how they meet provides all the data needed to compute the homology of the real surface. Also, in [33], the technique has been put to use in designing mechanisms for generating one degree of freedom mechanical motion. Allowing a design parameter, such as a link length, to vary, the collection of all motion curves forms a surface. Then, the cell decomposition determines all the critical values of the design parameter where the topology of the motion curve changes. This allows an engineer to choose a design whose motion has the desired characteristics for the intended purpose of the machine.

Throughout this paper, we rely on the algorithms of numerical algebraic geometry that are implemented in the software package Bertini [8]. These methods are based on numerically tracking paths defined by polynomial homotopy functions, which can be constructed so that the end points of the paths expose all the irreducible solution

components of a polynomial system. These methods treat the solution set in complex space and require additional work to find the real solutions, if any, inside each complex component. An algorithm for the cell decomposition of the real points in a complex curve is given in [27]. This article extends that approach to the surface case. We give a summary of these prior results before using them to treat the case at hand.

1.1 Related work

The decomposition of the set of real solutions to polynomial systems has been studied extensively using many different approaches including cylindrical algebraic decompositions, subdivision, and isotopic methods. The cylindrical algebraic decomposition [14] provides a decomposition of the real space into connected sets based on the signs of the given polynomials. Such a decomposition naturally leads to algorithms for decomposing algebraic sets, and in particular, algebraic surfaces, e.g., [4, 11, 13, 16]. These methods rely upon computing critical points of projections, as does our method presented below. The symbolic-numeric decomposition method in [10] produces certifiable results without assumptions regarding the genericity of the projection.

Subdivision methods, e.g., [1, 26, 34], start with a bounded input domain and divide the domain based on a regularity criterion. Classically, the presence of singularities caused difficulties with topological regularity criteria. One approach to handle singularities with subdivision methods was presented in [1]. This approach, which is based on polar varieties, can be applied to a bounded part of a surface which has infinitely many singularities, such as the singular "handle" of the Whitney umbrella on a bounded domain.

Isotopic methods, e.g., [2, 12], compute simplicial complexes which are isotopic to the algebraic set. These methods, as does the method presented below, can yield topological information, such as the number of connected components, from a finite amount of data.

The significant new contribution of our approach is its reliance on exclusively numerical and local geometrical methods, without the need of any algebraic information (i.e. generators of the ideal) of the relevant components of the solution set. The resulting algorithms are therefore easily parallelized.

2 Background on Numerical Algebraic Geometry

The cell decomposition algorithm depends on concepts and sub-algorithms from numerical algebraic geometry, which we briefly review here. See [41, 43] for detailed expositions of the field.

> **Genericity.** If $A \subset \mathbb{C}^N$ is an irreducible algebraic set, we say that a property P holds *generically* on A if it holds on a dense, Zariski-open subset of A, that is, P is true on the complement in A of some proper algebraic subset of A. Thus, if we pick a point of A at random, P will be true for that point with *probability one*.

Almost all the algorithms in numerical algebraic geometry depend on the choice of one or more random quantities from some algebraic set, and the algorithm succeeds generically on that set. Such an algorithm is said to succeed with probability one.

Randomization. If Z is a k-dimensional irreducible component of $V(f)$ with f a system of ν equations as in (1), then for a generic matrix $M \in \mathbb{R}^{(N-k)\times\nu}$, Z is also an irreducible component of $M \cdot f$. By the simple version of Bertini's Theorem given as Theorem 13.5.1 in [41], the statement holds for generic complex $M \in \mathbb{C}^{(N-k)\times\nu}$. That it also holds for generic real M is a consequence of the fact that for any n, the intersection of $\mathbb{R}^n$ with a nonempty Zariski-open subset of $\mathbb{C}^n$ is also a nonempty Zariski-open set of $\mathbb{R}^n$. Because of this, we may assume throughout the remainder of this paper that f is a system of $N-2$ equations, for if $\nu > N-2$, we simply replace f by a smaller randomized version.

Note also that all probability one results still hold if M is replaced by $\Lambda \cdot M'$, where M' is generic in $\mathbb{R}^{(N-k)\times\nu}$ and Λ is any invertible matrix of consistent dimension. For example, M may be chosen in reduced row echelon form: $\begin{bmatrix} I & M'' \end{bmatrix}$, where I is the size $(N-k)$ identity matrix and M'' is random.

Witness sets. Let Z be an irreducible component of the solution set of the system $f(z) = 0$ given in (1). 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 polynomials and W is the intersection $Z \cap V(L)$. By genericity, $V(L)$ meets Z transversely in a finite set of $d = \deg Z$ smooth points $\{w_1, \ldots, w_d\}$ of Z. In addition to finding the degree and dimension of Z, some methods used to compute a witness set also determine the multiplicity of Z as a component of $f^{-1}(0)$.

The set W in $\{f, L, W\}$ is called a *witness point set* for Z. Witness sets are fundamental constructs for the numerical representation and manipulation of algebraic sets.

Numerical Irreducible Decomposition. A numerical irreducible decomposition consist of a collection of witness sets, one for each irreducible component of the solution set $V(f)$.

Here below we list the main available algorithms in numerical algebraic geometry that will be called upon by the cell decomposition. Each algorithm solves an essential problem in the field. Except as noted, all of these algorithms are provided in Bertini, a publicly available software package [8].

Problem 1: Path tracking	
Input	: Homotopy function $H(x,t) : \mathbb{C}^N \times \mathbb{C} \to \mathbb{C}^N$; a start point w such that $H(w,1) = 0$.
Output	: The endpoint $w' = \lim_{t\to 0} x(t)$, if it exists.

The term "homotopy function" implies that the Davidenko ordinary differential equation,

$$\frac{\partial H}{\partial x}\frac{dx}{dt} + \frac{dH}{dt} = 0,$$

with initial condition $x(1) = w$, has a unique solution path for $t \in (0, 1]$. As long as $H_x := \partial H/\partial x$ remains full rank, standard o.d.e. solution methods can be applied to integrate $x(t)$. Better yet, predictor-corrector methods use an o.d.e. method to predict along t, then apply Newton's method at fixed t to remove integration error. When H is an analytic function and the endpoint w' exists, it can be computed accurately even if H_x loses rank at $t = 0$ by use of a Puiseux series [32] or a Cauchy integral [31]. The limit might not exist if the path diverges to infinity, but in the context of solving polynomials, this phenomenon can be eliminated using a homogeneous transformation [29] (that is to say, by working on a random patch of projective space).

Path tracking is the basis of most algorithms in numerical algebraic geometry, and so it is applied automatically as necessary to solve homotopies defined inside the Bertini software package. There is also a provision for user-defined homotopies, which facilitates the development of new algorithms by the user.

Problem 2: Isolated solutions

Input : Polynomial system, $f : \mathbb{C}^N \to \mathbb{C}^n$, $N \leq n$

Output: Finite set of points, $X \subset V(f)$, that includes all isolated points in $V(f)$

For $N = n$, this functionality is provided by classical polynomial homotopy algorithms. These define a function $H(x, t)$ and compute a set of start points such that, with probability one, H is a homotopy function and after solving Problem 1 for each start point, the set of endpoints that do not diverge is the set X we seek. In the earliest formulations [15, 17], the number of homotopy paths is the total degree of f (the product of the degrees of the polynomials in f). Later approaches are able to take advantage of structure in the monomials that appear in f, such as multi-homogeneity [30] or more detailed monomial sparsity using polyhedral methods [23, 25]. For large systems, and especially for those with structure not captured by sparsity alone, the regeneration method [19] can be the most efficient approach. For $N < n$, it suffices to replace f with N random linear combinations of its polynomials, apply one of the foregoing methods, and then discard any point x for which $f(x) \neq 0$. Since such randomization often destroys sparsity, regeneration is favorable in this situation. The software package Bertini [8] provides the total-degree, multi-homogeneous, and regeneration methods. Polyhedral methods are available in software packages HOM4PS-2.0 [24] and PHC [42].

Problem 3: Numerical irreducible decomposition

Input : Polynomial system, $f : \mathbb{C}^N \to \mathbb{C}^n$

Output: A collection of witness sets, one for each irreducible component in $V(f)$

The computation of a numerical irreducible decomposition proceeds in three main stages: witness generation, junk removal, and decomposition. First, witness supersets are generated for each pure-dimensional component of $V(f)$. In essence, for each dimension i, $i = 0, \ldots, N$, this consists of choosing a system, L_i, of i random linear equations, and then applying an algorithm to solve Problem 2 for the system $\{f, L_i\}$. Although such a dimension-by-dimension procedure is valid, cascade methods are generally more efficient [35, 20]. The solution sets so obtained may contain extra points: the computation for dimension i may yield some points from some pure-dimensional components of dimension greater than i. A local dimension test [6] is used to remove these junk points. Finally, each pure-dimensional component must be factored into its irreducible pieces, which is done by monodromy [38] and verified by trace tests [39].

Problem 4: Sampling

Input : A witness set $\{f, L, W\}$ for i-dimensional irreducible component Z; a system of i linear equations L'.

Output: A finite set of points, $W' \subset Z$, that includes all isolated points of $Z \cap V(L')$.

Assuming that L is generic with respect to L', this problem is solved by tracking solution paths (see Problem 1) of $H(x,t) = 0$ defined by the homotopy function

$$H(x,t) := \{f(x), tL(x) + (1-t)L'(x)\}$$

with W as the set of start points. If L is not generic, this formulation can be modified to become a bona fide homotopy using the "gamma trick" [41, p.94]. This is called the sampling problem, because it produces new points W' on the component Z. If L' is chosen generically, then $\{f, L', W'\}$ is also a witness set for Z.

Problem 5: Membership

Input : A witness set $\{f, L, W\}$ for i-dimensional irreducible component $Z \subset \mathbb{C}^N$; a point $y \in \mathbb{C}^N$.

Output: Determine if $y \in Z$.

This can be solved using a homotopy membership test [38], as follows. Pick a generic matrix $A \in \mathbb{C}^{i \times N}$, let $L'(x) = A(x - y)$, and solve Problem 4 to get W'. Then $y \in W'$ if and only if $y \in Z$.

Problem 6: Extension

Input : A witness set $\{f, L, W\}$ for irreducible component $Z \subset \mathbb{C}^N$; a polynomial system $g : \mathbb{C}^N \times \mathbb{C}^k \to \mathbb{C}^k$.

Output: A numerical irreducible decomposition of $(Z \times \mathbb{C}^k) \cap V(g)$.

A valid, but often inefficient, procedure is to first compute a numerical irreducible decomposition of $h := \{f, g\}$ (Problem 3) and then use membership tests (Problem 5) to determine which of the irreducible components is in $Z \times \mathbb{C}^k$. Witness points are generic points of the component they represent, so if $(x, y) \in \mathbb{C}^N \times \mathbb{C}^k$ is a witness point for an irreducible component of $V(h)$ and if $x \in Z$, then that irreducible component is in $(Z \times \mathbb{C}^k) \cap V(g)$.

Since $V(h)$ contains the intersections of all the irreducible components of $V(f)$ with $V(g)$, when $V(f)$ has more than one irreducible component, it is usually more efficient to restrict computations just to $Z \subset V(f)$. One approach is to first find a numerical irreducible decomposition of $V(g)$ and then intersect each of its irreducible components with Z using the diagonal intersection homotopy [40], also available in Bertini. A more direct solution method based on regeneration is also possible [21], but this is not yet implemented in Bertini.

A variant of this problem, relevant to the cell decomposition of a surface, occurs when g is a system of k homogeneous polynomials in $k + 1$ new variables. This reduces to the above problem by appending to system g a random inhomogeneous linear equation in the new variables.

3 Cell decompositions

Our approach to computing a cell decomposition uses projections and critical sets in the spirit of classical methods as found in Tarski-Seidenberg's elimination of quantifiers [5] and applications of Morse theory [3, 28, 18]. This is in line with the numerical approach to the curve case [27].

A *cell decomposition* of a compact real surface $S \subset \mathbb{R}^N$ consists of the following:

- a finite list of vertices (0-cells), $\mathbf{V} = \{v_1, \ldots, v_{n_0}\}$, $v_i \in S$;
- a finite list of edges (1-cells), $\mathbf{E} = \{e_1, \ldots, e_{n_1}\}$, where each edge, e_i, is a structure containing two indices specifying its boundary points in $\mathbf{V}$; and
- a finite list of faces (2-cells), $\mathbf{F} = \{\mathfrak{f}_1, \ldots, \mathfrak{f}_{n_2}\}$, where each face contains indices specifying the 1-cells in $\mathbf{E}$ that form its boundary.

Furthermore, for each edge, e_i, $i = 1, \ldots, n_1$, there exists a map $\phi_i : [0, 1] \to S$, such that ϕ_i is a homeomorphism and is diffeomorphic on $(0, 1)$. Similarly, for each face, $\mathfrak{f}_i$, $i = 1, \ldots, n_2$, there exists a map, $\psi_i : A_i \to S$, where A_i is a subset of $\mathbb{R}^2$ homeomorphic to a disk. Moreover, ψ_i is a homeomorphism that is diffeomorphic on the interior of A_i and it maps the boundary of A_i to the subset of edges in $\mathbf{E}$ that are listed in $\mathfrak{f}_i$. In the cell decomposition we shall construct, we use $A_i = [0, 1] \times [0, 1]$. Finally, these cells must cover S uniquely, that is, every point of S is either a vertex in $\mathbf{V}$ or it is contained in the interior of exactly one 1-cell or 2-cell described in $\mathbf{E}$ or $\mathbf{F}$, respectively.

A cell decomposition of a compact real curve is analogous, but of course, contains only vertices and edges.

If a given surface S is not compact, there are two basic alternatives. One is to trim off any part of the surface that extends beyond a designated finite region of $\mathbb{R}^N$ and then to compute a cell decomposition of the remainder, which is now compact. For example, one could specify a box $B = [a_1, b_1] \times \cdots \times [a_n, b_n] \subset \mathbb{R}^N$ and produce a cell decomposition of $B \cap S$. An alternative is to compactify S by placing it in real projective space, $\mathbb{P}^N_{\mathbb{R}}$. That is, given a real surface S in an irreducible component Z of the solution set of a polynomial system $f(x_1, \ldots, x_n)$, as in (1), one homogenizes[1] f to obtain a homogeneous system $F(X_0, X_1, \ldots, X_n)$. Under the mapping $z \mapsto (1, z)$, the set $Z \subset \mathbb{C}^N$ maps to a Zariski open subset of some irreducible component of $V(F)$, say $Z' \subset \mathbb{P}^N$. The real part of Z', say $S' = \mathbb{P}^N_{\mathbb{R}} \cap Z'$, is compact, and we may compute a cell decomposition of S'. To implement the second approach it is convenient to first build a cell decomposition of real projective space, which we do next.

3.1 Cell decomposition of $\mathbb{P}^N_{\mathbb{R}}$

The N-dimensional real projective space is equivalent to the set of lines through the origin in $\mathbb{R}^N$ and can be represented by homogeneous coordinates $X = [X_0, X_1, \ldots, X_N] \in \mathbb{R}^{N+1}$, where $X = 0$ is not allowed, and $X \equiv Y$ if $X = aY$ for some nonzero $a \in \mathbb{R}$. A cell decomposition of $\mathbb{P}^N_{\mathbb{R}}$ can be developed from the "positive" hyperfaces of the hypercube in $\mathbb{R}^N$, as follows. First, define $\mathbb{B}_N$ to be the unit hypercube in $\mathbb{R}^N$:

$$\mathbb{B}_N := \underbrace{[-1, 1] \times \cdots \times [-1, 1]}_{N \text{times}}$$

The surface of $\mathbb{B}_{N+1}$ consists of $2(N + 1)$ hyperfaces, and we call $(N + 1)$ of these "positive", defined as

$$H_i^+ := \{X \in \mathbb{R}^{N+1} \mid X_i = 1, X_{\neq i} \in \mathbb{B}_N\}, \qquad i = 0, \ldots, N.$$

Each point in $\mathbb{P}^N_{\mathbb{R}}$ can be mapped to a point on one of the positive hyperfaces of $\mathbb{B}_{N+1}$ by dividing through by the homogeneous coordinate of largest magnitude:

$$[X_0, \ldots, X_N] \equiv [X_0, \ldots, X_N]/X_i, \quad \text{where} \quad i = \text{argmax}_i |X_i|.$$

Where the maximum coordinate is not unique, the point is mapped to a lower-dimensional hyperface that is part of the boundary of two or more hyperfaces.

The construction is easiest to understand by examining its application to $\mathbb{P}^1_{\mathbb{R}}$ and $\mathbb{P}^2_{\mathbb{R}}$, which will be the most important cases for our subsequent developments. For $\mathbb{P}^1_{\mathbb{R}}$, we have two positive hyperfaces which are just edges of the unit square:

$$E_0 = \{(1, u_1) \mid u_1 \in \mathbb{B}_1\} \text{ and } E_1 = \{(u_0, 1) \mid u_0 \in \mathbb{B}_1\}.$$

The end points of these edges are two vertices $A = (1, 1)$ and $B = (-1, 1) \equiv (1, -1)$, where we invoke the equivalence of points on the same line through the origin. This cell decomposition is illustrated in Figure 1.

[1]The homogenization a polynomial $f_i(x_1, \ldots x_n)$ of degree d_i is, after clearing denominators, $F_i(X_0, \ldots, X_N) = X_0^{d_i} f_i(X_1/X_0, \ldots, X_N/X_0)$.

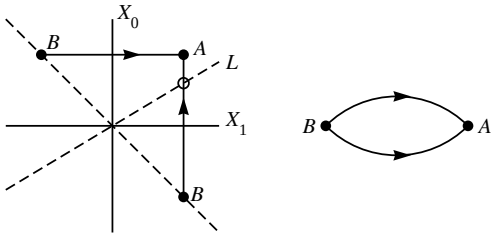

Figure 1: A cell decomposition of $\mathbb{P}^1_{\mathbb{R}}$ as the positive edges of the unit square (on left) and its topology (right). On the left diagram an arbitrary line L is drawn and an open circle marks the point used to represent it.

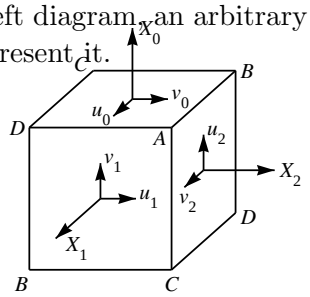

Figure 2: A cell decomposition of $\mathbb{P}^2_{\mathbb{R}}$ as the positive faces of the unit cube. Local coordinates (u_i, v_i) are indicated on the ith face, $i = 0, 1, 2$.

For $\mathbb{P}^2_{\mathbb{R}}$, we have three faces

$$F_0 = \{(1, u_0, v_0) \mid (u_0, v_0) \in \mathbb{B}_2\}, F_1 = \{(v_1, 1, u_1) \mid (u_1, v_1) \in \mathbb{B}_2\}, \text{ and}$$
$$F_2 = \{(u_2, v_2, 1) \mid (u_2, v_2) \in \mathbb{B}_2\}.$$

Referring to Figure 2, the cell decomposition has four vertices, A, B, C, D, and six edges, AB, BC, CD, DA, AC, BD. Call u_i the "right-left" local coordinate and v_i the "up-down" coordinate on each face. One sees that the right edge CA for face F_1 is the top edge for face F_2, while the left edge BD of face F_1 is the bottom edge DB of face F_2. Using local coordinates to describe this latter equivalence, we have

$$BD = \{(v_1, 1, -1) \mid v_1 \in [-1, 1]\} \equiv \{(u_2, -1, 1) \mid u_2 \in [-1, 1]\}.$$

Notice that as v_1 goes from -1 to 1, the left edge of F_1 goes from B to D, whereas as u_2 goes from -1 to 1, the bottom edge of F_2 goes from D to B. Similar relations hold *mutatis mutandis* on all three faces. When gluing together the cells to form $\mathbb{P}^2_{\mathbb{R}}$, the proper local orientation of each edge must be respected.

For the case of $\mathbb{P}^1_{\mathbb{R}}$, the edges E_i are the hyperfaces: $E_i = H_i^+$, $i = 0, 1$; whereas in the case of $\mathbb{P}^2_{\mathbb{R}}$, we have $F_i = H_i^+$, $i = 0, 1, 2$. It should be clear from these constructions how $\mathbb{P}^N_{\mathbb{R}}$ is decomposed into $(N + 1)$ cells H_i^+, $i = 0, \ldots, N$.

4 Linear projections

We shall build cell decompositions of surfaces where for each face $\mathfrak{f}_i$ the domain $A_i \subset \mathbb{R}^2$ of the map $\psi_i : A_i \to S$ has local coordinates $u \in \mathbb{R}^2$ that are an affine linear projection of the coordinates $x \in \mathbb{R}^N$. That is, let $M \in \mathbb{R}^{2\times N}$ be a rank 2 matrix, and define the affine linear projection $p : \mathbb{C}^N \to \mathbb{C}^2 : x \mapsto M \cdot x$. Analogously, in the curve case one considers such projections where M is a nonzero $1 \times N$ matrix.

More generally, suppose $f : \mathbb{R}^N \to \mathbb{R}^{N-k}$ and let $Jf(x)$ denote the Jacobian matrix of f at x:

$$Jf(x) := \frac{\partial f}{\partial x}(x). \tag{2}$$

Further, let $p : \mathbb{C}^N \to \mathbb{C}^k : x \mapsto M \cdot x$ for some matrix $M \in \mathbb{C}^{k\times N}$. The Jacobian matrix for the system (f, p) is

$$J(f,p)(x) = \begin{bmatrix} Jf(x) \\ M \end{bmatrix}. \tag{3}$$

Definition 4.1 *An affine linear projection $p : x \mapsto M \cdot x$ is* full rank *at $x^* \in V(f)$ if and only if*

$$\text{rank } J(f,p)(x^*) = N, \tag{4}$$

which is equivalent to the condition $\det J(f,p)(x^*) \neq 0$. *The points where p is not full rank are called* critical points*, and for Z an irreducible component of $V(f)$, the union of all such points in Z is the* critical set *of Z with respect to p, which we shall denote as $\mathcal{K}(Z,p)$:*

$$\mathcal{K}(Z,p) := V(f, \det J(f,p)) \cap Z. \tag{5}$$

Suppose that $x^* \in Z_{\mathbb{R}}$. Let $u^* = p(x^*) = M \cdot x^*$ for a real M, and define $p_*^{-1}(u^*) := x^*$. Then, by the implicit function theorem, if p is full rank at x^*, then p is continuously invertible on an open neighborhood of u^*. Thus, there exists a connected open two-dimensional subset $A \subset \mathbb{R}^2$ with $u^* \in A$ on which p_*^{-1} is well-defined and continuous. For any point $u \in A$, the value of $p_*^{-1}(u)$ is found by establishing any smooth path $\gamma \subset A$ from u^* to u, that is, γ is a smooth function $\gamma : [0,1] \to A$ with $\gamma(1) = u^*$ and $\gamma(0) = u$, and following the continuation path $x(t)$ defined by

$$H(x,t) = \{f(x), p(x) - \gamma(t)\} \tag{6}$$

from $x = x^*$ at $t = 1$ to $t = 0$. Then, $p_*^{-1}(u) = x(0)$. Let $\bar{A}$ be the closure of A in the standard topology. Then, the domain of p_*^{-1} can be extended to any point u in the boundary $\delta\bar{A}$ if the limit as $t \to 0$ of $x(t)$ exists and is independent of the path $\gamma(t)$ from u^* to u. In that case, we say that p is invertible on $\bar{A}$. Note that $p^{-1}(u) : \mathbb{C}^2 \to \mathbb{C}^N$ is the entire linear space in the pre-image of u for the map $p : \mathbb{C}^N \to \mathbb{C}^2$, while the restriction p_Z of p to Z, $p_Z : Z \to \mathbb{C}^2$, has a possibly multi-sheeted inverse, and $p_*^{-1}(u) : \bar{A} \to Z$ is an inverse of p_*, being p restricted locally to a single sheet of Z.

In the case at hand, where S is $Z \cap \mathbb{R}^N$ for Z an irreducible algebraic set, then p is invertible on a set A that is homeomorphic to a closed disk if p is full rank on the interior of A and $p_*^{-1}(u)$ as defined by continuation does not diverge to infinity anywhere on A. Thus, to build a cell decomposition of S, we need to carve S into such pieces. If S is not compact, it is not possible to cover it with a finite number of closed faces, making a full cell decomposition not feasible in $\mathbb{R}^N$. As discussed above, in this case two remedies are available: we can decompose only a finite piece of S, for example, decompose the intersection of S with a box, or we can compactify S into projective space and decompose it there. In the case that we choose to decompose the intersection of S with a box, some edges of the cells may be the intersection of S with faces of the box.

The following lemmas allow us to construct cells A_i with valid maps ψ_i taken as local inverses of a projection.

Definition 4.2 *For a k-dimensional irreducible algebraic set $Z \subset \mathbb{C}^N$, we say that Z is in general position with respect to a linear projection $p : \mathbb{C}^N \to \mathbb{C}^k$ if p is generically full rank on Z and $p_Z^{-1}(u)$ does not diverge to infinity for any $u \in \mathbb{C}^2$.*

Lemma 4.3 *For a k-dimensional irreducible and generically reduced algebraic set $Z \subset V(f) \subset \mathbb{C}^N$, there is a dense Zariski-open subset U of $\mathbb{R}^{k \times N}$ such that for any $M \in U$, Z is in general position with respect to projection $p : x \mapsto M \cdot x$.*

Proof. The assumption that Z is generically reduced means that Jf is generically rank $N - k$ on Z, and hence $J(f, p)$ is generically full rank for (x, M) in $Z \times \mathbb{R}^{k \times N}$. The non-divergence condition can be seen to hold by considering the homogenization F of f and the irreducible component $Z' \subset \mathbb{P}^N$ of $V(F)$ that corresponds to Z. By assumption, Z is an irreducible component in $\mathbb{C}^N$ so Z' is not entirely in the hyperplane at infinity. This implies that $Z' \cap V(X_0)$ is $(k-1)$-dimensional; call it Z_∞. Divergence of $p_Z^{-1}(u)$ for $u \in \mathbb{C}^k$ occurs where

$$M \cdot \left[X_1, \ldots, X_N\right]^T = 0$$

for $[0, X_1, \ldots, X_N] \in Z'$. But k generic hyperplanes will not meet a $(k-1)$-dimensional algebraic set, so for generic M divergence does not happen for any $u \in \mathbb{C}^k$. □

Remark 4.4 Note that Lemma 4.3 also holds when restricting M to have orthonormal rows, because orthonormalizing the rows of a matrix M does not change the linear subspace defined by $M \cdot x = 0$

Corollary 4.5 *For a k-dimensional irreducible and generically reduced component Z of $V(f)$ and a projection p that is the natural projection of $\mathbb{C}^N$ onto k of its coordinates, there is a dense Zariski-open subset $U \subset O_{\mathbb{R}}(N)$ (where $O(N)$ is the set of orthogonal matrices of size N) such that for any $R \in U$, the algebraic set $Y = R \cdot Z$ is in general position with respect to p.*

Proof. Since Y and Z are related by a nonsingular change of coordinates, Y is in general position with respect to p if Z is in general position with respect to a projection formed from the k rows of R corresponding to the k coordinates picked out by p. Thus, Remark 4.4 applies. □

Lemma 4.6 *Let Z be a 2-dimensional irreducible and almost smooth component of $V(f)$, and for R an orthogonal matrix of size N, let $Y = R \cdot Z$ be a rotation of Z. let p be the natural projection of $\mathbb{C}^N$ onto 2 of its coordinates, and let p_j $j = 1, 2$ be the natural projections of $\mathbb{C}^N$ onto, respectively, one of the two coordinates in the range of p. Let $C_i^1 \subset Y$ $i = 1, \ldots, m_1$ be a finite number of cross-sectional curves of Y of the form $Y \cap p_1^{-1}(c_{1,i})$ and similarly let $C_i^2 \subset Y$ $i = 1, \ldots, m_2$ be $Y \cap p_2^{-1}(c_{2,i})$ for any constants $c_{1,i}, c_{2,i} \in \mathbb{R}$. Then there is a dense Zariski-open subset $U \subset O_{\mathbb{R}}(N)$ (where $O(N)$ is the set of orthogonal matrices of size N) such that for any $R \in U$, the algebraic set $Y = R \cdot Z$ is in general position with respect to p and the algebraic set $\mathcal{K}(Y, p)$ is in general position with respect to p_1 and p_2, C_i^1, $i = 1, \ldots, m_1$ are all in general position with respect to p_2, and C_i^2, $i = 1, \ldots, m_2$ are all in general position with respect to p_1*

Proof. Y is in general position by Corollary 4.5. This implies that $\mathcal{K}(Y, p)$ is a curve, and by the almost smoothness of Z, it has at most a finite number of singularities. Since the rotation R can be decomposed into an initial rotation that fixes $\mathcal{K}(Y, p)$ as a subset of Y followed by a rotation in the subspace of the first two coordinates, $\mathcal{K}(Y, p)$ is in general position with respect to these first two coordinates. Similarly, after an initial rotation fixes the cross-sectional curves C_i^1, they are placed in general position with respect to p_2 by rotations that leave $p_1(x)$ fixed, and *mutatis mutandis*, the same holds for C_i^2. Each of these claims holds for R on a Zariski-open set of $O_{\mathbb{R}}(N)$, and the intersection of a finite number of Zariski-open sets is still a Zariski-open set. □

The system of which $Y = R(Z)$ is an irreducible component is simply $g(y) = f(R^{-1}y)$. A cell decomposition of Y can be mapped to the original coordinates as $z = R^{-1}y$ to get the decomposition we seek. The matrix R could be chosen from the nonsingular $N \times N$ matrices, but we take it as a random orthogonal matrix for the sake of good numerical conditioning.

In working with polynomials symbolically, a random change of coordinates can be an expensive operation, as it transforms a sparse polynomial system into a dense one. This may make subsequent operations on the polynomial system exorbitantly expensive. However, in a numerical setting, where we work with straight-line programs, $g(y)$ is simply evaluated in a two-step process as $x = R^{-1}y$, $g(y) = f(x)$. Similarly, by the chain rule, $Jg := \partial g/\partial y = Jf \cdot R^{-1}$. With a random change of coordinates, we only need the natural linear projections onto individual coordinates:

$$\pi_i(z_1, \ldots, z_N) = z_i, \quad i = 1, \ldots, N. \tag{7}$$

The following lemma, which says that critical points of a slice of a surface are critical

points of the surface, is useful in piecing together boundaries in the cell decomposition of a surface.

Lemma 4.7 *Let $Z \subset \mathbb{C}^N$ be a two-dimensional irreducible component of the solution set of f as in (1), and let $p_1, p_2 : \mathbb{C}^N \to \mathbb{C}$ be linear projections. Let z^* be a point of Z, and let C be the component in $Z \cap p_1^{-1}(p_1(z^*))$ 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 p_2, then z^* is also a critical point of Z with respect to $p = (p_1, p_2)$.*

Proof. By assumption, C is a component of $g^{-1}(0)$, where $g = \{f(z), p_1(z) - p_1(z^*)\}$. Let Jf and Jg be the Jacobian matrices of f and g, resp., and let M_i be the $1 \times N$ matrix such that $p_i : (z) \mapsto M_i \cdot z$, $i = 1, 2$. Recognizing that $Jg = \begin{bmatrix} Jf \\ M_1 \end{bmatrix}$, then

$$J(f, p) \equiv J(g, p_2) = \begin{bmatrix} Jf \\ M_1 \\ M_2 \end{bmatrix},$$

so the lemma follows immediately. □

5 The one-dimensional case

The two dimensional cell decomposition algorithm relies on the one dimensional version that we recall here. The problem statement below includes some features that make surface decomposition easier, even though these features are strictly not needed for curve decomposition. We first present the algorithm for decomposing a curve, with the option of restricting the decomposition to an assigned interval. To fully decompose a curve, this version of the algorithm is called either once or twice, depending on whether the curve is known *a priori* to be compact. The use of this algorithm to compute the complete decomposition of a curve is discussed at the end of this section.

The assumption that a curve is irreducible is not restrictive in view of (Problem 3).

5.1 Curve cell decomposition (over assigned intervals I_1 and I_2)

Some notes on the problem statement are necessary. For brevity, let $Z_{\mathbb{R}} = Z \cap \mathbb{R}^N$ be the set of real points of Z. If Z is not self-conjugate, then $Z_{\mathbb{R}}$ is finite and can be found via intersection of Z with its complex conjugate. Hence, we may assume that Z is self-conjugate. Furthermore, the assumption in the problem statement that Z is generically reduced means that Jf is generically rank $N-1$ and by randomization (see Section 2), we may assume that f consists of $N-1$ polynomials. By "general position," it is meant that Z is in general position with respect to p_1, and if present, also with respect to p_2. The optional argument $\mathcal{I}_1$ can be omitted only if it is known that $Z_{\mathbb{R}}$ is compact. Finally, the role of the input argument *merge*, which is specified as either

Problem 7: Curve Cell Decomposition (over assigned intervals I_1 and I_2)	
Input	: A witness set $Z_w = \{f, L, W\}$ for 1-dimensional irreducible and reduced algebraic set $Z \subset \mathbb{C}^N$, where f has real coefficients, Z in general position (see notes); independent real linear projections $p_1, p_2 : \mathbb{C}^N \to \mathbb{C}$; real intervals $\mathcal{I}_1 = [a_1, b_1]$, $\mathcal{I}_2 = [a_2, b_2]$; a finite subset $T \subset \mathcal{I}_1$, and logical value *merge*. Inputs p_2, $\mathcal{I}_1$, $\mathcal{I}_2$, and T are optional.
Output	: A cell decomposition, $\mathcal{D}_1(Z_w, p_1, \mathcal{I}_1, p_2, \mathcal{I}_2, T, \mathit{merge})$ of $Z \cap \mathbb{R}^N$ or, as appropriate, $Z \cap \mathbb{R}^N \cap p_1^{-1}(\mathcal{I}_1) \cap p_2^{-1}(\mathcal{I}_2)$.

"true" or "false," will become clear in step 5 of the algorithm below. Later, the merge option is used in the two-dimensional algorithm to simplify certain cross-sectional curves of a real surface. The values of T are used to specify required breakpoints in the cell decomposition. If T is present, then *merge* must be false, because otherwise the merging procedure may eliminate the breakpoints added by T.

The output, that is, the cell decomposition $\mathcal{D}_1$, is a structure that contains the following:

- a list of vertices, $\mathbf{V} = \{v_1, \ldots, v_{n_0}\}$, $v_i \in Z_{\mathbb{R}}$; and
- a list of edges, $\mathbf{E} = \{e_1, \ldots, e_{n_1}\}$.

Furthermore, each edge, e_i, $i = 1, \ldots, n_1$ is a structure that contains:

- the polynomial system f;
- the projection p_1;
- the indices ℓ and r of the edge's left and right endpoints in $\mathbf{V}$,
- a general point $w \in Z_{\mathbb{R}}$ of the edge.

The precise meaning of w is described below. For an edge e_i, the data (f, p_1, ℓ, r, w) suffice to determine the homeomorphic map $\phi_i : [0, 1] \to Z_{\mathbb{R}}$ that describes the edge, as required by the specification of a cell decomposition in Section 3. One could question the inclusion of f and p_1 as edge-related data, given that they are common to all edges of the curve. This choice is motivated by the fact that in the surface decomposition we will have edges from several different curves, hence the edges need to store information about which curve they come from. Of course, rather than storing a representation of f in many different edges, it would be more efficient to build a database of systems and have the edges link to these. Nothing essential changes in the algorithm if one thinks of f as a pointer to the relevant system instead of as a copy of the system itself.

To solve the problem, we follow [27] with a few minor changes. First, the method in [27] uses a random rotation of coordinates to ensure that Z is in general position. Here, general position is assumed, attained by a random rotation before calling the algorithm.

Second, we introduce the option of decomposing only that part of $Z_{\mathbb{R}}$ that projects to the intervals $\mathcal{I}_1$ and $\mathcal{I}_2$. This comes into play when decomposing a surface and also plays a role in performing a complete cell decomposition of a noncompact curve. Finally, the arguments T and *merge* affect the details of how finely the curve is decomposed.

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 find $Z_{\mathbb{R},0}$ and to break $Z_{\mathbb{R},1}$ into 1-cells. In the output described above, the points $Z_{\mathbb{R},0}$ will be contained in the list of vertices, $\mathbf{V}$. They are distinguishable as the points in $\mathbf{V}$ which are not the endpoints of any edges in $\mathbf{E}$.

We give ϕ_i for each 1-cell implicitly in terms of a homotopy function as follows. Let v_ℓ and v_r be the left and right endpoints of the 1-cell, and let $c_\ell = p_1(v_\ell)$ and $c_r = p_1(v_r)$. Define the homotopy

$$H(x,t) = \begin{bmatrix} f(x) \\ p_1(x) - ((1-t)c_\ell + tc_r) \end{bmatrix} \tag{8}$$

to start from $(x,t) = (w,t_0)$, where $w \in Z_{\mathbb{R}}$ with $c_\ell < p(w) < c_r$ and

$$t_0 = (p_1(w) - c_\ell)/(c_r - c_\ell).$$

Using a path tracker (see Problem 1: Path tracking), we can evaluate x for any value of $t \in [c_\ell, c_r]$, and thus define $\phi(t) := x(t)$ on that interval.

The homotopy function must be such that $H_x := \partial H/\partial x$ remains full rank as t ranges over $(0,1)$. Since $H_x = J(f,p_1)$, the assumption that Z is in general position implies that H_x is full rank on a Zariski-open set of Z. Hence, the key step in finding a cell decomposition of $Z_{\mathbb{R}}$ is to find the critical points of p_1 on $Z_{\mathbb{R}}$ and set the boundaries of the 1-cells to match. When p_2 and $\mathcal{I}_2$ are given, we additionally truncate any edges that go beyond $p_2^{-1}(\mathcal{I}_2)$.

Accordingly, an algorithm to compute a cell decomposition of $Z_{\mathbb{R}}$ is as follows.

1. **Find critical points:** Find the set of critical points $\mathcal{B}$ on $Z_{\mathbb{R}}$ with respect to p_1, that is, find the solutions in Z of the system

$$f_{crit_0}(x) = \begin{bmatrix} f(x) \\ \det J(f,p_1)(x) \end{bmatrix} = 0. \tag{9}$$

   As we have a witness set for Z, this is a case of Problem 6: Extension. Because Z is one-dimensional and p_1 is generically full rank on Z, $V(f_{crit_0})$ is a set of isolated points.

As an alternative, one may solve

$$f_{crit_1}(x,\xi) = \begin{bmatrix} f(x) \\ J(f,p)(x)\cdot\xi \end{bmatrix} = 0, \qquad x \in Z \subset \mathbb{C}^N, \xi \in \mathbb{P}^{N-1}, \tag{10}$$

which is another variation of the Extension Problem. This might be preferred over f_{crit_0} for avoiding the determinant, but that comes at the cost of introducing new variables and more polynomials. Moreover, as discussed in [27], the solution of $V(f_{crit_1})$ must be done with a method that finds solutions at all dimensions, since singular points of Z may have higher-dimensional tangent spaces satisfying (10). Let $\bar{p} : \mathbb{C}^N \times \mathbb{P}^{N-1} \to \mathbb{C}^N$ be the natural projection $(z,\xi) \mapsto (z)$. By the assumption that Z is a multiplicity one curve, for any solution component V of $Z \cap V(f_{crit_1})$, $\bar{p}(V)$ must consist of isolated points.

Regardless of whether one uses (9) or (10), $\mathcal{B}$ is the set of real solution points obtained. Record the results, respecting the limits imposed by $\mathcal{I}_1$ and $\mathcal{I}_2$ as follows:

(a) If $\mathcal{I}_1$ is given, replace $\mathcal{B}$ by $\mathcal{B} \cap p_1^{-1}(\mathcal{I}_1)$.

(b) If $\mathcal{I}_2$ is given, replace $\mathcal{B}$ by $\mathcal{B} \cap p_2^{-1}(\mathcal{I}_2)$.

(c) Initialize $\mathbf{V} = \mathcal{B}$, and insert $p_1(\mathbf{V})$ into T.

(d) If $\mathcal{I}_1$ is given, append a_1 and b_1 to T.

2. **Apply $\mathcal{I}_2$ limits:** If $\mathcal{I}_2 = [a_2, b_2]$ is given,

(a) Solve for $C^- := Z_{\mathbb{R}} \cap p_2^{-1}(a_2)$ and $C^+ := Z_{\mathbb{R}} \cap p_2^{-1}(b_2)$. These are cases of Sampling (Problem 4) applied to Z with linear system L' being $p_2(x) = a_2$ or $p_2(x) = b_2$ and keeping only the real points found. The assumption that Z is in general position with respect to p_2 implies that C^- and C^+ are finite sets of isolated points.

(b) For every point $x \in C^- \cup C^+$, if $p_1(x) \in \mathcal{I}_1$, then append x to $\mathbf{V}$ and $p_1(x)$ to T.

3. **Cut at t_i:**

(a) Sort the entries of $T = \{t_1, \ldots, t_m\}$ into ascending order.

(b) Let

$$\mathcal{E}_i = \{p_1^{-1}(t_i) \cap Z_{\mathbb{R}}\} \setminus \mathcal{B}, \qquad i = 1, \ldots, m.$$

Since $p_1(x) = t_i$ is a linear system, this is another case of Sampling (Problem 4), keeping only the real solutions found and dropping any that are already in $\mathcal{B}$.

(c) For $j = \{1, m\}$, if $p_1^{-1}(t_j) \cap\ Z_{\mathbb{R}} = \emptyset$, delete t_j from T. This can only happen if a_1 or b_1 is beyond the limits of $p_1(Z_{\mathbb{R}})$. If necessary, renumber the elements of T to once again run from 1 to m.

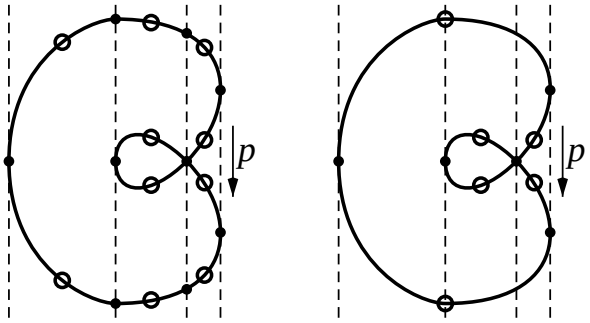

Figure 3: A decomposition of a curve before and after merging edges that meet outside of $\mathcal{B}$. Filled dots are edge endpoints, the centers of the open circles mark general points on the edges.

(d) Append all the points $\mathcal{E}_i$, $i = 1, \ldots, m$ to $\mathbf{V}$.

4. **Find edges:** For $i = 1, \ldots, m-1$, do the following.

   (a) **Find general points:** For $i = 1, \ldots, m-1$, let t_i^* be a generic number in (t_i, t_{i+1}) and solve to find $W_i^* = p_1^{-1}(t_i^*) \cap Z_{\mathbb{R}}$. Again, this is a case of Sampling (Problem 4), this time with the linear system $p_1(x) = t_i^*$. Since t_i^* is between, not at, critical values, any value in the interval suffices. We set the convention of using the midpoint, $t_i^* = (t_i + t_{i+1})/2$.

   (b) **Find endpoints:** For $i = 1, \ldots, m-1$ and each $w \in W_i^*$,

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

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

      iii. Record a new edge in $\mathbf{E}$ having left index ℓ such that $\alpha = v_\ell \in \mathbf{V}$, right index r such that $\beta = v_r \in \mathbf{V}$, and general point w.

   The homotopy for these steps is $H(x,t) := \{f(x), \pi(x) - t\} = 0$, and tracking the path is a case of the Problem 1. Any path that ends at a point in $\mathcal{B}$ has a singular endpoint, but the singular endgame of the tracker algorithm accommodates this possibility.

5. **Merge:** If *merge* is "true," then replace any two edges that meet at a point in $\mathcal{E} = \cup_{i=1}^m \mathcal{E}_i$ with a single edge. That is, if edge $e = (\ell, r, w)$ meets edge $e' = (\ell', r', w')$ at vertex $v = v_r = v_{\ell'} \in \mathcal{E}$, these are merged to form a single edge $e'' = (\ell, r', v)$. Then e and e' are deleted from $\mathbf{E}$, e'' is inserted into $\mathbf{E}$, and vertex v is deleted from $\mathbf{V}$.

   The effect of merging edges is illustrated in Figure 3. The points in the set $\mathcal{E}$ are not critical points of the projection, so edges can cross them while keeping a nonsingular homotopy function.

It may happen that a curve has no real critical points at Step 1. However, this implies that the curve is noncompact, so $\mathcal{I}_1$ must be present and T cannot be empty at Step 3. In this way, the algorithm finds the portion of the curve that crosses through $p_1^{-1}(\mathcal{I}_1)$.

5.2 Compact curve cell decomposition

If a curve is known to be compact, then the above algorithm can be called with p_1 being a random real projection and no limits.

5.3 Noncompact curve cell decomposition

If a curve is noncompact or its compactness is unknown, we can still perform a complete cell decomposition by first projectivizing the curve and then performing decomposition on two cells of $\mathbb{P}^1_{\mathbb{R}}$. While this is not needed for the main surface decomposition problem, we include it for completeness.

Assume that $Z \subset \mathbb{C}^N$ is an irreducible and reduced curve in $V(f)$, and let F be the homogenization of f. Under the mapping $\eta : z \mapsto (1, z)$, Z maps to a Zariski-open subset of some irreducible component of $F^{-1}(0)$, say $Z' \subset \mathbb{P}^N$. Given a witness set $Z_w = \{f, L, W\}$ for Z, a witness set for Z' is $Z'_w = \{F, L', W'\}$, where L' is the homogenization of L and $W' = \eta(W)$. Once we compute a cell decomposition for Z', any finite part of it can be mapped back to the corresponding part of Z using the map $\mu_0 : (Z_0, Z_1, \ldots, Z_N) \mapsto (Z_1/Z_0, \ldots, Z_N/Z_0)$. Consequently, we consider the noncompact case to be solved if we can compute a cell decomposition of Z'. To aid readability in the algorithm to follow, we rename Z', L', W' as simply Z, L, W.

Problem 8: Projective Curve Cell Decomposition

Input : A witness set $\{F, L, W\}$ for 1-dimensional irreducible and generically reduced algebraic set $Z \subset \mathbb{P}^N$, where F is a system of homogeneous polynomials with real coefficients.

Output: A cell decomposition, $\mathcal{D}_1(Z)$.

In the following, and throughout this article, we use "[ ]" to denote an argument that is omitted.

1. **Change coordinates:** Choose a random real rotation matrix $R \in O(\mathbb{R}, N+1)$ and set $F'(y) = F(R^{-1}y)$. Modify L and W to match to get L' and W'. Define $Y = R \cdot Z$.

2. **Decompose $Y_{\mathbb{R}}$ on patch 0:** Let $f_0 := \{F', y_0 - 1\}$ and let $W_0 = \mu_0(W')$, where $\mu_0 : (y_0, y_1, \ldots, y_N) \mapsto (1, y_1/y_0, \ldots, y_N/y_0)$. Then compute a curve cell decomposition with limits as

$$D^0 = \mathcal{D}_1(\{f_0, L', W_0\}, \pi_1, [-1, 1], [\,], [\,], [\,], \text{true}).$$

3. **Decompose $Y_{\mathbb{R}}$ on patch 1:** Let $f_1 := \{F', y_1 - 1\}$ and let $W_1 = \mu_1(W')$, where $\mu_1 : (y_0, y_1, \ldots, y_N) \mapsto (y_0/y_1, 1, y_2/y_1, \ldots, y_N/y_1)$. Then compute a curve cell decomposition with limits as

$$D^1 = \mathcal{D}_1(\{f_1, L', W_1\}, \pi_0, [-1, 1], [\,], [\,], [\,], \text{true}).$$

4. **Paste cells:** As illustrated in Figure 1, the two cells of $\mathbb{P}^1_{\mathbb{R}}$ meet at their left and right ends. The cells are pasted together by recognizing equivalences between points in projective space. This means the vertex lists in D^0 and D^1 are merged in $\mathbb{P}^N_{\mathbb{R}}$ and the indices in the two edge lists are updated to match.

5. **Restore coordinates:** All the data is restored to the original coordinate system using the R^{-1} map.

6 Two dimensional case

The following algorithm for decomposing a surface over a box is analogous to the curve cell composition over intervals. One can use it directly to decompose a compact surface in one fell swoop, or one can decompose the projectivization of a noncompact surface by three applications of the algorithm over the cells of $\mathbb{P}^2_{\mathbb{R}}$ illustrated in Figure 2.

Problem 9: Surface Cell Decomposition over a box

Input : A witness set $Z_w = \{f, L, W\}$, where f has real coefficients, for a self-conjugate 2-dimensional irreducible and almost smooth algebraic set $Z \subset \mathbb{C}^N$ in general position with respect to linear projection $p := (p_1, p_2)$ for independent real linear projections $p_1, p_2 : \mathbb{C}^N \to \mathbb{C}$; (optional) real intervals $\mathcal{I}_1 = [a_1, b_1]$ and $\mathcal{I}_2 = [a_2, b_2]$.

Output: A cell decomposition, $S = \mathcal{D}_2(Z_w, p_1, \mathcal{I}_1, p_2, \mathcal{I}_2)$ of $Z_{\mathbb{R}}$ or, as appropriate, $Z_{\mathbb{R}} \cap p_1^{-1}(\mathcal{I}_1) \cap p_2^{-1}(\mathcal{I}_2)$.

As in the curve case, the intervals $\mathcal{I}_1$ and $\mathcal{I}_2$ are optional only if it is known that $Z_{\mathbb{R}}$ is compact.

Remark 6.1 The assumptions on self-conjugacy and almost smoothness of Z are easily checked. To check self-conjugacy, take a witness point w of Z and check if the conjugate of w lies on Z (see Problem 5: Membership). Since w is a generic point of Z, this test decides the matter. The almost smoothness condition requires that the witness points of Z be nonsingular, implying that Z is singular on at most a curve. This is easily assessed by testing the rank of the system Jacobian matrix at a single witness point. Further testing to evaluate the almost smoothness of Z occurs at Step 2 below.

Remark 6.2 The algorithm we give for decomposing a surface will still work properly for some exceptions to the almost smoothness condition. For example, any singularity

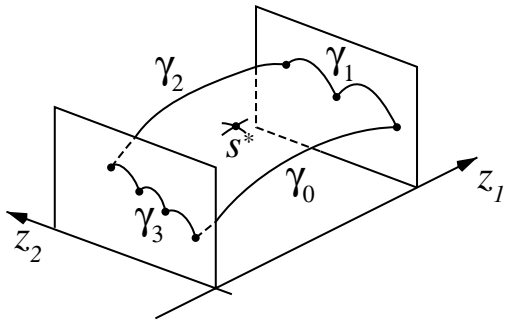

Figure 4: A two cell: generic point s^*; simple edges γ_0 and γ_2; and compound edges γ_1 and γ_3.

sets 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.

Similar to the one-dimensional case, the output, that is, the cell decomposition S, is a structure that contains the following:

- a list of vertices, $\mathbf{V} = \{v_1, \ldots, v_{n_0}\}$, $v_i \in Z_{\mathbb{R}}$;
- a list of edges, $\mathbf{E} = \{e_1, \ldots, e_{n_1}\}$; and
- a list of faces, $\mathbf{F} = \{\mathfrak{f}_1, \ldots, \mathfrak{f}_{n_2}\}$.

The isolated points $Z_{\mathbb{R},0}$ are those points of $\mathbf{V}$ that have no edges connected to them, while the faces form $Z_{\mathbb{R},2}$.

The 2-cells we will construct for $Z_{\mathbb{R},2}$ have a particular form, illustrated in Figure 4. Each face, $\mathfrak{f}_i$, $i = 1, \ldots, n_2$, has an associated map $\psi_i : [0,1] \times [0,1] \to Z_{\mathbb{R}}$ that is defined implicitly by a homotopy (see below) that initializes at an interior point. Accordingly, a face is a structure consisting of:

- the polynomial system f;
- a general point s^* in the interior of the face,
- a list $\gamma_0 \subset \mathbb{N}$ of indices for the edges forming $\psi_i([0,1], 0)$, the "down" boundary,
- a list $\gamma_1 \subset \mathbb{N}$ of indices for the edges forming $\psi_i(1, [0,1])$, the "right" boundary,

- a list $\gamma_2 \subset \mathbb{N}$ of indices for the edges forming $\psi_i([0,1],1)$, the "up" boundary, and
- a list $\gamma_3 \subset \mathbb{N}$ of indices for the edges forming $\psi_i(0,[0,1])$, the "left" boundary.

As illustrated, the up and down boundaries are single edges, while the left and right boundaries can be a single edge or multiple edges connected end-to-end, or they can degenerate to a single point. In the case of degeneration to a point, γ_1 or γ_3 is simply an empty set.

The map ψ_i for each face $\mathfrak{f}_i$ will be given implicitly in terms of a homotopy function as follows. The cell will be constructed such that the entire left boundary, γ_3, is mapped to the same value by p_1, say $c_\ell = p_1(\gamma_3)$. Similarly, the entire right boundary γ_1 is mapped to a constant, say $c_r = p_1(\gamma_1)$. Meanwhile, the general point s^* is mapped to $p_1(s^*)$ lying between c_ℓ and c_r. Moreover, the general point, say w_0, of the down edge γ_0 and the general point, say w_2, of the up edge γ_2 are such that $p_1(w_0) = p_1(w_2) = p_1(s^*) = (c_\ell + c_r)/2$. Also, edges γ_0 and γ_2 have associated polynomial systems f_0 and f_2, respectively. Then, the homotopy for computing $\psi(u,v)$ for any $(u,v) \in [0,1]\times[0,1]$ is

$$H(x,y_0,y_2,u,v) = \begin{bmatrix} f(x) \\ p_1(x) - [(1-u)c_\ell + uc_r] \\ f_0(y_0) \\ p_1(y_0) - [(1-u)c_\ell + uc_r] \\ f_2(y_2) \\ p_1(y_2) - [(1-u)c_\ell + uc_r] \\ p_2(x) - [(1-v)p_2(y_0) + vp_2(y_2)]/[p_2(y_2) - p_2(y_0)] \end{bmatrix} \tag{11}$$

initialized at $(x,y_0,y_2,u,v) = (s^*,w_0,w_2,1/2,1/2)$. By abuse of language, we refer to $H(x,y_0,y_2,u,v)$ as a homotopy, because whenever it is used to do continuation, a path $(u(t),v(t))$ is specified. After tracking from $(u,v) = (1/2,1/2)$ to any other (u,v) following a continuous path in $[0,1]\times[0,1]$, never touching the boundary except possibly at the end of the path, the evaluation of $\psi(u,v)$ is the corresponding x. The idea behind this homotopy is that y_0 and y_2 move left-right on edges γ_0 and γ_2 while x tracks with them in the p_1 direction and interpolates between them in the p_2 direction. The point x only reaches the boundaries of the face at the limits of (u,v) on the box $[0,1]\times[0,1]$, and the cell is constructed so that no other segments of the critical curve for Z with respect to p cross into the face. The homotopy may become singular at any of the boundaries, but if so, a singular endgame in the path tracker finds the limit of the end of the path, thereby extending ψ to the boundaries of the cell.

The construction of the cell decomposition is best described by illustrating the action of the algorithm step by step on a simple example. To that end, we use a surface in $\mathbb{R}^3$ defined by the equation

$$((x+0.35)^2(1-x^2) - y^2)^2 + z^2 - 0.00531441 = 0. \tag{12}$$

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

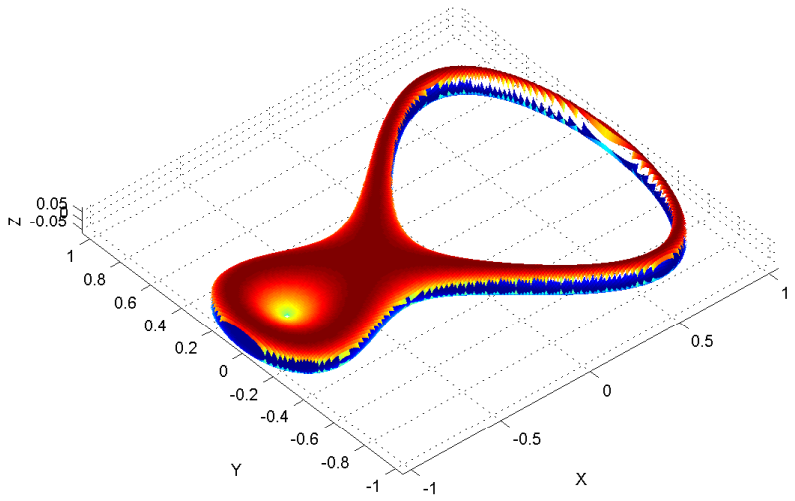

Figure 5: The running example drawn by meshing a grid of samples.

(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 5. The mesh is two disconnected pieces, jagged around the boundary, because this crude approach does not solve for the boundary curve.

The algorithm is illustrated with the compact surface rotated to make sure it is in general position with respect to $\pi = (\pi_1, \pi_2)$. It is easier to visualize these natural projections than if we were to decompose the surface in its original coordinates but using random projections, even though the two approaches are equivalent. Figure 6 illustrates the example surface as a rotated point cloud, which we will decompose in the algorithm below. In this particular case, the rotation is unnecessary, but we apply it anyway to illustrate the general procedure.

The algorithm for the two-dimensional case is as follows. Because the algorithm involves cell decompositions of several different curves, we use the dot notation to indicate fields in a structure. For example, for a cell decomposition C, the symbol $C.\mathbf{V}$ indicates the set of vertices in C. The algorithm builds S, the cell decomposition of $Z_{\mathbb{R}}$.

For clarity, we note that at several steps we construct cell decompositions for slices of $Z_{\mathbb{R}}$ in the vertical direction, meaning we decompose $Z_{\mathbb{R}} \cap p_1^{-1}(c)$ for some $c \in \mathbb{R}$. When we do these decompositions, we use p_2 as the projection. Then, "left" and "right" of the 1-cell decomposition of the slice with respect to p_2 are "down" and "up" in the 2-cell decomposition of the surface.

1. **Initialize:** Instantiate S with $S.\mathbf{V}$, $S.\mathbf{E}$ and $S.\mathbf{F}$ all empty.

2. **Find witness set for critical curve $\mathcal{K}(Z, p)$:** Let $G(z) = \{f, \det J(f, p)\}$ and find a numerical irreducible decomposition of $\mathcal{K}(Z, p) = Z \cap V(G)$. This is a case

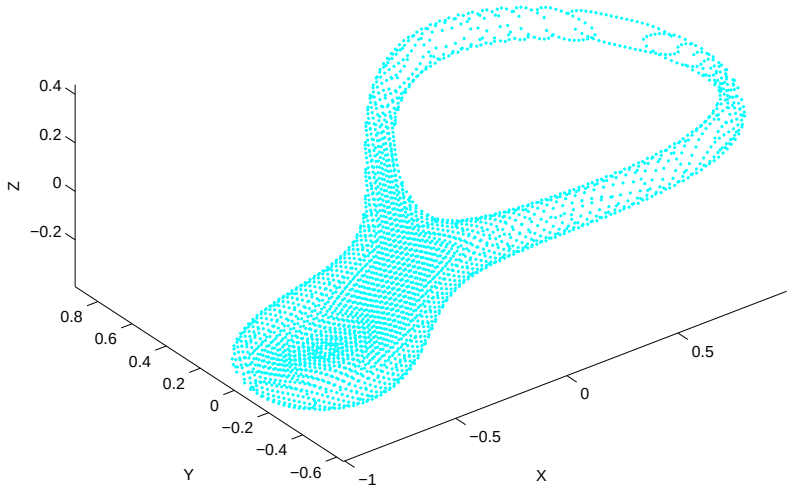

Figure 6: Point cloud on the surface after general rotation.

of the Extension Problem 6, beginning from the witness set Z_w for Z. If $\mathcal{K}(Z,p)$ is two-dimensional, then Z is not in general position with respect to p, and the algorithm stops. Else, if the witness points of $\mathcal{K}(Z,p)$ are singular, Z is not almost smooth, and again, the algorithm stops. Otherwise, Z is in general position and almost smooth, and the algorithm proceeds with $\mathcal{K}_w$ being the witness set for $\mathcal{K}(Z,p)$.

3. **Decompose the critical curve:**

   (a) Use the curve algorithm to compute a cell decomposition of $\mathcal{K}_{\mathbb{R}}$:

   $$C = \mathcal{D}_1(\mathcal{K}_w, p_1, \mathcal{I}_1, p_2, \mathcal{I}_2, \text{false}).$$

   (b) Let T be as developed in the curve algorithm; that is, $T = p_1(C.\mathbf{V})$ sorted in ascending order.

   (c) Set $S.\mathbf{V} = C.\mathbf{V}$ and $S.\mathbf{E} = C.\mathbf{E}$.

   Figure 7 shows the critical curve and its decomposition for the illustrative example.

4. **Add up and down limit curves:** These are the curves where $Z_{\mathbb{R}}$ intersects the limits imposed by $\mathcal{I}_2$. Recall that $Z_w = \{f, (L_1, L_2), W\}$ where L_1 and L_2 are generic linear polynomials. To slice the surface with a hyperplane given by linear equation P, we may move L_1 to P using Sampling (Problem 4), getting witness point set W'. Then, a witness set for $Z \cap V(P)$ is simply $\{(f, P), L_2, W'\}$.

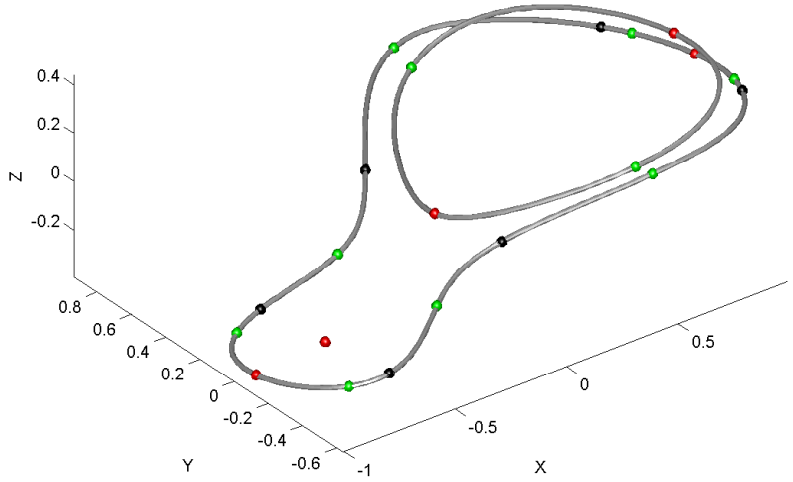

Figure 7: Decomposition of the critical curve at Step 3. Red dots are critical points, black dots are endpoints from slicing at the critical points, green dots are the generic points of the 1-cells. There are ten 1-cells, and the isolated pinch point.

(a) Find a witness set, say U_w^-, for the lower limit curve $Z \cap V(p_2(z) - a_2)$.

(b) Compute

$$U^- = \mathcal{D}_1(U_w^-, p_1, \mathcal{I}_1, [\,], [\,], T, \text{false}),$$

a cell decomposition for the real part of the lower limit curve, with breakpoints T that agree with those of C. Note that by Lemma 4.7, any vertex in C where the critical curve $\mathcal{K}$ crosses the lower limit $p_2^{-1}(a_2)$ is also a critical point of $Z \cap V(p_2(z) - a_2)$ with respect to projection p_1.

(c) Similarly, for the upper limit curve $Z \cap V(p_2(z) - b_2)$, compute a witness set U_w^+ and then its cell decomposition $U^+ = \mathcal{D}_1(U_w^+, p_1, \mathcal{I}_1, [\,], [\,], T, \text{false})$.

(d) Append the vertices and edges in U^- and U^+ to $S.\mathbf{V}$ and $S.\mathbf{E}$, and for any new point x appended to $S.\mathbf{V}$ also append $p_1(x)$ to T.

5. **Slice $Z_{\mathbb{R}}$ vertically at critical values:** Let the elements of T be $t_1, \ldots, t_m$. Then, for $i = 1, \ldots, m$ do:

(a) Using Sampling, find a witness set for $Z \cap p_1^{-1}(t_i)$. Let K_w^i be the witness set for this vertical slice of Z.

(b) Find the curve decomposition

$$K_i = \mathcal{D}_1(K_w^i, p_2, \mathcal{I}_2, [\,], [\,], [\,], \text{true}).$$

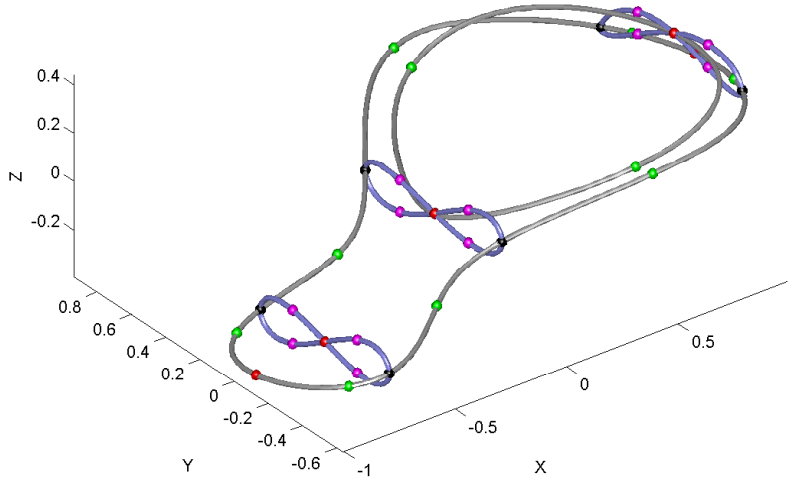

Figure 8: Slice $Z_{\mathbb{R}}$ vertically at critical values, Step 5. There are five slices: three look like figure eights, two on the far ends are just single points. Magenta balls mark the generic points of the 1-cells of the decompositions K_i of these curves.

(c) Append the vertices and edges in K_i to those of S.

At this stage, as illustrated in Figure 8, all the vertices and edges of S are complete. It remains to place a general point in each face and connect it to the edges that form the face's boundary.

6. **Slice $Z_{\mathbb{R}}$ vertically at generic values:** This step will find a general point in each face and connect it to the down and up edges, γ_0 and γ_2. For $i = 1, \ldots, m-1$, let t_i^* be the same point in (t_i, t_{i+1}) that appears in the decomposition C of $\mathcal{K}$ from Step 3. By convention, we use $t_i^* = (t_i + t_{i+1})/2$. Then for $i = 1, \ldots, m-1$, do:

   (a) Find a witness set K_w^{i*} for $Z \cap p_1^{-1}(t_i^*)$, another case of Sampling (Problem 4).

   (b) Compute the cell decomposition

   $$K^{i*} = \mathcal{D}_1(K_w^{i*}, p_2, \mathcal{I}_2, [\,], [\,], [\,], \text{true}).$$

   Figure 9 is an illustration of the running example with all the K^{i*} shown.

   (c) Each edge in K^{i*} corresponds to a face of S. By Lemma 4.7, each edge in K^{i*} either ends on an arc of the critical curve $\mathcal{K}$ or else meets a limit of $p_2^{-1}(\mathcal{I}_2)$. Either way, these edges are already in the data stored in S. We complete the face as follows. For each edge $e \in K^{i*}.\mathbf{E}$ do:

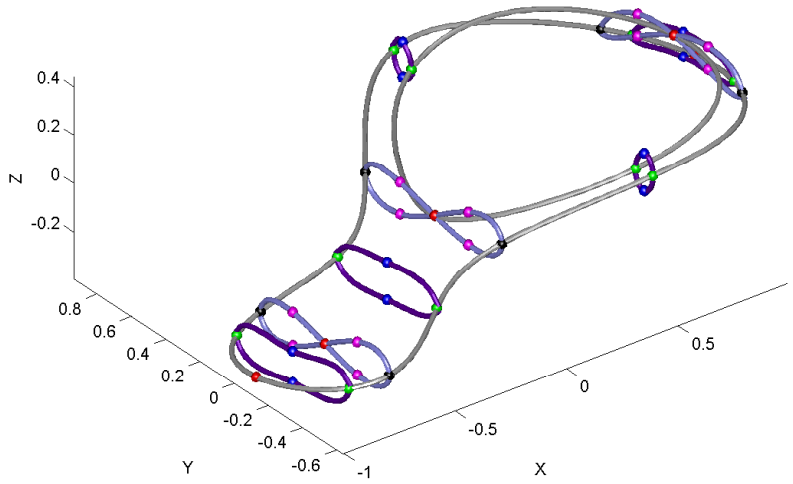

Figure 9: Slice $Z_{\mathbb{R}}$ vertically at generic values, Step 6. Blue points are the generic points of the curve decompositions, K_i^*. These points become the generic points of the 2-cells. The critical points for these curves coincide with the green generic points of the decomposition of the critical curve, see Step 6c.

i. Instantiate a new face, $\mathfrak{f}$.

ii. Set the general point of $\mathfrak{f}$ to the general point of e: $\mathfrak{f}.s^* = e.w$

iii. The left endpoint of e is the general point of some edge in S. That edge is the down boundary, γ_0, of the face. Accordingly, search through $S.\mathbf{E}$ until that edge is found. Then set $\mathfrak{f}.\gamma_0$ to the index of that edge.

iv. Similarly, search through $S.\mathbf{E}$ to find the edge whose general point is the right endpoint of e, and set $\mathfrak{f}.\gamma_2$ to its index.

v. Find the left boundary, γ_3 of face $\mathfrak{f}$. The left boundary connects the left endpoints of $\mathfrak{f}.\gamma_0$ and $\mathfrak{f}.\gamma_2$. Let w_0 and w_2 be these endpoints. Initialize $\mathfrak{f}.\gamma_3 = \emptyset$. If $w_0 = w_2$, then leave $\mathfrak{f}.\gamma_3$ empty to indicate a degenerate left edge. Otherwise, the left edge of the face consists of one or more edges from the vertical slice K^i. To find these, we work from w_0 going up, connecting edges that border $\mathfrak{f}$ end-to-end until we reach w_2, as follows:

   A. Find all edges in K^i whose left endpoint is w_0. (These edges were already added to $S.\mathbf{E}$ in Step 5.) One of these must be part of γ_3. To determine which it is, we test them sequentially as follows:

   - Let e be the edge to be tested, so $e.w$ is its general point. Let

$$v_w = [p_2(e.w) - p_2(w_0)]/[p_2(w_2) - p_2(w_0)].$$

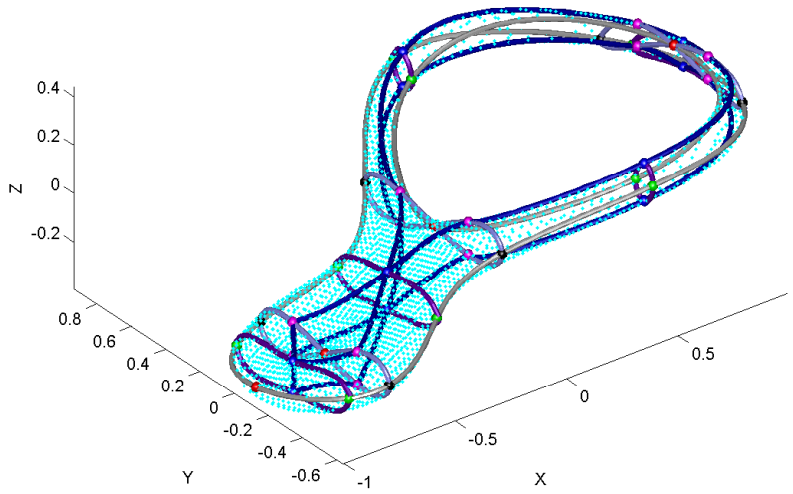

Figure 10: Find the left and right edges by connecting the generic point (blue) of each 2-cell to generic points (magenta) of the neighboring curves K_i and K_{i+1} (Step 6(c)v).

- Use the homotopy $H(x, y_0, y_2, u, v) = 0$ from (11) to move along a continuous path in $[0,1] \times [0,1]$ from $(u,v) = (1/2, 1/2)$ to $(u,v) = (0, v_w)$. A straight line path suffices. Note that the starting point for (x, y_0, y_2) in the homotopy is already known: $x = \mathfrak{f}.s^*$, $y_0 = S.\mathbf{E}(\gamma_0).w$, $y_2 = S.\mathbf{E}(\gamma_2).w$.
- If the endpoint of this homotopy is $e.w$, then we have found the next edge in γ_3. Otherwise, loop back to Step 6(c)vA to test the next edge that starts at w_0. Note that because of the inner workings of the one-dimensional algorithm, several of the edges to be tested may have the same value of $p_2(e.w)$, in which case the homotopy does not have to be recomputed.

B. Append e to $\mathfrak{f}.\gamma_3$.

C. Set w_0 to be the right endpoint of e.

D. If $w_0 = w_2$, the left boundary $\mathfrak{f}.\gamma_3$ is complete. Otherwise, we loop back to Step 6c to add another edge.

vi. Find the right boundary, γ_1 of face $\mathfrak{f}$. This follows the identical procedure as finding the left boundary except now the edges will come from K^{i+1} and consequently the homotopy tests end at $u = 1$.

vii. Append $\mathfrak{f}$ to the list of faces, $S.\mathbf{F}$.

The cell decomposition of $Z_{\mathbb{R}}$ over the patch $p_1^{-1}(\mathcal{I}_1) \times p_2^{-1}(\mathcal{I}_2)$ is complete.

Theorem 6.3 *Let $Y \subset \mathbb{C}^N$ be an almost smooth, irreducible algebraic set. For R in a nonempty Zariski-open algebraic subset of $O_{\mathbb{R}}(N)$, the above algorithm applied to $Z := R \cdot Y$ produces a cell decomposition of $Z_{\mathbb{R}} \cap p_1^{-1}(\mathcal{I}_1) \cap p_2^{-1}(\mathcal{I}_2)$.*

Proof. Lemma 4.6 implies that all of the curves decomposed in the course of the algorithm are all in general position and so the required the curve decompositions will succeed with probability one. This implies that all edges in the decomposition can be tracked from their general points to their endpoints, and in particular, the upper and lower edges of a two cell can be so tracked. For the same reason, for any face, the arc of the curve $Z \cap p_1^{-1}(1/2)$ that passes through its generic point s^* can be tracked vertically to the generic points of the lower and upper edges without encountering a singularity, except possibly when reaching the edge. By continuity, this situation persists for u in some open subset of $1/2$, and so if any singularity were to be encountered for $u \in (0,1)$ there must be a smallest value in $(1/2,1)$ or largest value in $(0,1/2)$ where this occurs. But such a value would be a critical point of $\mathcal{K}(Z,p)$ with respect to p_1. Since by construction the 2-cells only extend between neighboring critical points, there can be no real critical points of $\mathcal{K}(Z,p)$ until the left or right boundary of the cell is reached. Hence, the homotopy (11) is nonsingular for any path in $(0,1) \times (0,1)$, and so the construction meets all the requirements of a cell decomposition. □

In the running example, as shown in Figure 10, there are ten 2-cells. At this point, the homology of the surface is fully revealed by the way the 2-cells glue together across their 1-cell boundaries.

6.1 Compact surfaces

If the surface $Z_{\mathbb{R}}$ is known to be compact, then the above algorithm can be called with p_1 and p_2 being random real linear projections and omitting the intervals $\mathcal{I}_1$ and $\mathcal{I}_2$.

6.2 Noncompact Surfaces

Similar to the curve case treated in Section 5.3, if a surface is noncompact or its compactness is unknown, we can still perform a complete cell decomposition by first projectivizing the surface and then performing decomposition on cells of $\mathbb{P}^2_{\mathbb{R}}$. The projectivization process is identical to the curve case, so we do not repeat it here. So let us assume that $Z \subset \mathbb{P}^N$ is an irreducible, almost smooth, two-dimensional component of $V(F)$, and we have a witness set for it. Furthermore, let us assume that Z has already been placed into general position by applying, if necessary, a random rotation from $O(\mathbb{R}, N+1)$.

We construct the cell decomposition by working over projections onto three rectangular patches that cover all of $\mathbb{P}^2_{\mathbb{R}}$. These patches are illustrated in Figure 2. To work on these patches, we dehomogenize $F(X_0, \ldots, X_N)$ by appending a linear polynomial. For $i = 0, 1, 2$, let $f_i = \{F, X_i - 1\}$, let L_i be similar dehomogenizations of L, and let W_i be compatible mappings of the witness set W. So $\{f_i, L_i, W_i\}$ is a witness set for working on patch i.

Problem 10: Projective Surface Cell Decomposition	
Input	: A witness set $Z_w = \{F, L, W\}$ for 2-dimensional irreducible, almost smooth, algebraic set $Z \subset \mathbb{P}^N$, where F has real coefficients. Z in general position.
Output: A cell decomposition of Z, $S = \mathcal{D}_2(Z_w)$.	

1. **Decompose on patch 0:** Compute

$$S^0 = \mathcal{D}_2(\{f_0, L_0, W_0\}, \pi_1, [-1, 1], \pi_2, [-1, 1]).$$

2. **Decompose on patch 1:** Compute

$$S^1 = \mathcal{D}_2(\{f_1, L_1, W_1\}, \pi_2, [-1, 1], \pi_0, [-1, 1]).$$

3. **Decompose on patch 2:** Compute

$$S^2 = \mathcal{D}_2(\{f_2, L_2, W_2\}, \pi_0, [-1, 1], \pi_1, [-1, 1]).$$

4. **Paste cells at boundaries:** This is mainly a bookkeeping exercise, taking into account the equivalence of a point and its negative in homogeneous coordinates. The right boundary at $u_i = 1$ on patch i is the upper boundary at $v_{i+1} = 1$ on patch $(i + 1)$ (using modulus 3 arithmetic). Also, the left boundary at $u_i = -1$ on patch i is the lower boundary at $v_{i+1} = -1$ on patch $(i + 1)$. In each case, the upper and lower boundary decompositions may subdivide some edges more finely than the corresponding right and left boundary of the neighboring patch. This is because a right boundary has breakpoints only where that (locally) vertical slice has critical points, but an upper boundary has breakpoints wherever the critical curve for that patch has critical points or where it crosses the boundary. Thus, the vertices in the upper boundary include all of those for the corresponding right boundary of its neighbor plus possibly some extra ones. These extra ones need to be inserted as breakpoints in the edges for that right boundary. Each such vertex belongs to the interior of a single edge on the right boundary, which edge can be discovered by homotopy membership testing. Once the edge is found, it is split into two edges, with the inserted vertex as edge endpoints. After splitting the edge, the boundary data for the adjoining face must also be updated. A similar process pastes left boundaries to lower boundaries.

7 Discussion and Conclusions

We have described an algorithm to decompose a real surface contained in an irreducible, almost smooth, two-dimensional complex algebraic set. The decomposition consists of

2-cells, each homeomorphic to $[0,1] \times [0,1]$, bounded by edges homeomorphic to $[0,1]$. We treat the case of compact surfaces directly in $\mathbb{R}^N$, while noncompact surfaces are projectivized and treated in $\mathbb{P}^N_{\mathbb{R}}$.

Although the algorithm has been illustrated on a surface in $\mathbb{R}^3$, the technique also applies to surfaces in any higher dimensional space.

We have restricted the algorithm to complex algebraic sets that have at most a finite number of singularities, and the algorithm 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] L. Alberti and B. Mourrain. Regularity criteria for the topology of algebraic curves and surfaces. In *Mathematics of Surfaces XII*, LNCS, Volume 4647, 1–28, 2007.

[2] L. Alberti, B. Mourrain, and J.-P. Técourt. Isotopic triangulation of a real algebraic surface. *J. Symbolic Comput.*, 44(9), 1291–1310, 2009.

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

[4] D.S. Arnon, G.E. Collins, and S. McCallum. An adjacency algorithm for cylindrical algebraic decompositions of three-dimensional space. *J. Symbolic Comput*, 5(1-2), 163–187, 1988.

[5] 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.

[6] D.J. Bates, J.D. Hauenstein, C. Peterson, and A.J. Sommese. A numerical local dimension test for points on the solution set of a system of polynomial equations. *SIAM Journal on Numerical Analysis*, 47:3608–3623, 2009.

[7] D. Bates, J.D. 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.

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

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

[10] E. Berberich, P. Emeliyanenko, A. Kobel, and M. Sagraloff. Exact symbolic-numeric computation of planar algebraic curves. Preprint, 2012. Available at `arXiv:1201.1548`.

[11] E. Berberich, M. Kerber, and M. Sagraloff. An efficient algorithm for the stratification and triangulation of an algebraic surface. *Computational Geometry*, 43(3), 257–278, 2010.

[12] J.-D. Boissonnat, D. Cohen-Steiner, and G. Vegter. Isotopic implicit surface meshing. *Discrete & Computational Geometry*, 39(1), 138–157, 2008.

[13] J.S. Cheng, X.S. Gao, and M. Li. Determining the topology of real algebraic surfaces. In *Mathematics of Surfaces XI*, LNCS, Volume 3604, 121.-146, 2005.

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

[15] F. J. Drexler. Eine Methode zur Berechnung sämtlicher Lösungen von Polynomgleichungssystemen. *Numer. Math.*, 29(1):45–58, 1977.

[16] E. Fortuna, P. Gianni, P. Parenti, and C. Traverso. Computing the topology of real algebraic surfaces. In *ISSAC 02*, pp. 92–100, ACM Press, New York, 2002.

[17] C. B. Garcia and W. I. Zangwill. Finding all solutions to polynomial systems and other systems of equations. *Math. Programming*, 16(2):159–176, 1979.

[18] 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.

[19] J. Hauenstein, A.J. Sommese, and C.W. Wampler. Regeneration homotopies for solving systems of polynomials. *Mathematics of Computation*, 80:345–377, 2011.

[20] J.D. Hauenstein, A.J. Sommese, and C.W. Wampler. Regenerative cascade homotopies for solving polynomial systems. Applied Math. Comp., 218:4:1240-1246, 2011.

[21] J.D. Hauenstein and C.W. Wampler. Solving structured polynomial systems using regeneration. In preparation, 2012.

[22] J.D. Hauenstein and C.W. Wampler. Isosingular sets and deflation. In review, 2012.

[23] B. Huber and B. Sturmfels. A polyhedral method for solving sparse polynomial systems. *Math. Comp.*, *64*(212), 1541–1555, 1995

[24] T.-L. Lee, T.-Y. Li, and C.H. Tsai. HOM4PS-2.0: a software package for solving polynomial systems by the polyhedral homotopy continuation method. *Computing*, 83:109–133, 2008.

[25] T.-Y. Li. Numerical solution of polynomial systems by homotopy continuation methods. In *Handbook of numerical analysis, Vol. XI*, pages 209–304. North-Holland, Amsterdam, 2003.

[26] W.E. Lorensen and H.E. Cline. Marching cubes: a high resolution 3d surface construction algorithm. *Comput. Graph.*, 21(4), 163–170, 1987.

[27] Y. Lu, D.J. Bates, A.J. Sommese, and C.W. 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.

[28] 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.

[29] A. P. Morgan. A transformation to avoid solutions at infinity for polynomial systems. *Appl. Math. Comput.*, 18(1):77–86, 1986.

[30] A. P. Morgan and A. J. Sommese. A homotopy for solving general polynomial systems that respects m-homogeneous structures. *Appl. Math. Comput.*, 24:101–113, 1987.

[31] A. P. Morgan, A. J. Sommese, and C. W. Wampler. Computing singular solutions to polynomial systems. *Adv. in Appl. Math.*, 13(3):305–327, 1992.

[32] A. P. Morgan, A. J. Sommese, and C. W. Wampler. A power series method for computing singular solutions to nonlinear analytic systems. *Numer. Math.*, 63(3):391–409, 1992.

[33] D. Myszka, A. Murray, and C.W. Wampler, Mechanism branches, turning curves, and critical points, in review.

[34] J.-K. Seong, G. Elber, and M.-S. Kim. Contouring 1- and 2-manifolds in arbitrary dimensions. In *SMI 05*, pp. 218-227, 2005.

[35] A.J. Sommese and J. Verschelde. Numerical homotopies to compute generic points on positive dimensional algebraic sets. *J. Complexity*, 16(3):572–602, 2000.

[36] A.J. Sommese, J. Verschelde, and C.W. Wampler. Numerical decomposition of the solution sets of polynomial systems into irreducible components. *SIAM Journal on Numerical Analysis*, 38, 2022–2046, 2001.

[37] A.J. Sommese, J. Verschelde, and C.W. 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.

[38] A.J. Sommese, J. Verschelde, and C.W. Wampler. Using monodromy to decompose solution sets of polynomial systems into irreducible components. In *Applications of algebraic geometry to coding theory, physics and computation (Eilat, 2001)*, volume 36 of *NATO Sci. Ser. II Math. Phys. Chem.*, pages 297–315. Kluwer Acad. Publ., Dordrecht, 2001.

[39] A.J. Sommese, J. Verschelde, and C.W. Wampler. Symmetric functions applied to decomposing solution sets of polynomial systems. *SIAM Journal on Numerical Analysis*, 40, 2026–2046, 2001.

[40] A.J. Sommese, J. Verschelde, and C.W. Wampler. Homotopies for intersecting solution components of polynomial systems. *SIAM Journal on Numerical Analysis*, 42, 1552–1571, 2004.

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

[42] J. Verschelde. Algorithm 795: PHCpack: A general-purpose solver for polynomial systems by homotopy continuation. *ACM Trans. on Math. Software*, 25(2):251–276, 1999.

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