On computing a cell decomposition of a real surface containing infinitely many singularities

Daniel J. Bates[1], Daniel A. Brake[2], Jonathan D. Hauenstein[3],
Andrew J. Sommese[4], and Charles W. Wampler[5]

[1] Colorado State University, USA
bates@math.colostate.edu,
www.math.colostate.edu/∼bates
[2] North Carolina State University, USA
danielthebrake@gmail.com,
danielthebrake.org
[3] North Carolina State University, USA
hauenstein@ncsu.edu,
www.math.ncsu.edu/∼jdhauens
[4] University of Notre Dame, USA
sommese@nd.edu,
www.nd.edu/∼sommese
[5] General Motors Research and Development, USA
charles.w.wampler@gm.com,
www.nd.edu/∼cwample1

Abstract. Numerical algorithms for decomposing the real points of a complex curve or surface in any number of variables have been developed and implemented in the new software package `Bertini_real`. These algorithms use homotopy continuation to produce a cell decomposition. The previously existing algorithm for surfaces is restricted to the "almost smooth" case, i.e., the given surface must contain only finitely many singular points. We describe the use of isosingular deflation to remove this almost smooth condition and describe an implementation of deflation via `Bertini` with `MATLAB`.

Keywords: Real decomposition, real algebraic set, numerical algebraic geometry, isosingular deflation, homotopy continuation

1 Introduction

Polynomial systems appear throughout the sciences, engineering, and mathematics. Given a polynomial system, $f(z)$, with N polynomials in n variables, a common problem is to find all solutions $\widehat{z}$ in $\mathbb{C}$ or in $\mathbb{R}$ such that $f(\widehat{z}) = 0$, i.e., the *solution set* of $f(z) = 0$, also denoted $V(f)$. Such a solution set (for either $\mathbb{C}$ or $\mathbb{R}$) may consist of points, curves, surfaces, and/or higher-dimensional components.

In *numerical algebraic geometry*, there are now several numerical methods to produce the *numerical irreducible decomposition* over $\mathbb{C}$ of $V(f)$. It is a fundamental fact from algebraic geometry that a degree d irreducible algebraic set meets a general linear space of complementary dimension in d distinct points. For each irreducible component $A \subset V(f)$ of dimension m, the numerical irreducible decomposition contains a *witness set*, which is the triplet of $f(z)$, $L(z)$, and W, where L is a general set of m linear equations, and W consists of numerical approximations to the set of d points $A \cap V(L)$. The books [3, 10] discuss these concepts and the associated algorithms, and many of these methods are implemented in [4].

Working over $\mathbb{R}$ is significantly different than working over $\mathbb{C}$, reflecting a more complicated geometry. Real slices of real algebraic sets do not behave so uniformly as their complex counterparts, so there is no simple real analog to the witness sets that suffice when working over $\mathbb{C}$. Instead, as described below, we break the real sets into a finite number of pieces, each having a uniform behavior within. The decomposition of the real subsets within complex curves was accomplished in [9], while the decomposition of the real subsets within an adequately nice complex surface was achieved in [5]. In particular, the surface was required to be "almost smooth," meaning that it could contain at most a finite number of singularities. In this article, we remove the almost smooth condition from the surface case by incorporating *isosingular deflation* [6] into the approach of [5]. The resulting surface method and the method for curves are both implemented in the software package `Bertini_real` [2].

A fundamental problem with real solution sets of polynomial systems is the choice of a data type. We have opted for a topological description, a *cell decomposition*, of real curves and surfaces, dependent on the (typically random) choice of two linear projections. The next section provides some basic details on this data type and the previously known numerical algebraic geometry method for computing it. Section 3 illustrates the need for isosingular deflation, which is then described in §4. The inclusion of isosingular deflation is finally briefly described and illustrated in §5.

2 Cell decomposition

The cell decomposition of an algebraic surface [5] breaks it into a finite number of regions over which the implicit function theorem holds. The construction is related to Morse theory and similar in essence to the Cylindrical Algebraic Decomposition [1], although the specifics of the data structure and the algorithms for computing it are quite different. The decomposition consists of '2-cells' or *faces*, which are bounded by '1-cells' or *edges*, which are themselves bounded by vertices. Each face and edge is equipped with a generic point in the middle and a homotopy such that the generic point can be tracked along the face.

This decomposition is computed with respect to two real linear projections, $\pi_1(x)$ and $\pi_2(x)$, typically chosen randomly, which give rise to the *implicit* parameterization of the surface. Each face describes some portion of the surface with boundary either a curve over which the generic point cannot be tracked or part of an artificially imposed edge.

The process for decomposing an irreducible algebraic surface is illustrated in Fig. 1 for a surface given by the Zitrus system [7]:

$$f(x, y, z) = x^2 + z^2 + y^3(y - 1)^3. \tag{1}$$

Letting S denote the surface to be decomposed, this process is loosely given as follows. Given a witness set for S, the techniques of numerical algebraic geometry allow us to restrict all of the following computations to S, even in the case where $V(f)$ contains other irreducible components.

1. **Compute the *critical set* $\mathcal{C}$ of S with respect to π_1, π_2.** $\mathcal{C}$ consists of points $\widehat{x}$ on S that are either singular or include the direction of the projection in the tangent space at $\widehat{x}$. The points of $\mathcal{C}$ are solutions of the system
$$\begin{bmatrix} f(x) \\ \det \begin{pmatrix} Jf \\ J\pi_1 \\ J\pi_2 \end{pmatrix} \end{bmatrix} = 0,$$
where J means the Jacobian matrix of partial derivatives. The top and bottom edges of the faces in the eventual surface decomposition will be edges coming from the curve decomposition of $\mathcal{C}$. See the top left of Fig. 1, where for the particular projection we illustrate, the critical curve consists of a ring around the surface and the two singular points at its extremities. The critical curve is itself decomposed with respect to π_1.
2. **Intersect with a suitably chosen sphere.** Because the surface might be noncompact, with parts that extend to infinity, we consider only the compact part of it lying within a suitably chosen sphere. In particular, after computing the critical curve, the locations of all topologically interesting parts of the surface are known, so we may choose a sphere containing all critical points of the critical curve, and intersect it with S. In the Zitrus example, the sphere intersection curve, i.e., the intersection of S with a sphere, is empty because the surface is compact.
3. **Slice at all critical points of π_1, and halfway between.** The boundary of a face is a graph of edges of curve decompositions, the right and left of which are slices of the surface at critical points under the first projection, π_1. In contrast, the midpoint of each face is the midpoint of an edge of a *midslice*, i.e., a slice of the surface at a point halfway between two critical points under the first projection, π_1. Each slice is the intersection of the surface with a plane corresponding to fixing π_1 at a projection value, and decomposing with respect to π_2. This step is the top right in Fig. 1.
4. **Connect midpoints to build faces**. For each edge of each midslice, track its midpoint to each candidate edge of each left- and right-bounding critical slice. Using a specially crafted homotopy which couples the midpoint, top, and bottom points, as in [5], we establish the network of connections between midpoints. This step corresponds to the bottom left in Fig. 1, where each color corresponds to an individual face. After this step is complete we have a topologically correct triangulation of the surface.

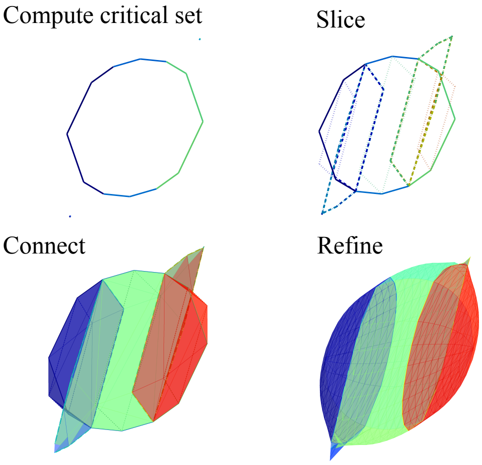

Fig. 1. Computing a cell decomposition of the Zitrus

5. **Refine and smooth**. The initially computed decomposition is rough, containing only the bare skeleton of the surface. Since each decomposition is equipped not only with a graph of connecting points, but also with a homotopy and generic point, we can refine the decomposition to obtain a more accurate geometric representation of S. The lower right figure of Fig. 1 is a moderately fine smoothing of the Zitrus.

3 Singular curves on surfaces

The Zitrus surface described in §2 is almost smooth since it has only two singular points. In the almost smooth case, the singular points are simply part of the critical set. In particular, numerical tracking does not need to be performed starting from such singular points.

In contrast, when the surface contains a curve of singularities, one needs the ability to numerically track along these singular curves to compute the cell decomposition. An example of such a curve is the “handle” of the Whitney umbrella [7], i.e., the z-axis, $x = y = 0$, in the surface implicity defined by $x^2 - y^2z = 0$. As another example, consider the Solitude surface [7] defined by the vanishing of

$$f(x, y, z) = x^2yz + xy^2 + y^3 + y^3z - x^2z^2. \tag{2}$$

There are two singular lines on this surface, one is defined by $x = y = 0$ while the other is defined by $y = z = 0$. In order to perform tracking on such singular curves, we use isosingular deflation [6], which is summarized in the next section.

4 Isosingular deflation

Deflation is a regularization procedure for an irreducible algebraic set $X \subset \mathbb{C}^N$ which produces a new polynomial system having X as an irreducible component of generic multiplicity 1. The advantage of such a polynomial system is that it facilitates numerical path tracking on X. Deflation was first introduced in the specific setting of polynomial systems in [8]. The following summarizes the more recent isosingular deflation approach of [6], depending on determinants, as is currently being used in `Bertini_real`.

Let $f : \mathbb{C}^N \to \mathbb{C}^n$ be a polynomial system and $S \subset \mathcal{V}(f) \subset \mathbb{C}^N$ be an irreducible surface of generic multiplicity 1. That is, S is an irreducible algebraic set of dimension 2 such that $\dim \operatorname{null} Jf(x) = 2$ for generic $x \in S$, where $Jf(x)$ is the Jacobian matrix of f at x. Suppose that C is an irreducible curve contained in the singular set of S, that is,

$$C \subset \{x \in S \mid \dim \operatorname{null} Jf(x) > 2\}.$$

Isosingular deflation results in a polynomial system $g(z)$ such that C is an irreducible component of the solution set of $g(z) = 0$ of generic multiplicity 1. Letting c be a generic point of curve C, isosingular deflation proceeds as follows:

```
1. Initialize $g := f$.
2. Loop until $\dim \operatorname{null} Jg(c) = 1$:
   (a) Set $r := \operatorname{rank} Jg(c)$.
   (b) Append to $g$ the $(r+1) \times (r+1)$ determinants of $Jg(x)$.
```

This loop will terminate and produce a polynomial system that can be used to perform computations on C. If the surface S was of multiplicity greater than 1, a minor modification to the stopping criterion would give a procedure for deflating S.

The following example illustrates the deflation of the singular curves on the Solitude surface.

Example 1. Let f be as in (2) and consider $C = \{(a, 0, 0) \mid a \in \mathbb{C}\}$. For simplicity of presentation, we take $c = (1, 0, 0)$. Since all first order partial derivatives of f vanish at c, $r = 0$ and we add all first partial derivatives to f, yielding

$$g(x,y,z) = \begin{bmatrix} x^2yz + xy^2 + y^3 + y^3z - x^2z^2 \\ y^2 + 2xyz - 2xz^2 \\ 2xy + x^2z + 3y^2z + 3y^2 \\ x^2y - 2x^2z + y^3 \end{bmatrix}$$

It is easy to verify that $\dim \operatorname{null} Jg(c) = 1$ so g has deflated C.

Now, we consider the other curve $C' = \{(0, 0, a) \mid a \in \mathbb{C}\}$ with $c' = (0, 0, 1)$. The first iteration of isosingular deflation again produces g as above, but since $\dim \operatorname{null} Jg(c') = 2$, we need to perform another iteration. Adding in the 2×2 determinants of $Jg(x, y, z)$ produces a polynomial system $g' : \mathbb{C}^3 \to \mathbb{C}^{22}$ such that $\dim \operatorname{null} Jg'(c') = 1$.

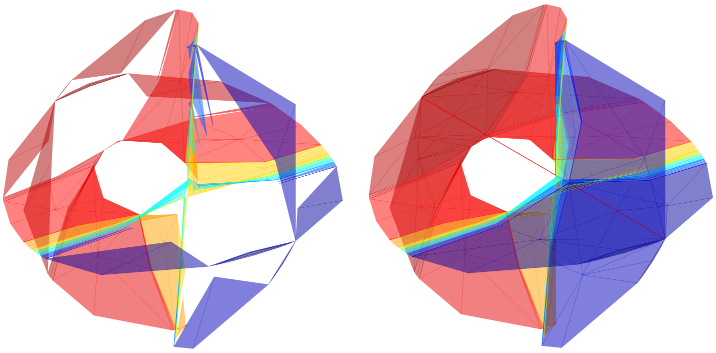

Fig. 2. Comparison of results from decomposition without deflation (left) and with deflation (right). This is the raw decomposition before refinement.

In the procedure above, the required null space dimension was known *a priori* The required determinants are computed via `MATLAB` with the rank r computed in multiple precision using `Bertini`. For deflating at points for which the corresponding dimension may not be known, we use as stopping criterion the isosingular stabilization test described in [6], as implemented in `Bertini`.

5 Decomposing surfaces

With isosingular deflation [6], we have now removed the almost smooth restriction from [5] so that this new algorithm can produce a cell decomposition of the set of real points on a complex surface regardless of the presence of singular curves on the surface. To demonstrate, Fig. 2 presents the Solitude surface defined by (2). The figure on the left shows the decomposition where the presence of the singular curves is ignored, demonstrating the failure of the decomposition method without using isosingular deflation. The figure on the right uses isosingular deflation to track along the singular curves, yielding a complete decomposition. In this figure, part of the singular line corresponding to the x-axis is isolated in that it is not an edge of any face, similar to the “handle” of the Whitney umbrella [7].

6 Conclusion

The use of isosingular deflation permits numerical path tracking to be performed on singular sets. We have applied this technique to remove the almost smooth assumption for the algorithm presented in [5] to allow one to compute a cell decomposition of the real points of any complex surface. There is no theoretical limitation on the number of variables. The drawback of using the determinantal formulation of isosingular deflation is the potentially large number of additional polynomials added to the system. We are currently exploring various approaches for limiting the number of additional polynomials needed to deflate the components of interest.

7 Acknowledgments

All authors were partially supported by AFOSR grant FA8650-13-1-7317. DJB was partially supported by NSF grant DMS-1025564. DAB and JDH were additionally supported by DARPA YFA.

References

1. D.S. Arnon, G.E. Collins, and S. McCallum. Cylindrical algebraic decomposition I: The basic algorithm. *SIAM J. Computing*, 13(4), 865–877, 1984.
2. D.A. Brake, D.J. Bates, W. Hao, J.D. Hauenstein, A.J. Sommese, and C.W. Wampler. Bertini_real: software for real algebraic sets. Available at `bertinireal.com`.
3. D.J. Bates, J.D. Hauenstein, A.J. Sommese, and C.W. Wampler. *Numerically Solving Polynomial Systems with Bertini*, SIAM, Philadelphia, 2013.
4. D.J. Bates, J.D. Hauenstein, A.J. Sommese, and C.W. Wampler. Bertini: software for numerical algebraic geometry. Available at `bertini.nd.edu`.
5. G.M. Besana, S. Di Rocco, J.D. Hauenstein, A.J. Sommese, and C.W. Wampler. Cell decomposition of almost smooth real algebraic surfaces. *Numer. Algorithms*, 63(4), 645–678, 2013.
6. J.D. Hauenstein and C.W. Wampler. Isosingular sets and deflation. *Found. Comp. Math.*, 13(3), 371–403, 2013.
7. H. Hauser and J. Schicho. “Algebraic Surfaces.” `www1-c703.uibk.ac.at/mathematik/project/bildergalerie/gallery.html`. Accessed May 19, 2014.
8. A. Leykin, J. Verschelde, and A. Zhao. Newton’s method with deflation for isolated singularities of polynomial systems. *Theor. Comp. Sci.*, 359(1–3), 111–122, 2006.
9. Y. Lu, D.J. Bates, A.J. Sommese, and C.W. Wampler. Finding all real points of a complex curve. *Contemp. Math.*, 448, 183–205, 2007.
10. A.J. Sommese and C.W. Wampler. *The Numerical Solution to Systems of Polynomials Arising in Engineering and Science*, World Scientific, Singapore, 2005.