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
jdhauens@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 with the currently employed algorithm for surfaces assuming that it is almost smooth, i.e., most finitely many singular points. The following summarizes using isosingular deflation to remove the almost smooth condition along describing the use of `Bertini` with `MATLAB` to perform the deflation.

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

1 Introduction

Introduction to cell decomposition, curves [5] and surfaces [3]. Isosingular deflation [4]. `Bertini_real` [1]. `Bertini` [2].

2 Cell decomposition

Cell decomposition of an algebraic surface [3] breaks it into a finite number of regions over which the implicit function theorem holds. 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.

The decomposition is computed with respect to two randomly chosen linear projections, $\pi_1(x), \pi_2(x)$, which give rise to the *implicit* parametrization of the surface we are computing. Each face then describes a portion of the surface, and its boundary is either a curve over which we cannot track the generic point due to the implicit function theorem, or is part of the artificially imposed edge of the view.

The process for decomposing an algebraic surface defined by system $f(x)$ is depicted in Fig. 1, and is loosely as follows:

The Zitrus surface is defined by the vanishing of

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

1. Compute the critical set with respect to π_1, π_2. This is where the surface is either singular or is tangent to the direction of projection, defined by the system:
$$\begin{bmatrix} f(x) \\ \det\begin{pmatrix} Jf \\ \pi_1 \\ \pi_2 \end{pmatrix} \end{bmatrix} = 0.$$
The edges coming from this curve decomposition will become the top and bottom edges of the faces in the end. See the top left of Fig. 1, where the ring around the surface and the points near the end are the critical curve.
2. Intersect with a suitably chosen sphere. After computing the critical curve, we know where all the interesting parts of the surface are, so choose a sphere containing all critical points of the critical curve, and intersect it with the surface. In the Zitrus example, the sphere intersection curve is empty because the surface is compact.
3. Slice at all collected critical points, and halfway between. The boundary of a face is a graph of edges of curve decompositions, the right and left of which will be slices of the surface **at critical points**. In contrast, the midpoint of each face is the midpoint of an edge of a midslice, occurring **halfway between critical points**. Each slice is the intersection of the surface with a plane corresponding to fixing π_1 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, we establish the network of connections between midpoints. This step corresponds to the bottom left in Fig. 1, where colored regions correspond to individual faces. After this step is complete we have a topologically correct triangulation of the surface.
5. Refine and smooth. The initially computed decomposition will be rough, containing only the bare skeleton of the surface, in terms of π_1, π_2. Since each decomposition is equipped not only with a graph of connecting points,

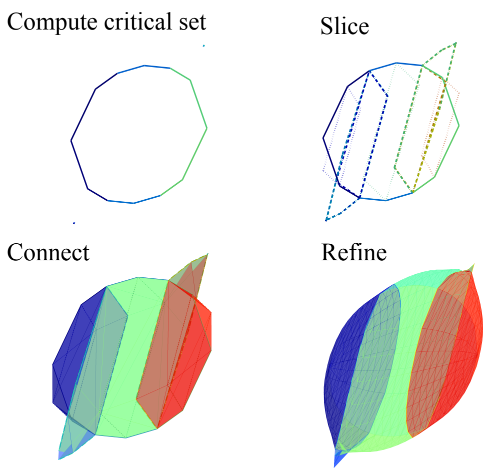

Fig. 1. Computing a cell decomposition of the Zitrus

but also with a homotopy and generic point, we can refine the decomposition arbitrarily. In the lower right of Fig. 1 is a moderately fine smoothing of the Zitrus.

3 Singular curves on surfaces

The Zitrus described in § 2 is almost smooth since it only has two singular points. In the almost smooth case, the singular points are simply added to the critical set. In particular, numerical tracking does not need to be performed starting from such singular points. However, when there is a curve of singularities, notably the "handle" of the Whitney umbrella, one needs the ability to numerically track along these curves to compute the cell decomposition.

As an example, consider the Solitude surface 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 [4] which we describe 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 the X as an irreducible component of multiplicity 1. That is, deflation produces a polynomial system which can be used to perform numerical path tracking on X. The following summarizes using the deflation routine of [4] via determinants, which 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 multiplicity 1. That is, S is an irreducible algebraic set of dimension 2 such that $\dim \text{null } Jf(x) = 2$ 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,

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

and $c \in C$ is general. Isosingular deflation constructs a polynomial system g such that C is an irreducible component of $\mathcal{V}(g)$ of multiplicity 1 as follows:

1. Initialize $g := f$.
2. Loop until $\dim \text{null } Jg(c) = 1$:
   (a) Set $r := \text{rank } Jg(c)$.
   (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 > 1, this procedure with a minor modification on the stopping criterion can be used to deflate S.

The following example considers the singular curves of 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, we need to add these derivatives to f producing

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

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

Now, we consider the other curve $C' = \{(0, 0, a) \mid a \in \mathbb{C}\}$ with $c' = (0, 0, 1)$. Since $\dim \text{null } Jg(c') = 2$, 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 $\dim \text{null } Jg'(c') = 1$.

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 using `Bertini`. For deflating at points for which the corresponding dimension may not be known, we use the isosingular stabilization test described in [4] that is implemented in `Bertini` for the stopping criterion.

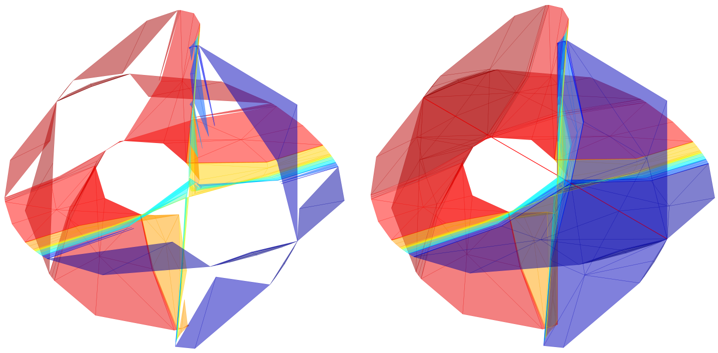

Fig. 2. Comparison of results from decomposition [left] without deflation, and [right] with deflation, to make singular curves trackable, and the decomposition complete.

5 Decomposing surfaces

With isosingular deflation [4], we have now removed the almost smooth restriction from [3] so that this new algorithm can produce a cell decomposition of the set of real points on a complex surface in any number of variables. To demonstrate, Fig. 2 presents the Solitude surface defined by (2). The one on the left shows the decomposition where are points on the singular curves are ignored demonstrating the decomposition without using isosingular deflation. The one on the right uses isosingular deflation to track along the singular curves to produce a complete decomposition.

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 [3] to allow one to compute a cell decomposition of the real points of any complex surface in any 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 of the authors were supported by AFOSR. DAB and JDH were additionally supported by DARPA YFA.

References

1. Citation for Bertini_real??
2. D.J. Bates, J.D. Hauenstein, A.J. Sommese, and C.W. Wampler. Bertini: software for numerical algebraic geometry. Available at `bertini.nd.edu`.
3. 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.
4. J.D. Hauenstein and C.W. Wampler. Isosingular sets and deflation. *Found. Comp. Math.*, 13(3), 371–403, 2013.
5. 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.