Bertini_real: Software for real algebraic sets

Daniel Brake
with
Dan Bates, Wenrui Hao, Jon Hauenstein,
Andrew Sommese, and Charles Wampler

14 October, 2014
Bertini_real Overview

Bertini_real is compiled command line software:

- performs almost purely numerical computations to produce a cellular decomposition of real algebraic components,
- uses Bertini as its homotopy continuation engine,
- uses Matlab for symbolic computations, such as deflation; as well as visualization,
- can decompose higher-dimensional curves and surfaces, including those with singularities,
- results are [almost] readily 3d printed.
Real Components

Bertini_real – Numerical Cellular Decomposition

Setup

- Let $f$ be a polynomial system with \( \mathbb{R} \) coefficients, and 
  \( f : \mathbb{C}^N \to \mathbb{C}^n \).
- Let $V(f)$ be the variety of $f$.
- Consider $C \subseteq V(f)$ be a component of dimension $k$.
- If $f$ is overdetermined, replace $f$ by a randomized version of itself.

Our objective is to decompose the real part of $C$; i.e., $C \cap \mathbb{R}^N$
Curve Cell
Curves

1. Find critical points
2. Intersect with sphere
3. Slice
4. Connect the dots
5. Merge
6. Refine

[Lu, Bates, Sommese, Wampler, 2006]
Curves

twisted cubic

\[ f(x, y, z) = \begin{bmatrix}
    y - x^2 \\
    z - x^3 \\
    y^2 - xz
\end{bmatrix} \]
Curves
Burmester 3-3 curve
Surface Cell

top

right (degenerate)

midpt

left

bottom
Surface Decomposition

1. Decompose critical curve
2. Decompose singular curves
3. Intersect with sphere
4. Slice
5. Connect the dots
6. Refine

[Besana, Di Rocco, Hauenstein, Sommese, Wampler, 2013]
Surface examples

\[ f(x, y, z) = (x^2 + y^2 + z^2 + 2^2 - \frac{1}{2})^2 - 16(x^2 + y^2) \]
Surface examples

\[
f(x, y, z) = x^2 + y^2 + z^2 + 1000(x^2 + y^2)(x^2 + z^2)(y^2 + z^2) - 1
\]
**Surface examples**

**solitude**

\[ f(x, y, z) = x^2yz + xy^2 + y^3 + y^3z - x^2z^2 \]

**klein**

\[ f(x, y, z, w) = \left[ w^2 + x^2 + y^2 + z^2 - 1 \right] \]
\[ 2wyz - x(y^2 + z^2) \]
Fivebar mechanism kinematics
Six-dimensional trigonometric system,
passed through an atan2 projection
Critical points of curves

Computing critical points of curves is easy. Since \( f \) defines a dimension-one component, the working witness set comes with one random linear:

\[
\begin{bmatrix}
  f \\
  \mathcal{L}_1
\end{bmatrix}
\]

Then we use regeneration to solve the system

\[
\begin{bmatrix}
  f \\
  \det(Jf) \\
  J_{\pi_1} \\
  \text{patch}_v
\end{bmatrix}
\]

[Hauenstein, Sommese, Wampler, 2009]
Regeneration to find crit

Regeneration uses products of linears to build up a start system for a homotopy. We’re solving by homotoping as

\[
H(x, v; t) = (1 - t) \left[ v^\top \cdot \begin{bmatrix} f(x) \\ Jf(x) \\ \text{patch}_v \end{bmatrix} \right] + t \left[ \begin{bmatrix} f(x) \\ M_1(v) \prod_{i=1}^{\delta} L_{1,i}(x) \\ \vdots \\ M_N(v) \prod_{i=1}^{\delta} L_{N,i}(x) \end{bmatrix} \right] = 0
\]

- \( \delta \) is the maximum degree any polynomial in \( f \).

We have a new result supporting this computation – a single homotopy of this form will find all isolated solutions in terms of \( x \) variables, regardless of the fiber dimension.
Building a start system

- To form the start system and solutions, we move the given random complex $\mathcal{L}(x)$ to each $L_{j,i}$ one at a time, to find the $x$ coordinates:

$$H(x; t) = (1 - t) \left[ \frac{f}{L_{j,i}} \right] + t \left[ \frac{f}{\mathcal{L}_1} \right] = 0$$

- Since only one $L_{j,i}$ vanishes for each start point, the remainder of the start functions must vanish due to $M_j(v)$, which are solved for $v$ start values by matrix inversion:

$$v = \begin{bmatrix} M_{1 \neq j} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ M_{N \neq j} & \cdots & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \vdots \\ 1 \end{bmatrix}$$

Then we take the $x$ solution from the top, and $v$ from bottom, and concatenate to form the start point.
Crit points of surfaces

Computing critical points of surfaces is hard.

- Surfaces have *critical curves*.
- Surfaces also have critical points themselves.
Current surface critical method

Using the \textit{determinantal} form of the criticality conditions:

witness set:

\[
\begin{aligned}
 f_{\text{crit\_curve}} &= \det \begin{pmatrix}
 f \\
 Jf \\
 J\pi_1 \\
 \mathcal{L}_1
\end{pmatrix} \\
 f_{\text{crit\_crit}} &= \det \begin{pmatrix}
 f \\
 Jf \\
 J\pi_1 \\
 J\pi_2
\end{pmatrix}
\end{aligned}
\]

\begin{itemize}
  \item Determinant operation produces high degree polynomials, contributing to numerical issues.
  \item Using this formulation for dimension 3 decompositions will be even worse w.r.t. degree and computational complexity.
  \item Fortunately, we can still use the nullspace method for finding critical points of this curve.
\end{itemize}
Crit points of crit curve

The actual system Bertini\_real currently solves to obtain crit points of crit curve, by regeneration:

\[
f_{\text{crit-crit}} = \begin{bmatrix}
    f \\
    \det \begin{pmatrix} Jf \\ J\pi_1 \end{pmatrix} \\
    v^T \cdot J \\
    \det \begin{pmatrix} Jf \\ J\pi_1 \end{pmatrix} \\
    \text{patch}_v
\end{bmatrix}
\]
3D Printing

1. Run Bertini

2. Run Bertini_real

3. Refine

4. Process into .stl

5. Thicken surface

6. Print

<table>
<thead>
<tr>
<th>dimension</th>
<th>components</th>
<th>classified</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>

***** Witness Set Decomposition *****

****** Decomposition by Degree ******

Dimension 2: 1 classified component
degree 4: 1 component
Thank you for your kind attention.