A Study In Multistability, and Criticality of Real Algebraic Sets

Daniel Brake

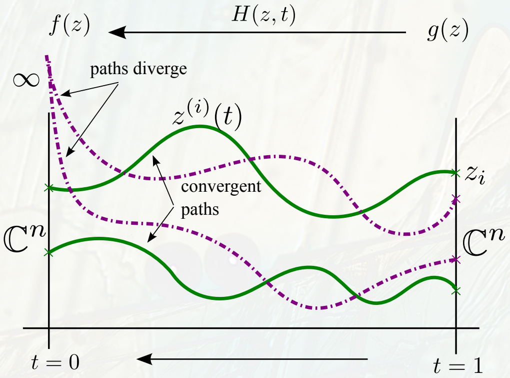
01 April, 2014

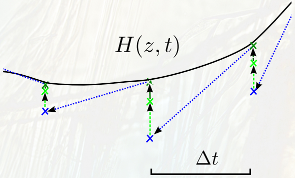
Collaborators

- Paramotopy: Matt Niemerg (CSU), Dan Bates (CSU).
- Multistability: Dan Bates, Chris Nam (Swarthmore), Benjamin Gori (Singapore), Jeremy Gunawadena (Harvard Medical).
- Bertini_Real: Dan Bates, Jon Hauenstein (NCSU), Wenrui Hao (MBI), Charles Wampler (GM), Andrew Sommese (Notre Dame).

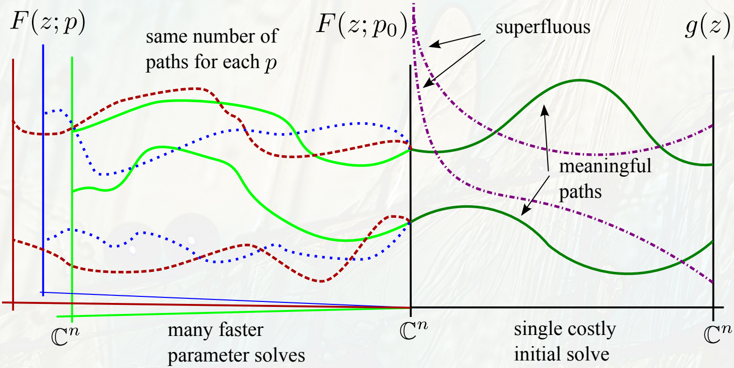
We thank our funding agencies, including the NSF and DARPA.

Homotopy Methods





Parameter Homotopy



Paramotopy

- Solves parameterized square systems of polynomials for arbitrary sets of parameter values.
- Re-solution of failed paths.
- Tunable system-specific settings.
- Uses `Bertini` to perform solves, under the hood.
- Numeric-menu driven interface which simplifies and automates Bertini functions.
- Open source, free. `www.paramotopy.com`

```
*******************************
 Welcome to Paramotopy.
     parametrized system analysis software by
     daniel brake and matthew niemerg, with dan bates.

     www.paramotopy.com     danielthebrake@gmail.com  matthew.niemerg@gmail.com
**************************
```

Demo system

$$
\begin{aligned}
x_0 + x_1 + x_2 + y_1 + y_2 + y_3 + y_4 &= k_1, \\
e + y_1 + y_2 &= k_2, \\
f + y_3 + y_4 &= k_3, \\
-a_1 x_0 e + b_1 y_1 + c_4 y_4 &= 0, \\
c_2 y_2 - a_3 x_2 f + b_3 y_3 &= 0, \\
a_1 x_0 e - (b_1 + c_1) y_1 &= 0, \\
a_2 x_1 e - (b_2 + c_2) y_2 &= 0, \\
a_3 x_2 f - (b_3 + c_3) y_3 &= 0, \\
a_4 x_1 f - (b_4 + c_4) y_4 &= 0.
\end{aligned}
$$

The system has 9 variables, 9 equations, and 15 parameters. The parameters in the problem are a_i, b_i, c_i, k_j, for $i \in \{1, 2, 3, 4\}$, and $j \in \{1, 2, 3\}$.

Starting Question

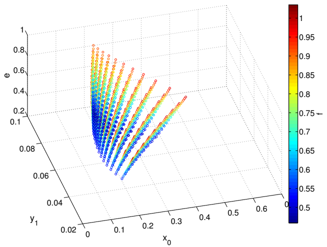
[The parameters] are all assumed to be positive. For some of them, the system has only one positive solution (i.e., all coordinates are real and positive) and for the others, it has three. How can numerical methods help us determining values, regions, etc., for this to happen?

Initial answer

To start answering questions about this system:

- Perform grid-sample solves across the unit cube $[0, 1]^{15}$, to find the number of solutions

  Found only points with a single positive real equilibrium.

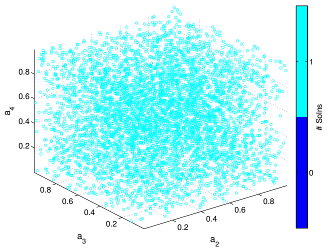


- Perform positive-dimensional numerical irreducible decomposition for random point in parameter space

  $\Rightarrow$ Only solutions are 'isolated' solutions – just points.

Monte Carlo sampling

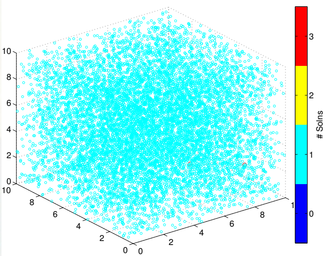
Grid-sampling the unit cube provided only single-equilibria points – maybe Monte Carlo sampling will help? Randomly sampling 10^5 times in all parameters yielded the same picture:



Monte carlo sampling of the unit cube seemed to lend itself to the conjecture that this portion of parameter space is devoid of points at which multiple positive equilibria exist. Perhaps the unit cube has only unique equilibria?

Beyond the Unit Cube

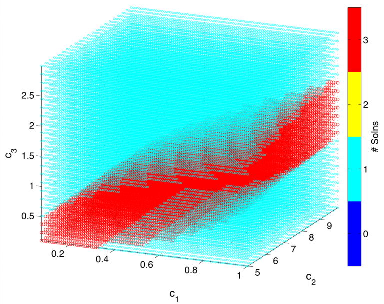
Let's move out 10 units – the 10-cube, $[0, 10]^{15}$
Randomly sampling can be quite fruitful for system with large numbers of parameters, with estimates on numbers (such as volumes) converging as the square root.



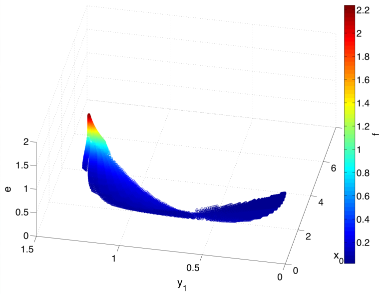
My first pass at sampling, using only 10^4 points, produced *two* parameter points with non-unique equilibria. See the red points?

Structure Emerges

Having seen a separation of the number of posreal equilibria depending on what looks like c_1, c_3, I chose to run a mesh around the point of interest.



Number of solutions - vs parameter point.



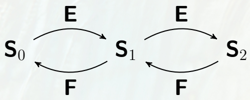
Solution set structure.

Summary

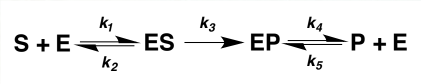
How can numerical methods help investigate a dynamical system?

- Provide numerical description of equilibria.
- Visualize to gain intuition.
- Search for anomalous points.
- Fix some parameters, vary others.
- Break down solution set in terms of dimensions and components.

A: SYSTEM SCHEMATIC



B: ENZYMOLOGY



C: POLYNOMIAL SYSTEM

$$
\begin{aligned}
0 &= \alpha v^2 + \zeta uv + \beta\zeta^2 u^2 + (\alpha\epsilon_0 - \alpha\epsilon_0\sigma - \alpha + \phi_1\zeta)\, uv^2 + \left(\epsilon_1\zeta - \epsilon_1\zeta\sigma - \zeta + \beta\phi_2\zeta^2\right) u^2 v \\
&\quad + \left(\beta\epsilon_2\zeta^2 - \beta\epsilon_2\zeta^2\sigma - \beta\zeta^2\right) u^3 + \alpha\phi_0 v^3 - (\alpha\epsilon_0 + \phi_1\zeta)\, u^2 v^2 - \left(\epsilon_1\zeta + \beta\phi_2\zeta^2\right) u^3 v \\
&\quad - \beta\epsilon_2\zeta^2 u^4 - \alpha\phi_0 u v^3 \\
0 &= \alpha v^2 + \zeta uv + \beta\zeta^2 u^2 + (\alpha\phi_0 - \alpha\phi_0\lambda - \alpha)\, v^3 + (\phi_1\zeta - \phi_1\zeta\lambda - \zeta + \alpha\epsilon_0)\, uv^2 \\
&\quad + \left(\beta\phi_2\zeta^2 - \beta\phi_2\zeta^2\lambda - \beta\zeta^2 + \epsilon_1\zeta\right) u^2 v + \beta\epsilon_2\zeta^2 u^3 - (\alpha\epsilon_0 + \phi_1\zeta)\, uv^3 - \left(\epsilon_1\zeta + \beta\phi_2\zeta^2\right) u^2 v^2 \\
&\quad - \beta\epsilon_2\zeta^2 u^3 v - \alpha\phi_0 v^4
\end{aligned}
$$

VARIABLES

$$u = \frac{[E]}{E_{\text{tot}}} \qquad v = \frac{[F]}{F_{\text{tot}}}$$

CONSERVED QUANTITIES

$$\sigma = \frac{S_{\text{tot}}}{E_{\text{tot}}} \qquad \lambda = \frac{S_{\text{tot}}}{F_{\text{tot}}} \qquad \zeta = \frac{E_{\text{tot}}}{F_{\text{tot}}}$$

PARAMETERS

$$\alpha = \frac{c^E_{1,0} - c^E_0}{c^F_0 - c^F_{1,0}} \quad \beta = \frac{c^E_2 - c^E_{1,2}}{c^F_{1,2} - c^F_2} \quad \epsilon_0 = \frac{E_{\text{tot}}}{K^E_0} \quad \epsilon_1 = \frac{E_{\text{tot}}}{K^E_1} \quad \epsilon_2 = \frac{E_{\text{tot}}}{K^E_2} \quad \phi_0 = \frac{F_{\text{tot}}}{K^F_0} \quad \phi_1 = \frac{F_{\text{tot}}}{K^F_1} \quad \phi_2 = \frac{F_{\text{tot}}}{K^F_2}$$

pic from Chris Nam, Swarthmore

Phosphorylation ctd.

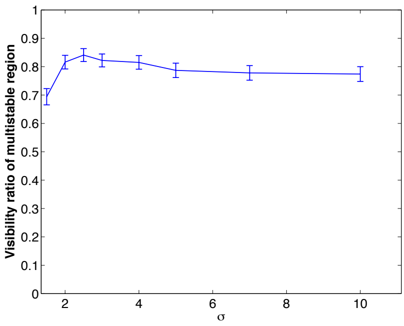
System f,

- 2 equations, of degree 4.
- 2 variables,
- 8 parameters.
- Care only about *posreal* solutions.

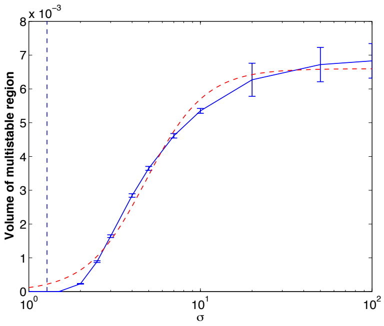
Questions about the parameter space:

- Is there a region of *multistability*?
- What is its measure? Does it saturate?
- Is it connected?
- Is it convex?

Phosphorylation ctd.



convexity as function of σ.



volume as function of σ.

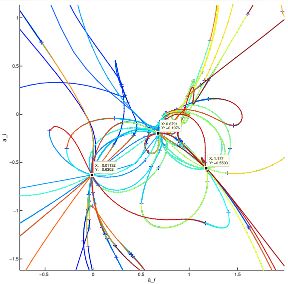
switching gears

Real Components

`bertini_real` – Numerical Cellular Decomposition

Setup

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



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

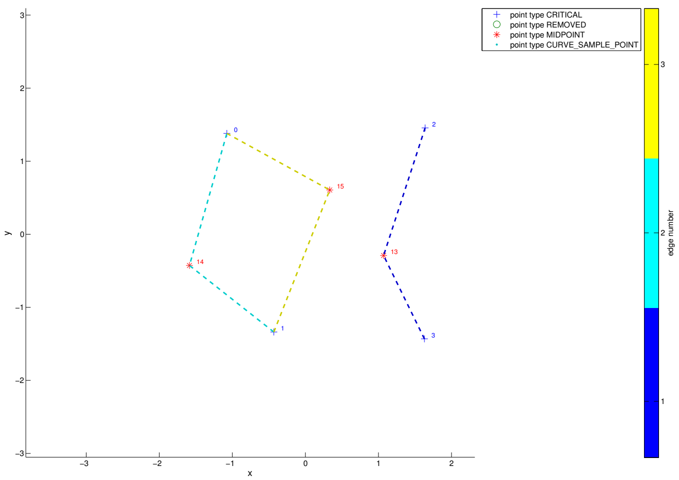
Curve Decomposition

Input: polynomial system f defining curve, witness set W, projection π.
Output: cell decomposition D_{curve}.

1: Compute Critical Points w.r.t. π (including singular points).
2: If not supplied, compute sphere parameters.
3: Compute Sphere Intersection.
4: Slice between all critical points to obtain midpoints.
5: Connect midpoints to critpoints, or to 'new' points.
$\Rightarrow$ each midpoint pairs with two critpoints to form an *edge*.
6: If desired, *merge* away the 'new' points.

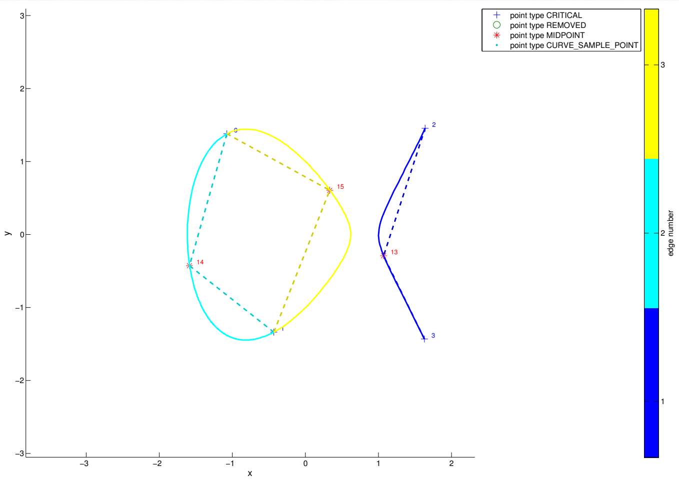
[Lu,Bates,Sommese,Wampler,2006]

Curves



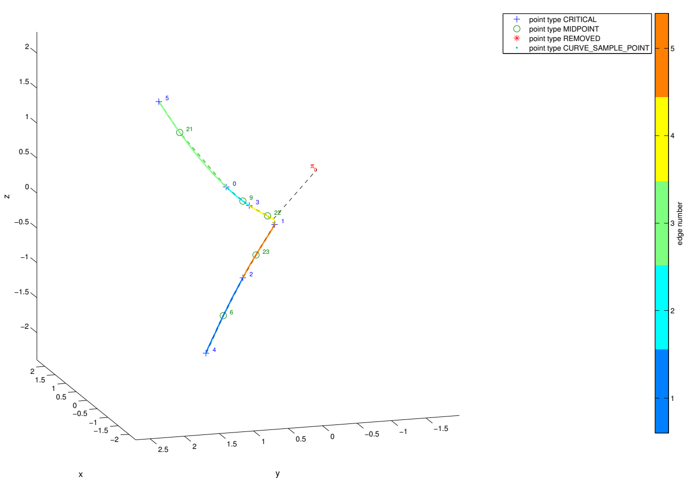
elliptic curve $f(x,y) = x^3 - 2x + 1 - y^2$

Curves



elliptic curve $f(x, y) = x^3 - 2x + 1 - y^2$

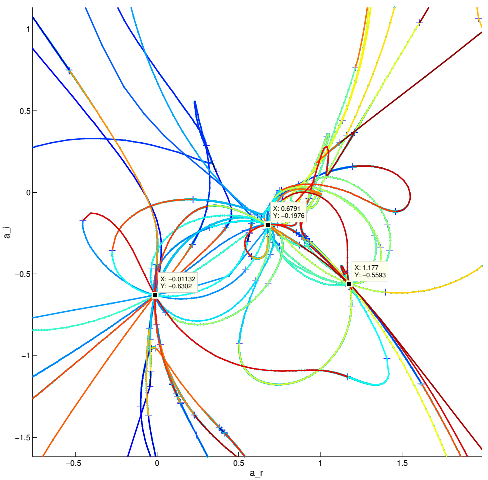
Curves



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 Decomposition

Input: polynomial system f defining surface, witness set W, projections π_1, π_2.
Output: cell decomposition D_{surf}.

1: Compute witness set for critical curve.
2: Separate points for singular curves, & deflate.
3: Compute critical points of:
- Critical curve,
- Singular curves.

4: If not provided, compute sphere parameters.
5: Obtain witness set for sphere curve, and critical points.
6: Merge together all crit points into χ.

Surface Decomposition (ctd)

Continued from prev slide.

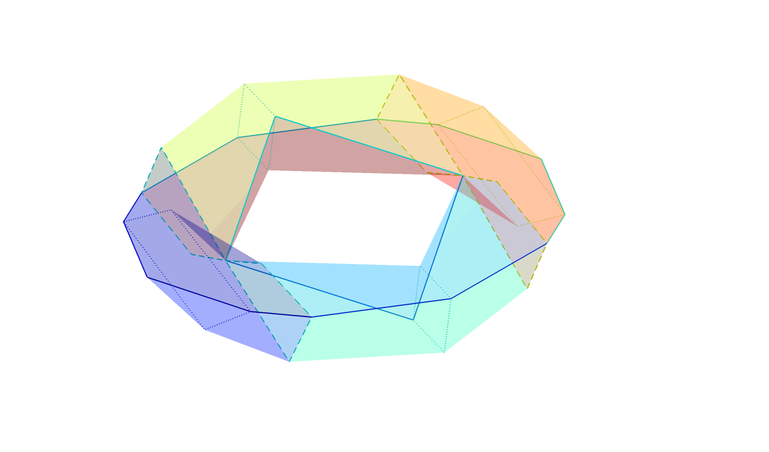
7: Using witness sets and precomputed χ, decompose each of:

- Critical curve,
- Singular curves,
- Sphere curve.

8: At each critical point in χ, perform a slice, and curve-decompose.
9: Between each point in χ, slice and curve decompose.
10: For each edge in each midslice, connect midpoints of edges to form faces.
$\Rightarrow$ Each midpoint of each edge in each midslice becomes the center point of a *face*.

[Besana,Di Rocco,Hauenstein,Sommese,Wampler, 2012]

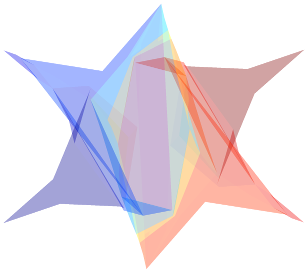
Surfaces



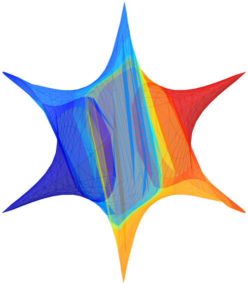
torus

$$f(x,y,z) = (x^2 + y^2 + z^2 + 2^2 - {\frac{1}{2}}^2)^2 - 16(x^2 + y^2)$$

Surfaces



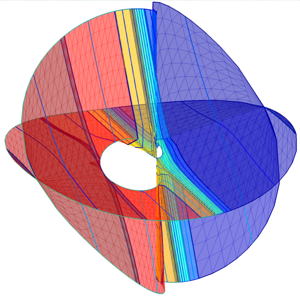
distel, unrefined



distel, refined

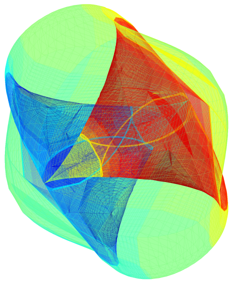
$$f(x,y,z) = x^2 + y^2 + z^2 + 1000(x^2+y^2)(x^2+z^2)(y^2+z^2) - 1$$

Surfaces



solitude

$$f(x,y,z) = x^2yz + xy^2 + y^3 + y^3z - x^2z^2$$



klein

$$f(x,y,z,w) = \begin{bmatrix} w^2 + x^2 + y^2 + z^2 - 1 \\ 2wyz - x(y^2+z^2) \end{bmatrix}$$

Critical point computations

Computing critical points of curves is easy.

Since f defines a curve, its witness set comes with one random linear:

$$\begin{bmatrix} f \\ \mathcal{L}_1 \end{bmatrix}$$

Then we use regeneration to solve the system:

$$\begin{bmatrix} f \\ v^{\mathsf{T}} \cdot \begin{bmatrix} Jf \\ \pi_1 \end{bmatrix} \\ \text{patch}_v \end{bmatrix}$$

[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 homotopoing as

$$H(x,v;t) = (1-t)\begin{bmatrix} f(x) \\ v^{\intercal} \cdot \begin{bmatrix} Jf(x) \\ \pi_1 \end{bmatrix} \\ \mathrm{patch}_v \end{bmatrix} + t \begin{bmatrix} f(x) \\ M_1(v)\prod_{ii=1}^{\delta} L_{1,ii}(x) \\ \vdots \\ M_N(v)\prod_{ii=1}^{\delta} L_{N,ii}(x) \\ \mathrm{patch}_v \end{bmatrix}$$

- δ is the maximum degree any polynomial in f.

Building a start system

- To form the start system, we move the given random complex $\mathcal{L}(x)$ to each $L_{jj,ii}$, one at a time:

$$H(x;t) = (1-t)\begin{bmatrix} f \\ L_{jj,ii} \end{bmatrix} + t\begin{bmatrix} f \\ \mathcal{L}_1 \end{bmatrix}$$

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

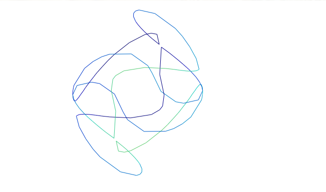
$$v = \begin{bmatrix} M_{1\neq jj} \\ \vdots \\ M_{N\neq jj} \\ \text{patch}_v \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

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.



What is functional right now

So how do I compute critical points of the critical curve now?

Using the *determinantal* form of the criticality conditions:

$$f_{\text{crit\_curve}} = \begin{bmatrix} f \\ \det\begin{pmatrix} Jf \\ \pi_1 \\ \pi_2 \end{pmatrix} \\ \mathcal{L}_1 \end{bmatrix}$$

- Determinant condition could be of high degree, causing numerical problems.
- Using this formulation for dimension 3 decompositions will be even worse with degree.
- Fortunately, we can still use the nullspace method for finding critical points of this curve.

What is functional right now

So how do I compute critical points of the critical curve now?

Using the *determinantal* form of the criticality conditions:

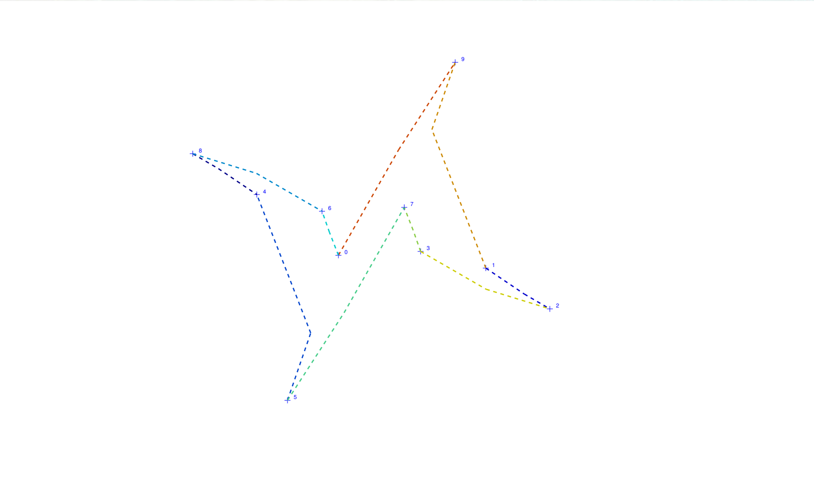
$$f_{\text{crit\_curve}} = \begin{bmatrix} f \\ \det \begin{pmatrix} Jf \\ \pi_1 \\ \pi_2 \end{pmatrix} \\ \mathcal{L}_1 \end{bmatrix}$$

- Determinant condition could be of high degree, causing numerical problems.
- Using this formulation for dimension 3 decompositions will be even worse with degree.
- Fortunately, we can still use the nullspace method for finding critical points of this curve.

Crit points of crit curve

The actual system I currently solve to obtain crit points of crit curve:

$$f_{\text{crit\_crit}} = \begin{bmatrix} f \\ \det\begin{pmatrix} Jf \\ \pi_1 \\ \pi_2 \end{pmatrix} \\ v^{\intercal} \cdot J \begin{pmatrix} f \\ \det\begin{pmatrix} Jf \\ \pi_1 \\ \pi_2 \end{pmatrix} \\ \pi_1 \end{pmatrix} \\ \text{patch}_v \end{bmatrix}$$



Thank you for your kind attention.

Generic Decomposition

Input: polynomial system f, witness set W, projections $\pi_1, ...\pi_d$.
Output: cell decomposition D.

1: Compute W_K, a witness set for the critical set of dimension d.
2: Compute witness sets for singular sets, sphere intersection.
3: Compute critical points of everything possible.
4: Decompose S, the singular set.
5: Decompose K, the critical set.
6: Intersect with sphere.
7: Slice using linears at all critical points.
8: Slice between all critical points.
9: Glue it all together.

Method not implemented or written up.