

Parametrized Polynomial Systems, and Real Numerical Algebraic Geometry

Daniel Brake

19 March, 2015

Collaborators

- ▶ Curves and surfaces: Dan Bates (Colorado State), Jon Hauenstein (Notre Dame), Wenrui Hao (MBI), Charles Wampler (General Motors), Andrew Sommese (Notre Dame).
- ▶ Parameterized Polynomial Systems: Matt Niemerg, Dan Bates.
- ▶ Multistability: Dan Bates, Chris Nam (Swarthmore), Benjamin Gori (Singapore), Jeremy Gunawadena (Harvard Medical).

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

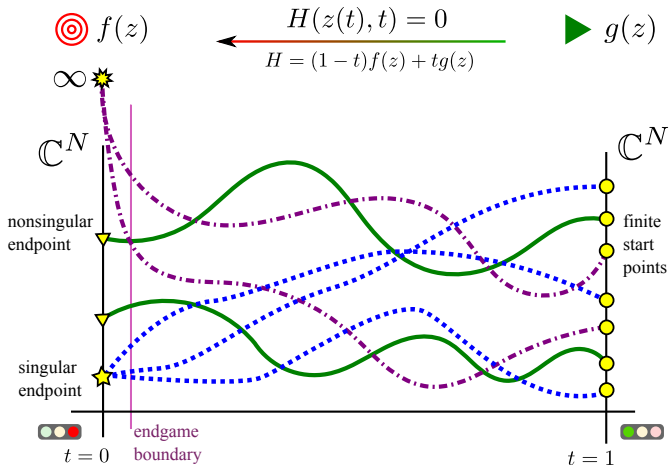
Table of Contents

Homotopy Continuation

Real Curves and Surfaces

Parameterized Polynomial Systems

Homotopy Methods



Predictor-corrector method

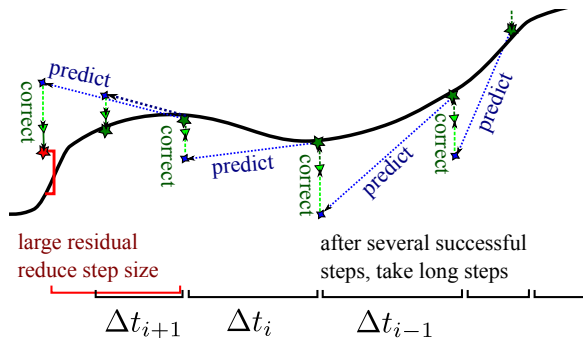


Table of Contents

Homotopy Continuation

Real Curves and Surfaces

Parameterized Polynomial Systems

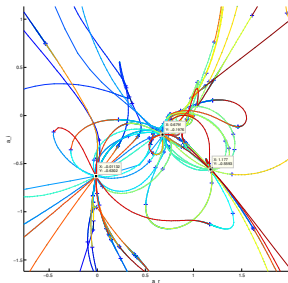
Real Components

Setup

- ▶ Let f be a polynomial system with $\mathbb{R}$ coefficients, with $f : \mathbb{C}^N \rightarrow \mathbb{C}^n$.
- ▶ 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., $C \cap \mathbb{R}^k$

bertini_real: software for real algebraic sets
www.bertiniereal.com



Choose a projection

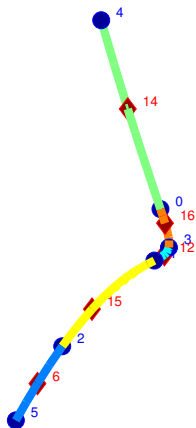
For real decompositions, choose a projection $\pi(x)$.

- ▶ π is a real linear projection map,
- ▶ π projects onto as many 'coordinates' as there are dimensions in the object,
- ▶ for best results, π is chosen randomly and orthogonally.

The projection values $\pi(x)$ will be the implicit parameters for the decomposition.

Essentially, choose a direction from which to view the object.

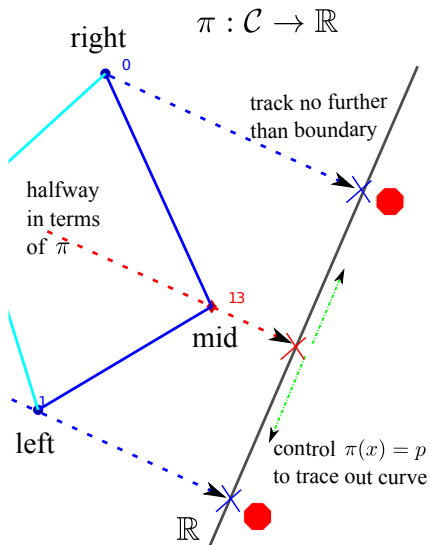
Curves



twisted cubic

$$f(x, y, z) = \begin{bmatrix} y - x^2 \\ z - x^3 \\ y^2 - xz \end{bmatrix}$$

Curve Cell



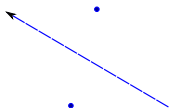
an *edge* consists of:

- ▶ midpoint,
- ▶ two endpoints,
- ▶ a homotopy:

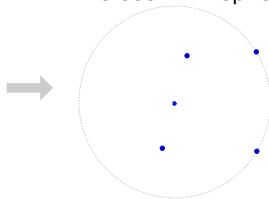
$$\begin{bmatrix} f_{\text{curve}}(x) \\ \pi(x) - p \end{bmatrix} = 0$$

Curves

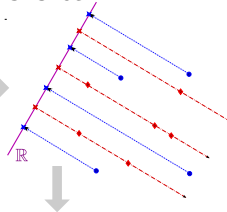
1. Find critical points



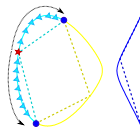
2. Intersect with sphere



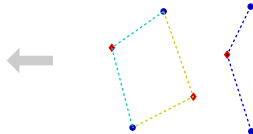
3. Slice



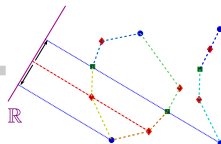
6. Refine



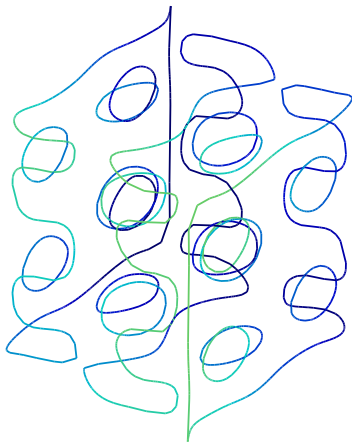
5. Merge



4. Connect the dots

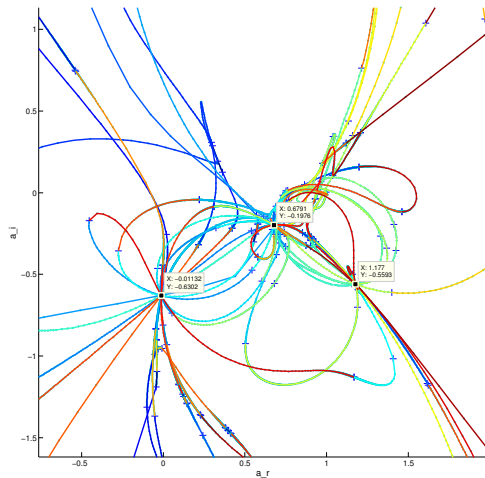


Curves



Curve on a Banchoff surface.

Curves



Burmester 3-3 curve

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:

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

Regeneration to find crit

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

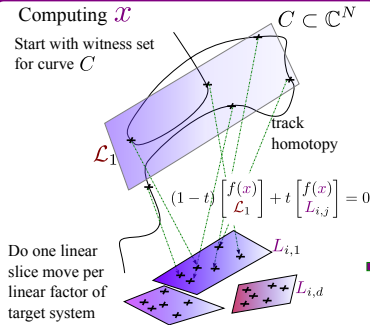
$$H(x, v; t) = (1 - t) \begin{bmatrix} f(x) \\ v^T \cdot \begin{bmatrix} Jf(x) \\ \pi_1 \end{bmatrix} \\ \text{patch}_v \end{bmatrix} + t \begin{bmatrix} f(x) \\ M_1(v) \prod_{ii=1}^{\delta} L_{1,ii}(x) \\ \vdots \\ M_N(v) \prod_{ii=1}^{\delta} L_{N,ii}(x) \\ \text{patch}_v \end{bmatrix}$$

- ▶ This is the *left nullspace* formulation of the critical system.
- ▶ Correspondingly, δ is the maximum degree any polynomial in f .

Curve Critical Points

Computing $\mathcal{X}$

Start with witness set
for curve C



Computing $\mathcal{V}$

$$v = \begin{bmatrix} M_1 \\ \vdots \\ M_{k \neq i} \\ \vdots \\ M_N \\ \text{patch}_v \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

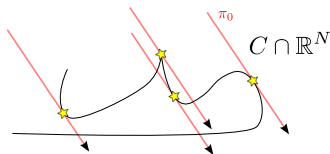
One $L_{i,d}$ vanishes on
each start point in $\mathcal{X}$.

Other functions vanish
on $\mathcal{V}$ variables.
Find by inverting M .

$$\begin{bmatrix} f(x) \\ M_1(v) \prod_{i=1}^d L_{1,i}(x) \\ \vdots \\ M_N(v) \prod_{i=1}^d L_{N,i}(x) \\ \text{patch}_v(v) \end{bmatrix} = 0$$

Multiply
linear
factors

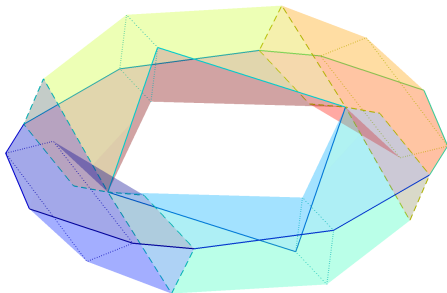
Track homotopy



$$\begin{bmatrix} f(x) \\ Jf(x) \\ J\pi_0 \\ \text{patch}_v(v) \end{bmatrix} = \begin{bmatrix} f(x) \\ Jf(x) \\ J\pi_0 \end{bmatrix} = 0$$

Result: real critical points $\star$
of complex curve C , with respect
to the linear projection π_0

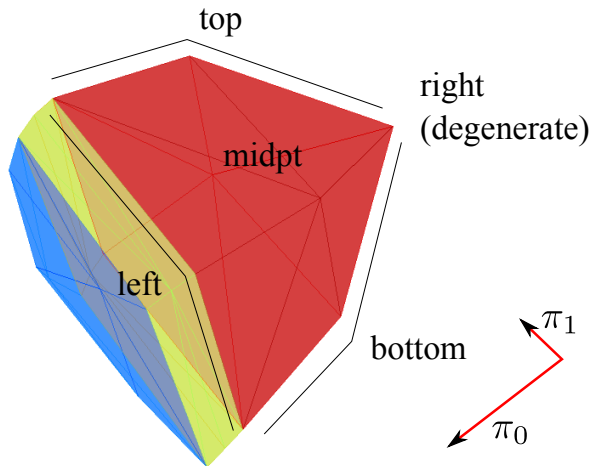
Surfaces



torus

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

Surface Cell



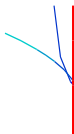
The 'top', 'bottom', 'left', and 'right' boundaries are edges from curve decompositions.

Surface Decomposition

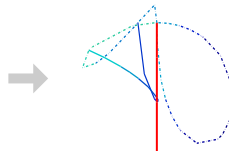
1. Decompose critical curve



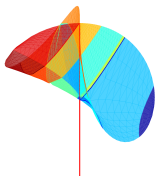
2. Decompose singular curves



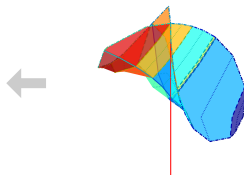
3. Intersect with sphere



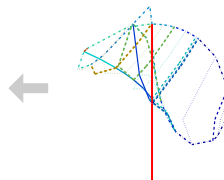
6. Refine



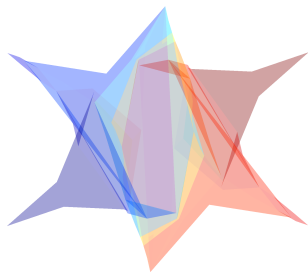
5. Connect the dots



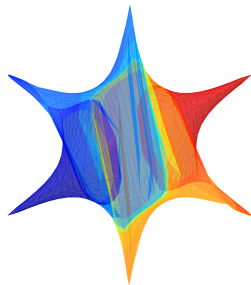
4. Slice



Surfaces



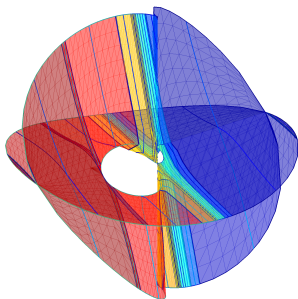
distel, unrefined



distel, refined

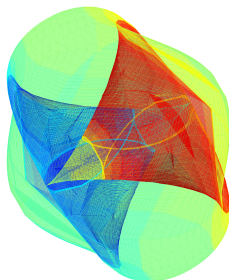
$$f(x, y, z) = x^2 + y^2 + z^2 + 1000(x^2 + y^2)(x^2 + z^2)(y^2 + z^2) - 1$$

Surfaces



solitude

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



klein

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

Crit points of crit curve

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

$$f_{\text{crit_crit}} = \left[\begin{array}{c} f \\ \det \left(\begin{array}{c} Jf \\ \pi_1 \\ \pi_2 \end{array} \right) \\ v^T \cdot J \left(\begin{array}{c} f \\ \det \left(\begin{array}{c} Jf \\ \pi_1 \\ \pi_2 \end{array} \right) \\ \pi_1 \end{array} \right) \\ \text{patch}_v \end{array} \right]$$

3D Printing

1. Run Bertini

Compute $\mathbb{C}$ witness points

```
***** Witness Set Decomposition *****
| dimension | components | classified | unclassified |
|-----|-----|-----|-----|
| 2 | 1 | 6 | 0 |
***** Decomposition by Degree *****
Dimension 2: 1 classified component
degree 6: 1 component
```

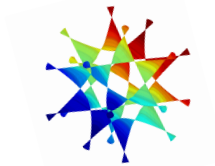
2. Run Bertini_real

Solve many $\mathbb{R}$
polynomial systems



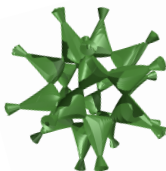
3. Refine

Smooth blocky regions
face-by-face



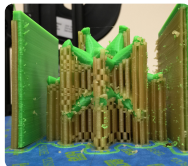
4. Post-process

Solidify, save as .stl



5. Print

Ninjabflex, PVA support



6. Remove support

Warm water bath



Table of Contents

Homotopy Continuation

Real Curves and Surfaces

Parameterized Polynomial Systems

Parameterized Polynomials

Some polynomial systems are parameter-free:

- ▶ $x^2 + \sqrt{2}x - 1$
- ▶ $x^4 - 3x^2y^2 + 3y^4$

A single variety is admitted for the system. Study it once, and you're done.

Others are parameterized:

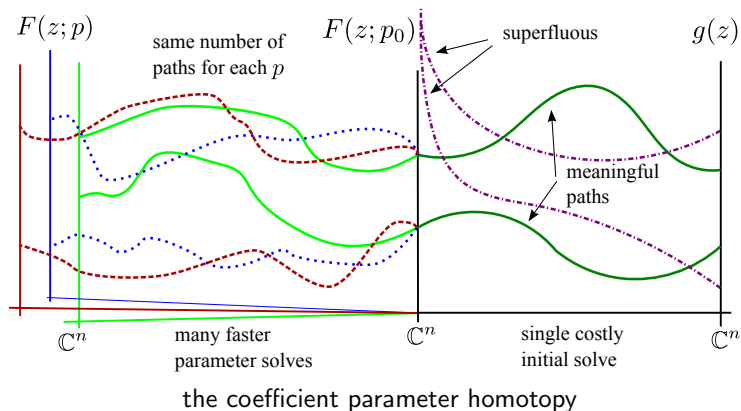
$$\begin{aligned}
 & - a_{00}s_0e + b_{00}x_0 + c_{01}y_1 \\
 & - a_{10}s_1e + b_{10}x_1 + c_{00}x_0 - a_{01}s_1f + b_{01}y_1 + c_{11}y_2 \\
 & - a_{20}s_2e + b_{20}x_2 + c_{10}x_1 - a_{11}s_2f + b_{11}y_2 + c_{21}y_3 \\
 & - a_{30}s_3e + b_{30}x_3 + c_{20}x_2 - a_{21}s_3f + b_{21}y_3 + c_{31}y_4 \\
 & a_{00}s_0e - (b_{00} + c_{00})x_0 \\
 & a_{10}s_1e - (b_{10} + c_{10})x_1 \\
 & a_{20}s_2e - (b_{20} + c_{20})x_2 \\
 & a_{30}s_3e - (b_{30} + c_{30})x_3 \\
 & a_{01}s_1f - (b_{01} + c_{01})y_1 \\
 & a_{11}s_2f - (b_{11} + c_{11})y_2 \\
 & a_{21}s_3f - (b_{21} + c_{21})y_3 \\
 & a_{31}s_4f - (b_{31} + c_{31})y_4 \\
 & k_1 - s_0 + s_1 + s_2 + s_3 + s_4 + \dots \\
 & \quad x_0 + x_1 + x_2 + x_3 + y_1 + y_2 + y_3 + y_4 \\
 & k_2 - e + x_0 + x_1 + x_2 + x_3 \\
 & k_3 - f + y_1 + y_2 + y_3 + y_4
 \end{aligned}$$

Want to study the behaviour of system in terms of a specific set of parameter values.

Tools for parameterized systems

- ▶ Algebraic description -
 - ▶ Gröbner bases, using Macaulay2, or other symbolic packages.
Perform elimination on variables.
 - ▶ Cylindrical Algebraic Decomposition.
Quantifier elimination.
- ▶ Geometric description -
 - ▶ Computing witness sets, using Bertini, or other numerical packages.
Numerical Irreducible Decomposition.
 - ▶ Solve problem defining boundaries between 'cells' in parameter space.
Compute the discriminant locus.
 - ▶ Sampling methods.
Discretize parameter space, and repeatedly solve.

Parameter Homotopy - Numerical Sampling Method



Paramotopy

- ▶ Solves parameterized square systems of polynomials for arbitrary sets of parameter values.
- ▶ Re-resolution 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

$$\begin{aligned}x_0 + x_1 + x_2 + y_1 + y_2 + y_3 + y_4 &= k_1, \\e + y_1 + y_2 &= k_2, \\f + y_3 + y_4 &= k_3, \\-a_1x_0e + b_1y_1 + c_4y_4 &= 0, \\c_2y_2 - a_3x_2f + b_3y_3 &= 0, \\a_1x_0e - (b_1 + c_1)y_1 &= 0, \\a_2x_1e - (b_2 + c_2)y_2 &= 0, \\a_3x_2f - (b_3 + c_3)y_3 &= 0, \\a_4x_1f - (b_4 + c_4)y_4 &= 0.\end{aligned}$$

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

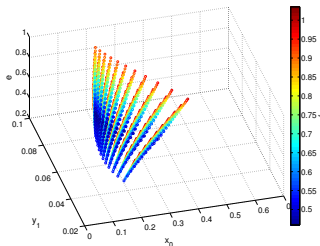
[The parameters] are all assumed to be positive. For some of them, the system has only one positive solution (i.e., all coordinates are real and positive) and for the others, it has three. How can numerical methods help us determining values, regions, etc., for this to happen?

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

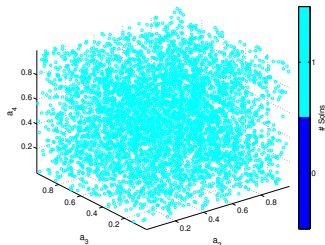
Found only points with a single positive real equilibrium.



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

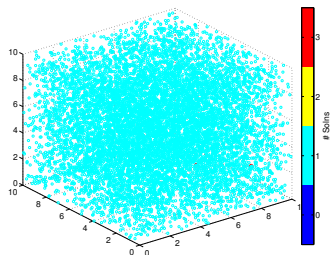


Monte carlo sampling of the unit cube seemed to lend itself to the conjecture that this portion of parameter space is devoid of points at which multiple positive equilibria exist. Perhaps the unit cube has only unique equilibria?

Beyond the Unit Cube

Let's move out 10 units – the 10-cube, $[0, 10]^{15}$

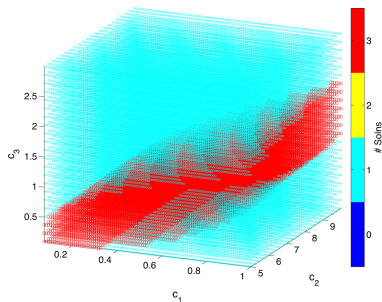
Randomly sampling can be quite fruitful for system with large numbers of parameters, with estimates on numbers (such as volumes) converging as the square root.



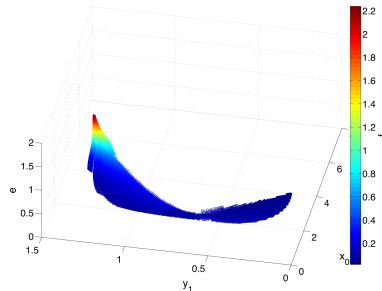
My first pass at sampling, using only 10^4 points, produced *two* parameter points with non-unique equilibria. See the red points?

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.



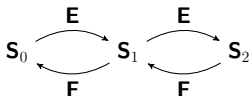
Solution set structure.

Summary

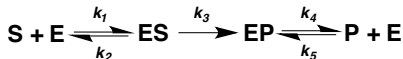
How can numerical methods help investigate a dynamical system?

- ▶ 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

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

VARIABLES

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

CONSERVED QUANTITIES

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

PARAMETERS

$$\alpha = \frac{c_{1,0}^E - c_0^E}{c_0^F - c_{1,0}^F} \quad \beta = \frac{c_2^E - c_{1,2}^E}{c_{1,2}^F - c_2^F} \quad \epsilon_0 = \frac{E_{\text{tot}}}{K_0^E} \quad \epsilon_1 = \frac{E_{\text{tot}}}{K_1^E} \quad \epsilon_2 = \frac{E_{\text{tot}}}{K_2^E} \quad \phi_0 = \frac{F_{\text{tot}}}{K_0^F} \quad \phi_1 = \frac{F_{\text{tot}}}{K_1^F} \quad \phi_2 = \frac{F_{\text{tot}}}{K_2^F}$$

diagram – Chris Nam, Swarthmore

Phosphorylation ctd.

System f ,

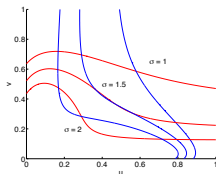
- ▶ 2 equations, of degree 4.
- ▶ 2 variables,
- ▶ 8 parameters.
- ▶ Care only about $\mathbb{R}^+$ solutions.

Questions about the parameter space:

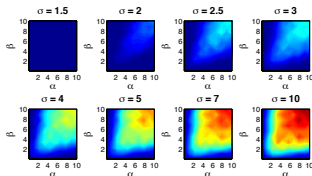
- ▶ Is there a region of *multistability*?
- ▶ What is its measure? Does it saturate?
- ▶ Is it connected?
- ▶ Is it convex?

VEGAS Sampling

A: POLYNOMIALS



B: DENSITY PLOTS



C: VEGAS SAMPLING

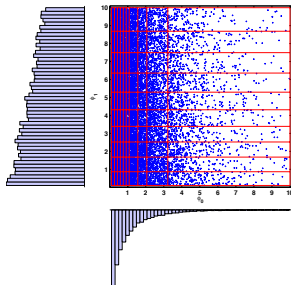


diagram – Chris Nam

Phosphorylation ctd.

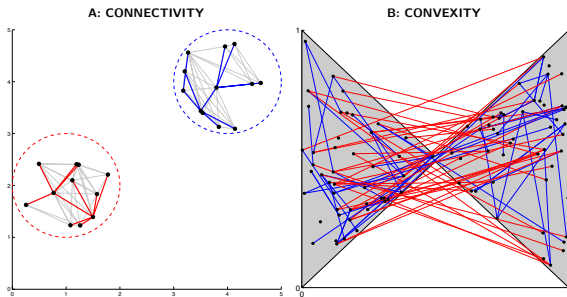
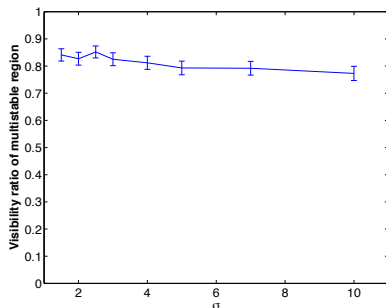
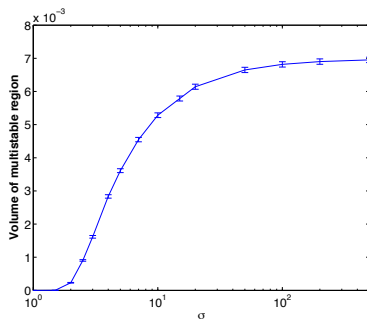


diagram – Chris Nam

Phosphorylation ctd.

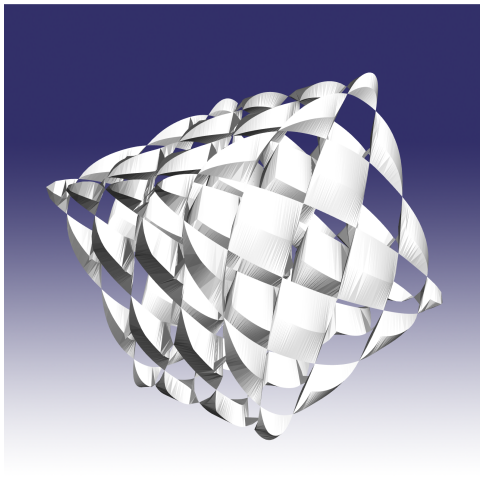


convexity as function of σ .



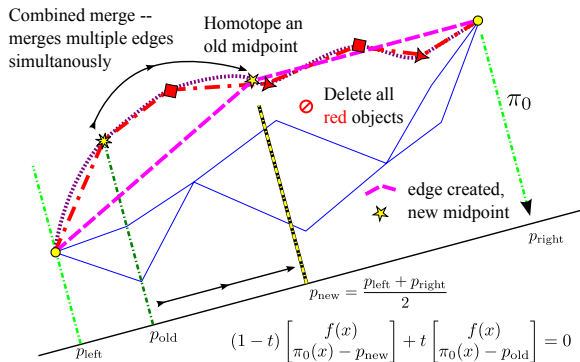
volume as function of σ .

diagrams – Chris Nam

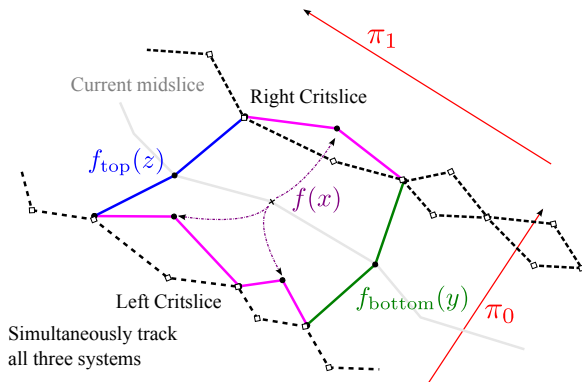


Thank you for your kind attention.

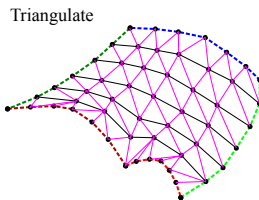
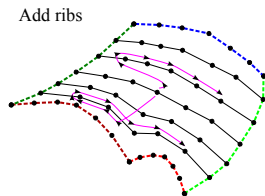
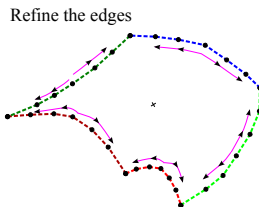
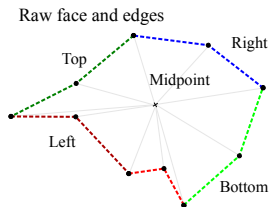
Curve Merge



Connect the dots



Surface Refinement



Generic Decomposition

Input: polynomial system f , witness set W , projections $\pi_1, \dots, \pi_d$.

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.