Numerical methods for investigating parameter spaces for parameterized polynomial systems

Dan Bates
Colorado State University

CUNY Symbolic-Numeric Computing Seminar
17 March 2016
Goal: Satisfy the computational needs of the user.

Researcher in some discipline with a parameterized polynomial system, asking questions about the parameter space.
Goal: Satisfy the computational needs of the user.

Researcher in some discipline with a parameterized polynomial system, asking questions about the parameter space.

Ultimate desired outcome: Software that one can easily query to answer (partially or completely) basic questions about parameter spaces.
Goal: Satisfy the computational needs of the user.

Researcher in some discipline with a parameterized polynomial system, asking questions about the parameter space.

Ultimate desired outcome: Software that one can easily query to answer (partially or completely) basic questions about parameter spaces.

Which methods? Whatever works. The consumers want reliable answers as rapidly as possible. (See Agnes Szanto’s 9/24/15 talk about certification!)
Goal: Satisfy the computational needs of the user.

Researcher in some discipline with a parameterized polynomial system, asking questions about the parameter space.

Ultimate desired outcome: Software that one can easily query to answer (partially or completely) basic questions about parameter spaces.

Which methods? Whatever works. The consumers want reliable answers as rapidly as possible. (See Agnes Szanto’s 9/24/15 talk about certification!)

Which software? Whatever works. I like Bertini-related products, but this could eventually be connected to any core continuator.
Today’s goal: Describe a few problems with parameterized polynomial systems and a few recent (numerical) techniques that might (not) eventually be part of the solution.
Today’s goal: Describe a few problems with parameterized polynomial systems and a few recent (numerical) techniques that might (not) eventually be part of the solution.

Caveats:

1. These lists are not comprehensive. There are no firm conclusions — just some thoughts to get the ball rolling.

2. There are different meanings of “parameter space” — real vs. complex, Euclidean space, an algebraic subset, etc.

3. Focus today is on numerical methods, but there are surely useful symbolic and hybrid methods for this, too. (See Robert Lewis’ talk in February 2015!)
GAME PLAN

1. **Motivation**: Several applications

2. Homotopy continuation, very briefly

3. Survey of recent developments
   a. **Heavy machinery**: Fiber products
   b. **Moving intelligently**: Gradient descent homotopies
   c. **Searching broadly**: Paramotopy

4. Bertini 2.x update

5. Other announcements
Steiner-Fulton real conic problem: Find 5 real plane conics such that there are 3264 real plane conics tangent to all 5. (Fulton’s variant on a counting problem by Steiner, asking for the number of plane conics tangent to 5 general plane conics.)

Goal: Find a real point in some parameter space at which all solutions of some polynomial system are real.

Progress: Solved by Fulton (unpublished), later in [RTV97], and confirmed numerically in [GH15].
**Steiner-Fulton real conic problem:** Find 5 real plane conics such that there are 3264 real plane conics tangent to all 5. (Fulton’s variant on a counting problem by Steiner, asking for the number of plane conics tangent to 5 general plane conics.)

**Goal:** Find a real point in some parameter space at which all solutions of some polynomial system are real.

**Progress:** Solved by Fulton (unpublished), later in [RTV97], and confirmed numerically in [GH15].

**Stewart-Gough mechanisms:** A general Stewart-Gough platform (with given leg lengths, etc.) has 40 (complex) assembly modes. Is there a choice of parameters for which there are 40 real assembly modes?

**Goal:** Same as previous.

**Progress:** Solved in [D98] and confirmed numerically in [GH15].
Motivation: Applications

**Exceptional mechanisms:** Given a mechanism type, find all examples of that type having “exceptional” mobility, i.e., more mobility than a general example of the mechanism type.

**Goal:** Find all points in some parameter space at which the dimension of the solution set of the polynomial system is greater than expected.

**Progress:** Simple cases solved, but still generally open.
Exceptional mechanisms: Given a mechanism type, find all examples of that type having “exceptional” mobility, i.e., more mobility than a general example of the mechanism type.
Goal: Find all points in some parameter space at which the dimension of the solution set of the polynomial system is greater than expected.
Progress: Simple cases solved, but still generally open.

Chemical reaction networks: Given a polynomial system derived from a chemical reaction network, find positive real parameter values (typically reaction rates) at which the solution set has an unexpectedly high number of positive, real solutions.
Goal: Find positive real parameter values at which the solutions of the polynomial system include more real solutions than “usual.”
Progress: Some work, but still very much open.
GAME PLAN

1. **Motivation**: Several applications

2. **Homotopy continuation, very briefly**

3. **Survey of recent developments**
   a. **Heavy machinery**: Fiber products
   b. **Moving intelligently**: Gradient descent homotopies
   c. **Searching broadly**: Paramotopy

4. **Bertini 2.x update**

5. **Other announcements**
Homotopy continuation

\[ H(z, t) = f(z) \cdot (1 - t) + \gamma g(z) \cdot t \]
Homotopy continuation

\[ H(z, t) = f(z) \cdot (1 - t) + \gamma g(z) \cdot t \]
Homotopy continuation

\[ H(z, t) = f(z) \cdot (1 - t) + \gamma g(z) \cdot t \]
Homotopy continuation

\[ H(z, t) = f(z) \cdot (1 - t) + \gamma g(z) \cdot t \]
Homotopy continuation

\[ H(z, t) = f(z) \cdot (1 - t) + \gamma g(z) \cdot t \]
Homotopy continuation

\[ H(z, t) = f(z) \cdot (1 - t) + \gamma g(z) \cdot t \]
Homotopy continuation

\[ H(z, t) = f(z) \cdot (1 - t) + \gamma g(z) \cdot t \]
Homotopy continuation

\[ H(z, t) = f(z) \cdot (1 - t) + \gamma g(z) \cdot t \]
Homotopy continuation

\[ H(z, t) = f(z) \cdot (1 - t) + \gamma g(z) \cdot t \]

This yields numerical approximations to complex isolated solutions (points).
Homotopy continuation

\[ H(z, t) = f(z) \cdot (1 - t) + \gamma g(z) \cdot t \]

Missing many details: Adaptive steplength, adaptive precision, endgames, etc.
GAME PLAN

1. **Motivation**: Several applications

2. Homotopy continuation, very briefly

3. **Survey of recent developments**
   a. **Heavy machinery**: Fiber products
   b. **Moving intelligently**: Gradient descent homotopies
   c. **Searching broadly**: Paramotopy

4. Bertini 2.x update

5. Other announcements
Goal: Identify points in the (complex) parameter space where the dimension of the (complex) solution set is higher than expected.

Heavy machinery: Fiber products
Goal: Identify points in the (complex) parameter space where the dimension of the (complex) solution set is higher than expected.

Example:

\[ ax^2 + bx + c = 0 \]

Parameter space \( P \) (a,b,c)
Variable space \( X \) (x)

Heavy machinery: Fiber products
Goal: Identify points in the (complex) parameter space where the dimension of the (complex) solution set is higher than expected.

Example:

\[ ax^2 + bx + c = 0 \]

Parameter space \( P \) \((a,b,c)\)
Variable space \( X \) \((x)\)

Two cases:
* 2 solutions “almost everywhere,” counting multiplicity.
  (Sometimes real, sometimes not.)
* **At \((0,0,0)\):** line of solutions.
Goal: Identify points in the (complex) parameter space where the dimension of the (complex) solution set is higher than expected.

Example:

$ax^2 + bx + c = 0$

Parameter space $P$ (a,b,c)
Variable space $X$ (x)

Two cases:
* 2 solutions “almost everywhere,” counting multiplicity. (Sometimes real, sometimes not.)
* At (0,0,0): line of solutions.
Heavy machinery: Fiber products

**PROBLEM:** \(\dim(P) = 3\), so the interesting fiber is buried in a 3-dimensional set.
**Heavy machinery:** Fiber products

**PROBLEM:** \( \dim(P) = 3 \), so the interesting fiber is buried in a 3-dimensional set.

Idea (from [SW08]): Take fiber products (just products in the fiber!).
**Problem:** \( \dim(P) = 3 \), so the interesting fiber is buried in a 3-dimensional set.

Idea (from [SW08]): Take fiber products (just products in the fiber!).

**Heavy machinery:** Fiber products
**Heavy machinery:** Fiber products

**PROBLEM:** $\dim(P) = 3$, so the interesting fiber is buried in a 3-dimensional set.

Idea (from [SW08]): Take fiber products (just products in the fiber!).
**PROBLEM:** \( \dim(P) = 3 \), so the interesting fiber is buried in a 3-dimensional set.

**Idea (from [SW08]):** Take fiber products (just products in the fiber!).

**Key points:**
1. Fiber dimension increases in a predictable way.
2. Higher fiber dimensions increase more rapidly than lower fiber dimensions, so interesting fibers eventually “pop out” of the non-interesting mess.
Current status: Original [SW08] formulation requires solving massive polynomial systems from scratch. OK for small examples (2-link mechanisms).

Coming up: Small team (B-Hanson-Hauenstein-Liddell-Wampler) making algorithmic improvements to [SW08] technique. For example, we use regeneration to move from the system with \( k \) fiber products to that with \( k+1 \). Soon: Tackle classes of mechanisms that were not previously considered methodically.
Goal: Move around parameter space intelligently to find points with more (or fewer) real solutions. More specifically, find nearest point on discriminant locus and move through it.
**Moving intelligently: Gradient descent homotopies**

**Goal:** Move around parameter space intelligently to find points with more (or fewer) real solutions. More specifically, find nearest point on discriminant locus and move through it.

Core of idea originated with Dietmaier [D98], but numerical difficulties (caused by linearizing near a singularity) were mitigated by Griffin-Hauenstein [GH15] by building a homotopy that moves *both* the point in parameter space and one point in the fiber. [GH15] also allows for algebraic parameter spaces.
**Moving intelligently:** Gradient descent homotopies

**Goal:** Move around parameter space intelligently to find points with more (or fewer) real solutions. More specifically, find nearest point on discriminant locus and move through it.

Core of idea originated with Dietmaier [D98], but numerical difficulties (caused by linearizing near a singularity) were mitigated by Griffin-Hauenstein [GH15] by building a homotopy that moves *both* the point in parameter space and one point in the fiber. [GH15] also allows for algebraic parameter spaces.

Paths followed (just in parameter space):

![Figure 3: Plot for (a) $f(x) = x_2^2 + x_1^2(x_1 - 1)(x_1 - 2)$ and (b) $g(x) = x_2^2 - x_1^2(x_1 + 1)$.](from [GH15])
Moving intelligently: Gradient descent homotopies

Notice that moving to (and through) the discriminant locus with just one point in the fiber need not change the number of real solutions:
**Moving intelligently:** Gradient descent homotopies

**Current status:** Exists, implemented. [GH15] applied this to solve the Steiner and Stewart-Gough problems.

**Coming up:** ??

Having many points in the fiber means both a complicated discriminant locus AND many wasted trips to the discriminant locus. The use of a branch-specific discriminant locus would speed things up.
Goal: Allow the user to efficiently collect data (including summary statistics) at many points in (Euclidean) parameter space.
**Goal:** Allow the user to efficiently collect data (including summary statistics) at many points in (Euclidean) parameter space.

* Software by B-Brake-Niemerg [BBN].
* Uses parameter homotopies [MS] (much like the cheater’s homotopy [LSY]).
* Relies heavily on Bertini [BHSW] as a computational engine.
**Goal:** Allow the user to efficiently collect data (including summary statistics) at many points in (Euclidean) parameter space.

- Software by B-Brake-Niemerg [BBN].
- Uses parameter homotopies [MS] (much like the cheater’s homotopy [LSY]).
- Relies heavily on Bertini [BHSW] as a computational engine.

**Idea:**

- Plotting points in parameter space.

```
  . q'
```

```
  . q_i
```
**Searching broadly: Paramotopy**

**Goal:** Allow the user to efficiently collect data (including summary statistics) at many points in (Euclidean) parameter space.

* Software by B-Brake-Niemerg [BBN].
* Uses parameter homotopies [MS] (much like the cheater’s homotopy [LSY]).
* Relies heavily on Bertini [BHSW] as a computational engine.

**Idea:**

\[ q_i \rightarrow q' \]
Goal: Allow the user to efficiently collect data (including summary statistics) at many points in (Euclidean) parameter space.

* Software by B-Brake-Niemerg [BBN].
* Uses parameter homotopies [MS] (much like the cheater’s homotopy [LSY]).
* Relies heavily on Bertini [BHSW] as a computational engine.

Idea:
**Searching broadly: Paramotopy**

**Goal:** Allow the user to efficiently collect data (including summary statistics) at many points in (Euclidean) parameter space.

* Software by B-Brake-Niemerg [BBN].
* Uses parameter homotopies [MS] (much like the cheater’s homotopy [LSY]).
* Relies heavily on Bertini [BHSW] as a computational engine.

**Idea:**
Goal: Allow the user to efficiently collect data (including summary statistics) at many points in (Euclidean) parameter space.

* Software by B-Brake-Niemerg [BBN].
* Uses parameter homotopies [MS] (much like the cheater’s homotopy [LSY]).
* Relies heavily on Bertini [BHSW] as a computational engine.

Idea:
Searching broadly: Paramotopy

Paramotopy input file

# vars, etc.

polynomials

fixed parameters

meshes for 2 parameters (could supply values)
Searching broadly: Paramotopy

Paramotopy is interactive, or you can run it as a black box.

There are Matlab files to collect and massage the output.
Searching broadly: Paramotopy

Path failure information is collected and brought to the attention of the user.

There were 84 points with path failures, out of 2000 total points.
current iteration 0

PathFailure current settings:
-name- -value-
maxautoiterations 3
newrandommethod 1
tightentolerances 1
turnon_securitylevel1 0

Path Failure Menu:
1) ReRun Failed Paths
2) Clear Failed Path Data (start over)
3) Change path failure settings (resets tolerance tightening)
* 
0) Go Back
Searching broadly: Paramotopy

**Current status:** Exists. Used in solving a chemical reaction network problem [BBGNN15] and elsewhere.

**Coming up:** Moving into b2 (in some form), perhaps including other parameter space algorithms (gradient descent homotopies?).
GAME PLAN

1. **Motivation**: Several applications

2. Homotopy continuation, very briefly

3. Survey of recent developments
   a. **Heavy machinery**: Fiber products
   b. **Moving intelligently**: Gradient descent homotopies
   c. **Searching broadly**: Paramotopy

4. **Bertini 2.x update**

5. Other announcements
Bertini 1.x: Pros & cons

First released in 2006, now on version 1.5.

**Pros**
Various techniques available (AMP, regeneration, NID)
Decent artificial intelligence for some tolerances
OK as a blackbox
Used in few hundred papers

**Cons**
Small dev team (BBHSW - only 2 active)
Onion has many layers - maintenance and extension are very hard & restricted
Many “special” versions & Easter eggs
Restrictive license (no redistribution)
No scripting, no access to subalgorithms

Time to reinvent the wheel!
Connections to other software

Bertini

Matlab
- BertinILab: B, Newell, Niemerg

R
- LocalDimFinder
- Bertini\_real
- Multiplicity
- Paramotopy
- other bits of code (recovery of exactness, Chern class computations)

Macaulay2
- Bertini.m2, in NAG4M2: B, Gross, Leykin, Rodriguez
- ApCoCoA dev team

CoCoA
- Hauenstein, Sottile
- Hauenstein, Liddell
- cadenza

Hauenstein, Sommese, Wampler
- Bertini\_real
- Bertini
- LocalDimFinder
- Hao, Sommese, Zeng
- Brake, Niemerg
- Brake, Hao, Hauenstein, Sommese, Wampler

Other software
- KhRo
- Bertini
- CoCoA
- Matlab
- R
- Macaulay2
Connections to other software

Matlab

BertiniLab: B, Newell, Niemerg

R

B, Brake, Hao, Hauenstein, Sommese, Wampler

Hauenstein, Sommese, Wampler

Bertini

Bertini.m2, in NAG4M2: B, Gross, Leykin, Rodriguez

ApCoCoA dev team

Macaulay2

CoCoA

B, Brake, Niemerg

B, Eklund, Hauenstein, McCoy, Peterson, Sommese, etc.

B, Sottile

Hauenstein, Sottile

Hauenstein, Liddell

alphaCertified

cadenza

B, Brake, Hao, Hauenstein, Sommese, Wampler

Hao, Sommese, Zeng

Bertini_real

Paramotopy

B, Brake, Niemerg

Hauenstein, Sommese

KhRo

other bits of code (recovery of exactness, Chern class computations)

LocalDimFinder

Multiplicity

See Jon Hauenstein’s 2/4/16 talk!
Bertini 2.0

Bertini Classic is now difficult to maintain and extend.

**New(ish) NSF grant (through ACI):** Development of Bertini 2.0

**Major points:**
- Highly modular in object-oriented language (C++)
- Different dev philosophy: Open-source (GPL v3, GitHub), large dev team
- Automatic frequent testing (jenkins), review process for core & modules
- Available as a library, incorporating other packages, including certification/verification, polyhedral methods, etc.
- Python interface (among others)
- Credit generator
- Workshops for novices and for module development
Bertini 2.0

Matlab

BertiniLab: B, Newell, Niemerg


R

B, Brake, Hao, Hauenstein, Sommese, Wampler

B, Brake, Niemerg

Hauenstein, Sommese, Zeng

Bertini

Bertini_real

LocalDimFinder

Matlab

Hauenstein, Sommese, Wampler

B, Brake, Hao, Hauenstein, Sommese, Wampler

B, Brake, Niemerg

Bertini.m2, in NAG4M2:
B, Gross, Leykin, Rodriguez

Macaulay2

ApCoCoA dev team

CoCoA

B, Brake, Niemerg

B, Eklund, Hauenstein, McCoy, Peterson, Sommese, etc.

Hauenstein, Sottile

Hauenstein, Sommese, Zeng

Paramotopy

Multiplicity

B, Brake, Niemerg

Bertini

Bertini.m2, in NAG4M2:
B, Gross, Leykin, Rodriguez

Hauenstein, Sottile

Hauenstein, Liddell

cadenza

BertiniLab: B, Newell, Niemerg

ApCoCoA dev team

B, Brake, Hao, Hauenstein, Sommese, Wampler

B, Brake, Niemerg

Bertini.m2, in NAG4M2:
B, Gross, Leykin, Rodriguez

Bertini 2.0

56

Bertini

Bertini_real

LocalDimFinder

other bits of code (recovery of exactness, Chern class computations)
Core Methods

Bertini 2.0

Unsupported modules

Supported Modules

Core Methods

Matlab

R

Macaulay2

CoCoA

Bertini_real
LocalDimFinder
KhRo
Paramotopy
alphaCertified

Other bits of code
(recovery of exactness, Chern class computations)

Multiplicity

Anything you want to write!
GAME PLAN

1. **Motivation**: Several applications

2. Homotopy continuation, very briefly

3. Survey of recent developments
   a. **Heavy machinery**: Fiber products
   b. **Moving intelligently**: Gradient descent homotopies
   c. **Searching broadly**: Paramotopy

4. Bertini 2.x update

5. **Other announcements**
Other Announcements

Workshop on Software & Applications of Numerical Algebraic Geometry
University of Notre Dame (South Bend, IN, USA)
23-25 May 2016

Summer school on Applied Algebraic Geometry
Notre Dame campus in Rome
Summer 2018

ICERM Special Semester on Nonlinear Algebra
Providence, RI
Fall 2018

SIAM Journal on Applied Algebra and Geometry (SIAGA)
Accepting submissions starting next Wednesday!
Thanks!

References


[BHSW]: Bates, Hauenstein, Sommese, and Wampler. Bertini: Software for numerical algebraic geometry. Available at bertini.nd.edu. (b2 is on GitHub.)


This work was partially supported by NSF grants DMS-1115668 and ACI-1440467.