Numer. Alg. Geometry Collaborators

- Daniel Bates*  (CSU)
- Jonathan Hauenstein*  (Fields/Texas A&M)
- Chris Peterson  (CSU)
- Charles Wampler*  (GM R & D)

*Bertini Team
Biological Modeling Collaborators

- Wenrui Hao (Notre Dame)
- Jonathan Hauenstein (Fields/Texas A&M)
- Bei Hu (Notre Dame)
- Yuan Liu (Notre Dame)
- Timothy McCoy (Notre Dame)
- Yong-Tao Zhang (Notre Dame)
Overview

- Numerical Algebraic Geometry
  - Adaptive Multiprecision
  - Local Dimension Test
  - Regeneration
- Bertini
- Zebra Fish
- Tumor Growth
Numerical Algebraic Geometry

- **Goal:** To numerically manipulate algebraic sets
- **Technical Challenge:** To combine high performance numerics with algebraic geometry
- **Applications:**
  - Robotics and Mechanism Theory
  - Chemical Reactions including combustion
  - Computation of algebraic-geometric invariants
  - Solution of discretizations of nonlinear differential equations

Robotics/Mechanism Theory

Combustion

\[
\begin{align*}
O_2 & \Rightarrow 2O \\
H_2 & \Rightarrow 2H \\
N_2 & \Rightarrow 2N \\
CO_2 & \Rightarrow O + CO \\
OH & \Rightarrow O + H \\
H_2O & \Rightarrow O + 2H \\
NO & \Rightarrow O + N
\end{align*}
\]

There are four conservation equations:

\[
\begin{align*}
T_H &= X_H + 2X_{H_2} + X_{OH} + 2X_{H_2O} \\
T_C &= X_{CO} + X_{CO_2} \\
T_O &= X_O + X_{CO} + 2X_{O_2} + 2X_{CO_2} + X_{OH} + X_{H_2O} + X_{NO} \\
T_N &= X_N + 2X_{N_2} + X_{NO}
\end{align*}
\]

graphics on right from Sommese-Wampler Book
General References up to end of 2004


Three Recent Articles


Adaptive Multiprecision

Is Costly!

<table>
<thead>
<tr>
<th></th>
<th>double (52 bits)</th>
<th>64 bits</th>
<th>96 bits</th>
<th>128 bits</th>
<th>256 bits</th>
<th>512 bits</th>
<th>1024 bits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2.447</td>
<td>32.616</td>
<td>35.456</td>
<td>35.829</td>
<td>50.330</td>
<td>73.009</td>
<td>124.401</td>
</tr>
</tbody>
</table>

GMP is convenient, but because of its relative machine independence it takes almost no advantage of the built-in hardware floating point.
Adaptive multiprecision

<table>
<thead>
<tr>
<th></th>
<th>96-bit</th>
<th>AMP</th>
<th>AMP2</th>
</tr>
</thead>
<tbody>
<tr>
<td>time (sec)</td>
<td>196.143</td>
<td>47.393</td>
<td>39.854</td>
</tr>
<tr>
<td>paths/sec</td>
<td>1.305</td>
<td>5.402</td>
<td>6.423</td>
</tr>
</tbody>
</table>

Average time, in seconds, needed to solve the *Inverse Kinematic Problem of General 6R Serial Robot* using Bertini with tracking tolerance of $10^{-6}$ and final tolerance of $10^{-12}$

2.4 GHz, Opteron 250 processor
Adaptive multiprecision

Near-singular conditions actually arise.

Out of 143,360 paths (for the 9-point problem):

1184 paths (0.826%) used higher precision and then dropped back to double precision before starting the endgame

680 paths (0.474%) used at least 96-bit precision and then dropped back to double precision before starting the endgame
Continuation’s Core Computation

- Given a system $f(x) = 0$ of $N$ polynomials in $N$ unknowns, continuation computes a finite set $S$ of solutions such that:
  - any isolated root of $f(x) = 0$ is contained in $S$;
  - any isolated root “occurs” a number of times equal to its multiplicity as a solution of $f(x) = 0$;
  - $S$ is often larger than the set of isolated solutions.
Local Dimension Test

- The essential case: check if $p$ is isolated
- Homotopy continuation yields a number which bounds the multiplicity if the point was isolated.
- If not isolated, the space of truncated Taylor series around $p$ of functions on the solution space is strictly increasing in dimension
- The Macaulay matrix (as presented by Dayton-Zeng) computes this dimension
Implementation Considerations

- Computation of the rank of the Macaulay matrix requires
  - Different levels of precision
  - Reliable multiple precision endgame to compute point p to needed accuracy
### Some Comparisons Using Bertini

<table>
<thead>
<tr>
<th>$m$</th>
<th>$is_isolated$ slicing</th>
<th>$is_isolated$ cascade</th>
<th>membership test slicing</th>
<th>membership test cascade</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.12</td>
<td>0.15</td>
<td>0.12</td>
<td>0.17</td>
</tr>
<tr>
<td>4</td>
<td>0.71</td>
<td>1.12</td>
<td>1.15</td>
<td>1.32</td>
</tr>
<tr>
<td>5</td>
<td>4.96</td>
<td>7.30</td>
<td>11.86</td>
<td>10.68</td>
</tr>
<tr>
<td>6</td>
<td>29.26</td>
<td>71.51</td>
<td>149.59</td>
<td>92.28</td>
</tr>
<tr>
<td>7</td>
<td>183.14</td>
<td>288.70</td>
<td>2036.73</td>
<td>854.33</td>
</tr>
<tr>
<td>8</td>
<td>1157.74</td>
<td>1714.35</td>
<td>17362.71</td>
<td>8720.14</td>
</tr>
<tr>
<td>9</td>
<td>7296.78</td>
<td>9533.50</td>
<td>219509.84</td>
<td>83060.43</td>
</tr>
</tbody>
</table>

*Comparison for computing a numerical irreducible decomposition for $G_m$, in seconds*

**2xm adjacent minors of 3xm matrix**
Parallel Version (64 cores)

- 8 dual quad-core Xeon 5410s (2.33 GH)

<table>
<thead>
<tr>
<th>$m$</th>
<th>$is_isolated$ slicing</th>
<th>$is_isolated$ cascade</th>
<th>membership test slicing</th>
<th>membership test cascade</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>15.83</td>
<td>16.36</td>
<td>82.59</td>
<td>30.03</td>
</tr>
<tr>
<td>8</td>
<td>35.87</td>
<td>49.88</td>
<td>350.96</td>
<td>168.46</td>
</tr>
<tr>
<td>9</td>
<td>138.91</td>
<td>213.23</td>
<td>3320.04</td>
<td>1399.43</td>
</tr>
</tbody>
</table>
Equation-by-Equation Methods

- Potential to solve systems with relatively few solutions that are completely outside of the beyond the pale of standard continuation methods
  - **Intersection Method** by Sommese, Verschelde, and Wampler
  - **Regeneration Method** by Hauenstein, Sommese, and Wampler
Basic Idea

It is a natural and ancient approach to work equation by equation.

First Solve $f_1(x) = 0$;

then solve

$$\begin{bmatrix} f_1(z) \\ f_2(z) \end{bmatrix} = 0;$$

then solve

$$\begin{bmatrix} f_1(z) \\ f_2(z) \\ f_3(z) \end{bmatrix} = 0; \ldots \text{and so on.}$$
Both methods generate a witness set for $f_1, \ldots, f_{k+1}$.

\[ H(z, t) = \begin{bmatrix}
  f_1(z) \\
  \vdots \\
  f_k(z) \\
  (1 - t) f_{k+1}(z) + \gamma t \prod_{i=1}^{\deg f_{k+1}} L_{k+1,i}(z)
\end{bmatrix} \]

\[ H(w, t) = \begin{bmatrix}
  f_1 \circ \pi_1 \\
  \vdots \\
  f_k \circ \pi_1 \\
  f_{k+1} \circ \pi_2 \\
\end{bmatrix} \left( (1 - t) \Psi(w) + \gamma t \Phi(w) \right) \]

Intersection
\[ H(z, t) = \begin{bmatrix} f_1(z) \\ \vdots \\ f_k(z) \\ (1 - t) f_{k+1}(z) + \gamma t \prod_{i=1}^{\deg f_{k+1}} L_{k+1,i}(z) \end{bmatrix} \]
Hardware

- Continuation is computationally intensive. On average:
  - in 1985: 3 minutes/path on largest mainframes.
Hardware

- Continuation is computationally intensive. On average:
  - in 1985: 3 minutes/path on largest mainframes.
  - in 1991: over 8 seconds/path on an IBM 3081; 2.5 seconds/path on a top-of-the-line IBM 3090.
Hardware

- Continuation is computationally intensive. On average:
  - in 1985: 3 minutes/path on largest mainframes.
  - in 1991: over 8 seconds/path on an IBM 3081; 2.5 seconds/path on a top-of-the-line IBM 3090.
  - 2008: 10+ paths a second on an single processor desktop CPU; 1000’s of paths/second on moderately sized clusters.
Bertini

- Developed by Daniel Bates, Jonathan Hauenstein, Charles Wampler, and myself

- Binaries for Linux (including clusters and multiple core workstations), Macs, Windows are freely available at

```
www.nd.edu/~sommese/bertini
```
Bertini

- Bertini is designed to
  - Be efficient and robust, e.g., straightline evaluation, numerics with careful error control
  - With data structures reflecting the underlying geometry
  - Take advantage of parallel hardware
  - To dynamically adjust the precision to achieve a solution with a prespecified error.
Major Ingredients in Bertini

- Adaptive Multiprecision
- Straightline evaluation
- Special Homotopies
- Genericity
- Endgames & ODE Methods
- Intersections
- Deflation
- Multiplicity & Local Dimension Testing
- Regeneration
Solving Differential Equations

- Direct solution and refinement.
Let $n \in \mathbb{N}$. For $1 \leq i \leq n$ and $1 \leq j \leq 4$, define

$$f_{ij} = \frac{1}{25} (u_{i+1,j} - 2u_{i,j} + u_{i-1,j})$$

$$+ \frac{1}{(n+1)^2} (u_{i,j+1} - 2u_{i,j} + u_{i,j-1}) + \frac{1}{25(n+1)^2} u_{i,j} (1 - v_{i,j})$$

$$g_{ij} = \frac{1}{25} (v_{i+1,j} - 2v_{i,j} + v_{i-1,j})$$

$$+ \frac{1}{(n+1)^2} (v_{i,j+1} - 2v_{i,j} + v_{i,j-1}) + \frac{1}{25(n+1)^2} v_{i,j} (u_{i,j} - 1)$$

with $u_{0,j} = v_{0,j} = u_{n+1,j} = v_{n+1,j} = u_{i,0} = v_{i,0} = u_{i,5} = v_{i,5} = 0$. 
- 8n quadratics with 8n variables
- Total degree $2^{8n}$
- Actually has $2^{4n}$ nonsingular isolated solutions

<table>
<thead>
<tr>
<th>n</th>
<th>total degree</th>
<th>2-homogeneous</th>
<th>polyhedral</th>
<th>regeneration</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>paths</td>
<td>paths</td>
<td>paths</td>
<td>paths</td>
</tr>
<tr>
<td>1</td>
<td>256</td>
<td>70</td>
<td>16</td>
<td>60</td>
</tr>
<tr>
<td>2</td>
<td>65,536</td>
<td>12,870</td>
<td>256</td>
<td>1020</td>
</tr>
<tr>
<td>3</td>
<td>16,777,216</td>
<td>2,704,156</td>
<td>4096</td>
<td>16,380</td>
</tr>
<tr>
<td>4</td>
<td>4,294,967,296</td>
<td>601,080,390</td>
<td>65,536</td>
<td>262,140</td>
</tr>
<tr>
<td>5</td>
<td>1,099,511,627,776</td>
<td>137,846,528,820</td>
<td>1,048,576</td>
<td>4,194,300</td>
</tr>
</tbody>
</table>

Randomization, Relaxation, & Complexity
BIRS, March 4, 2010
<table>
<thead>
<tr>
<th>n</th>
<th>PHC</th>
<th>HOM4PS-2.0</th>
<th>Bertini</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>polyhedral</td>
<td>polyhedral</td>
<td>regeneration</td>
</tr>
<tr>
<td></td>
<td>0.6s</td>
<td>0.1s</td>
<td>0.3s</td>
</tr>
<tr>
<td>2</td>
<td>4m57s</td>
<td>7.3s</td>
<td>15.6s</td>
</tr>
<tr>
<td>3</td>
<td>18d10h18m56s</td>
<td>9m32s</td>
<td>9m43s</td>
</tr>
<tr>
<td>4</td>
<td>-</td>
<td>3d8h28m30s</td>
<td>5h22m15s</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>7m32s</td>
</tr>
<tr>
<td>5</td>
<td>-</td>
<td>-</td>
<td>6d16h27m3s</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3h41m24s</td>
</tr>
</tbody>
</table>

\(n = 5\) (40 equations & 40 variables): < 80 min. with 200 cores (25 dual Xeon 5410 nodes)
Zebra Fish

- Why do the *stripes* on a zebra fish or the *spots* on a tiger form the patterns they do?
  - Alan Turing (1952), *The chemical basis of morphogenesis: nonlinear diffusion equations*.

- A good reference for this story is *Mathematical Biology* by J.D. Murray
Based on the model developed in


Our work

- W. Hao, Y. Liu, J. Hauenstein, B. Hu, A. Sommese, and Y.-T. Zhang, Multiple stable steady states of a reaction-diffusion model on zebrafish dorsal-ventral patterning, *to appear* *Discrete and Continuous Dynamical Systems - Series S*.
The differential equation system

\[
\frac{\partial[L]}{\partial t} = D_L \frac{\partial^2[L]}{\partial x^2} - k_{on}[L](R_0 - [LR]) + k_{off}[LR] - j_{on}[L][C] + (j_{off} + \tau)[LC] + V_L;
\]

\[
\frac{\partial[LR]}{\partial t} = k_{on}[L](R_0 - [LR]) - (k_{off} + k_{deg})[LR];
\]

\[
\frac{\partial[LC]}{\partial t} = D_{LS} \frac{\partial^2[LC]}{\partial x^2} + j_{on}[L][C] - (j_{off} + \tau)[LC];
\]

\[
\frac{\partial[C]}{\partial t} = D_C \frac{\partial^2[C]}{\partial x^2} - j_{on}[L][C] + j_{off}[LC] + V_C,
\]

\[
V_C = V_{C_{\text{min}}} + \frac{V_{C_{\text{max}}} - V_{C_{\text{min}}}}{1 + \gamma_C[LR]} + \begin{cases} 
V_{C_{\text{org}}} e^{-at}, & \text{if } x \geq \frac{7}{8} x_{\text{max}}; \\
0, & \text{otherwise.}
\end{cases}
\]

\[
V_L = V_{L_{\text{min}}} + \frac{V_{L_{\text{max}}} - V_{L_{\text{min}}}}{1 + \gamma_L[LR]^{-1}} + V_{L_{\text{mat}}} e^{-bt}.
\]
Solutions
Some timings

- Total degree $16^{N-1}$ (which = 4,294,967,296 when $N = 9$).

<table>
<thead>
<tr>
<th>N</th>
<th>lin. prod. bound</th>
<th>solutions over $\mathbb{C}$</th>
<th>solutions over $\mathbb{R}$</th>
<th>computing nodes</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>25</td>
<td>16</td>
<td>6</td>
<td>serial</td>
<td>2.7s</td>
</tr>
<tr>
<td>4</td>
<td>125</td>
<td>98</td>
<td>16</td>
<td>serial</td>
<td>14.4s</td>
</tr>
<tr>
<td>5</td>
<td>625</td>
<td>544</td>
<td>28</td>
<td>1</td>
<td>21.1s</td>
</tr>
<tr>
<td>6</td>
<td>3,125</td>
<td>2,882</td>
<td>184</td>
<td>5</td>
<td>51.6s</td>
</tr>
<tr>
<td>7</td>
<td>15,625</td>
<td>14,896</td>
<td>930</td>
<td>25</td>
<td>2m43s</td>
</tr>
<tr>
<td>8</td>
<td>78,125</td>
<td>75,938</td>
<td>3,720</td>
<td>25</td>
<td>35m2s</td>
</tr>
<tr>
<td>9</td>
<td>390,625</td>
<td>384,064</td>
<td>17,974</td>
<td>25</td>
<td>11h3m</td>
</tr>
</tbody>
</table>

Table 2.1: Summary of solving the discretized system for $3 \leq N \leq 9$
Tumor growth

\[ \sigma_t - \Delta \sigma = -\sigma \quad \text{in } \Omega(t) \]
\[ -\Delta p = \mu (\sigma - \bar{\sigma}) \quad \text{in } \Omega(t) \]

\[ \sigma = 1 \quad \text{on } \partial \Omega(t) \]
\[ p = \kappa \quad \text{on } \partial \Omega(t) \]
\[ \frac{\partial p}{\partial n} = -V_n \quad \text{on } \partial \Omega(t). \]
Assumptions

- In vitro

$\Omega(t)$ denotes the tumor region, $\sigma$ denote the concentration of nutrients, $p$ denote the pressure, $\tilde{\sigma}$ denote the concentration of nutrients needed for sustainability, and $\mu$ denote the aggressiveness of the tumor. Let $\kappa$ denote the mean curvature, $n$ denote the outward normal direction, and $V_n$ denote the velocity of $\partial\Omega(t)$ in the outward normal direction $n$. 
Governing equations:

• Diffusion of the nutrients:

\[ \sigma_t - \Delta \sigma + \sigma = 0 \quad \text{in } \Omega(t). \]

• Conservation of mass: \( \text{div } \vec{V} = S \), \( S \) = proliferation rate. Assuming linear dependence on \( \sigma \): \( S = \mu(\sigma - \bar{\sigma}) \), (here \( \bar{\sigma} > 0 \) is the death rate)

• Porous medium in tumor region: Darcy’s law: \( \vec{V} = -\nabla p \). Thus

\[ \Delta p = -\mu(\sigma - \bar{\sigma}) \quad \text{in } \Omega(t). \]

• Continuity: \( V_n = -\frac{\partial p}{\partial n} \) on \( \partial \Omega(t) \)

where \( V_n \) = velocity in the normal \( n \) direction.
Radial solution is quite cheap: < 1 sec. (one core)
Moving Grid
3rd Order Stencil
Critical Points 3 minutes with 200 cores
Far Along the Branch
Further work

- Stability
- More realistic models
  - Three Dimensional Models
  - Necrotic Core Models (disconnected free boundaries)
Stationary Problem

\[
\sigma_t - \Delta \sigma + \sigma = 0, \quad x \in \Omega(t), \quad t > 0,
\]
\[
\sigma = 1, \quad x \in \Omega(t), \quad t > 0,
\]
\[
-\Delta \vec{v} + \nabla p = \frac{\mu}{3} \nabla (\sigma - \bar{\sigma}), \quad x \in \Omega(t), \quad t > 0,
\]
\[
\text{div} \vec{v} = \mu (\sigma - \bar{\sigma}), \quad x \in \Omega(t), \quad t > 0 \quad (\bar{\sigma} < 1),
\]
\[
T(\vec{v}, p) \vec{n} = \left( -\gamma \kappa + \frac{2}{3} \mu (1 - \bar{\sigma}) \right) \vec{n}, \quad x \in \Gamma(t), \quad t > 0,
\]
\[
T(\vec{v}, p) = (\nabla \vec{v})^T + \nabla \vec{v} - p \; I, \quad I = (\delta_{ij})_{i,j=1}^3,
\]
\[
V_n = \vec{v} \cdot \vec{n} \quad \text{on} \; \Gamma(t),
\]

subject to the constraints

\[
\int_{\Omega(t)} \vec{v} \; dx = 0, \quad \int_{\Omega(t)} \vec{v} \times \vec{x} \; dx = 0.
\]
Governing equations:

- Diffusion of the nutrients: \( \sigma_t - \Delta \sigma + \sigma = 0 \) in \( \Omega(t) \).

- Conservation of mass: \( \text{div} \vec{V} = S \), \( S \) = proliferation rate. Assume linear dependence on \( \sigma \): \( S = \mu (\sigma - \bar{\sigma}) \), (here \( \bar{\sigma} > 0 \) is the death rate)

- Instead of Darcy’s law, Stoke’s equation is used: \(-\nu \Delta \vec{V} + \nabla p - \frac{1}{3} \nu \nabla \text{div} \vec{V} = 0 \) in \( \Omega(t) \).

- Introducing the stress tensor \( Q = \nu (\nabla \vec{V} + (\nabla \vec{V})^T) - (p + \frac{2}{3} \nu \text{div} \vec{V}) I \) with components \( Q_{ij} = \nu \left( \frac{\partial v_j}{\partial x_i} + \frac{\partial v_i}{\partial x_j} \right) - \delta_{ij} \left( p + \frac{2}{3} \nu \text{div} \vec{V} \right) \), we then have

\[
Q \vec{n} = -\gamma \kappa \vec{n} \quad \text{on} \quad \Gamma(t), \quad t > 0,
\]

here the cell-to-cell adhesion equal to a constant \( \gamma \), \( \kappa \) is the mean curvature.

- Continuity: \( V_n = \vec{V} \cdot \vec{n} \) on \( \partial \Omega(t) \)
where \( V_n \) = velocity in the normal \( n \) direction.

Since \( \vec{V} \) is determined up to \( \vec{b} \times \vec{x} \), some additional constraints are needed.
Algebraic Geometry

- Infinite Dimensional Algebraic Sets = Solutions of Differential Equations?

- Coupled Towers of Finite Dimensional Algebraic Sets?
Summary

- Basic but difficult questions about Scientific Models lead to algebraic sets defined by highly structured, sparse systems of polynomials that are extremely large by classical standards.

- Numerical Algebraic Geometry can make contributions when coupled with moderate amounts of computer power and appropriate numerical software.