

# Numerical Algebraic Geometry: Theory and Practice

Andrew Sommese

University of Notre Dame

[www.nd.edu/~sommese](http://www.nd.edu/~sommese)

In collaboration with

Daniel Bates (Colorado State University)

Jonathan Hauenstein (University of Notre Dame)

Charles Wampler (General Motors R & D)



# Many have contributed to the Development of Numerical Algebraic Geometry

- Daniel Bates, Daniel Brake, Tianran Chen, Brent Davis, Wenrui Hao, Bei Hu, Jonathan Hauenstein, Tsung-Lin Lee, Anton Leykin, Tien-Yien Li, Timothy McCoy, Mathew Niemerg, Christopher Peterson, Jose Rodriguez, Jan Verschelde, Charles Wampler, Zhonggang Zeng, Ailing Zhao, . . . . .

# Surveys of Numerical Algebraic Geometry

## ■ Reference on the area up to 2005:

- A.J. Sommese and C.W. Wampler, *Numerical solution of systems of polynomials arising in engineering and science*, (2005), World Scientific Press.
- Survey up to 2010 oriented towards Kinematics
  - C.W. Wampler and A.J. Sommese, *Numerical Algebraic Geometry and Algebraic Kinematics*, Acta Numerica 20 (2011), 469-567.
- Up to 2013 oriented towards Bertini
  - D.J. Bates, J.D. Hauenstein, A.J. Sommese, and C.W. Wampler, *Numerically solving polynomial systems with Bertini*, (2013), SIAM.
- Developments related to systems of PDEs upto 2013
  - W. Hao, B. Hu, and A.J. Sommese, *Numerical algebraic geometry and differential equations*, in Future Vision and Trends on Shapes, Geometry and Algebra, ed. by R. De Amicis and G. Conti, Springer Proc. in Mathematics & Statistics, Vol. 84 (2014), 39-54.

# Overview

- Numerical Algebraic Geometry
  - Homotopy Continuation and Computation of Isolated Solutions
    - Numerical Issues
    - Genericity and Algebraic Geometry
  - Positive Dimensional Solution Sets
    - How to represent them
    - The Core Computation
  - Singular Points, Endgames, and Adaptive Precision
- Bertini: open-source software for Num. Alg. Geom.
  - Brief History
  - Future

# Computing Isolated Solutions of Polynomial Systems

- Find all isolated solutions of a polynomial system

$$f(x) = \begin{bmatrix} f_1(x) \\ \vdots \\ f_n(x) \end{bmatrix} = 0$$

on  $C^N$

# Why?

---

- To solve problems from engineering and science.

# Characteristics of Engineering Systems

- systems are sparse: they often have symmetries and have much smaller solution sets than would be expected.
- systems depend on parameters: typically they need to be solved many times for different values of the parameters.
- usually only real solutions are interesting.
- usually only finite solutions are interesting.
- nonsingular isolated solutions *were* the center of attention.



# Solving a system

---

- Homotopy continuation is our main tool:
  - Start with known solutions of a known system and then track those solutions as we deform the start system to the system that we wish to solve.



# Path Tracking

This method takes a system  $g(x) = 0$ , whose solutions we know, and makes use of a homotopy, e.g.,

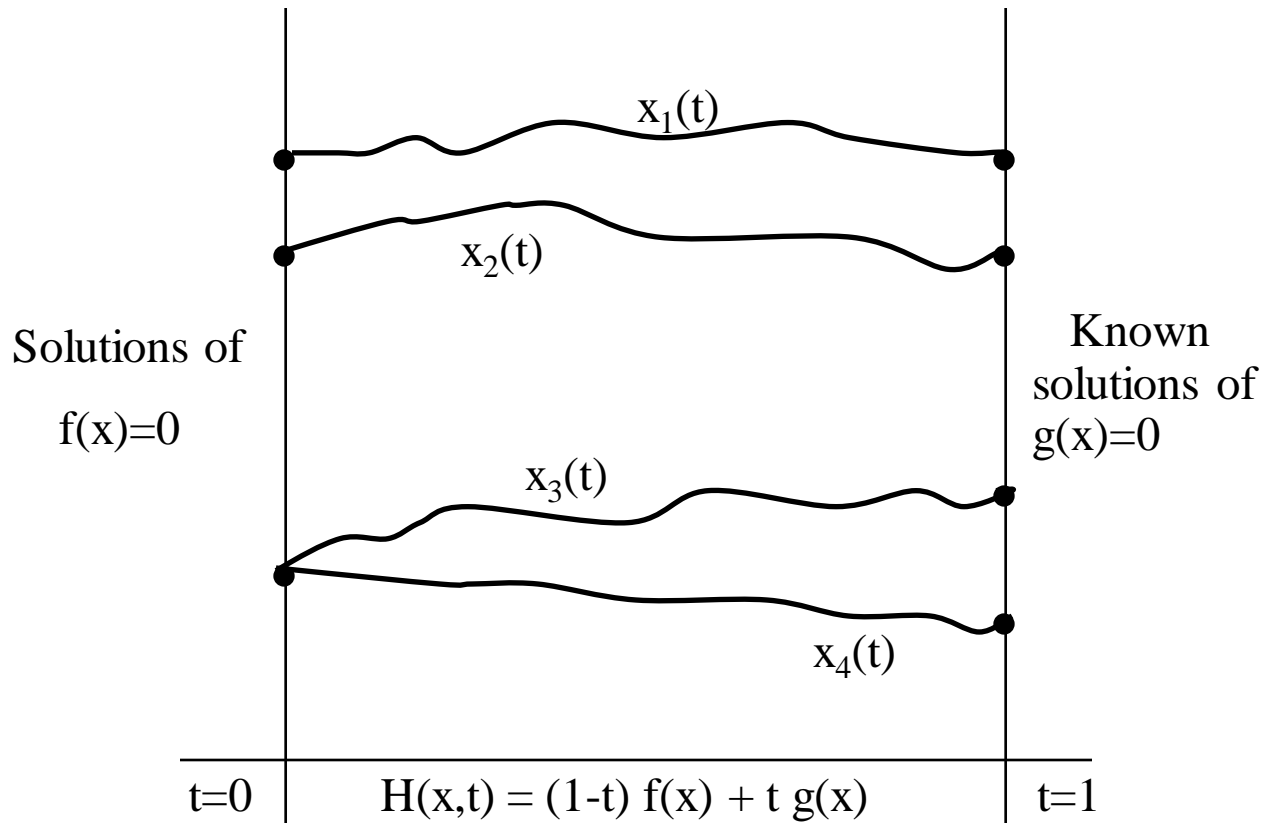
$$H(x, t) = (1 - t)f(x) + tg(x).$$

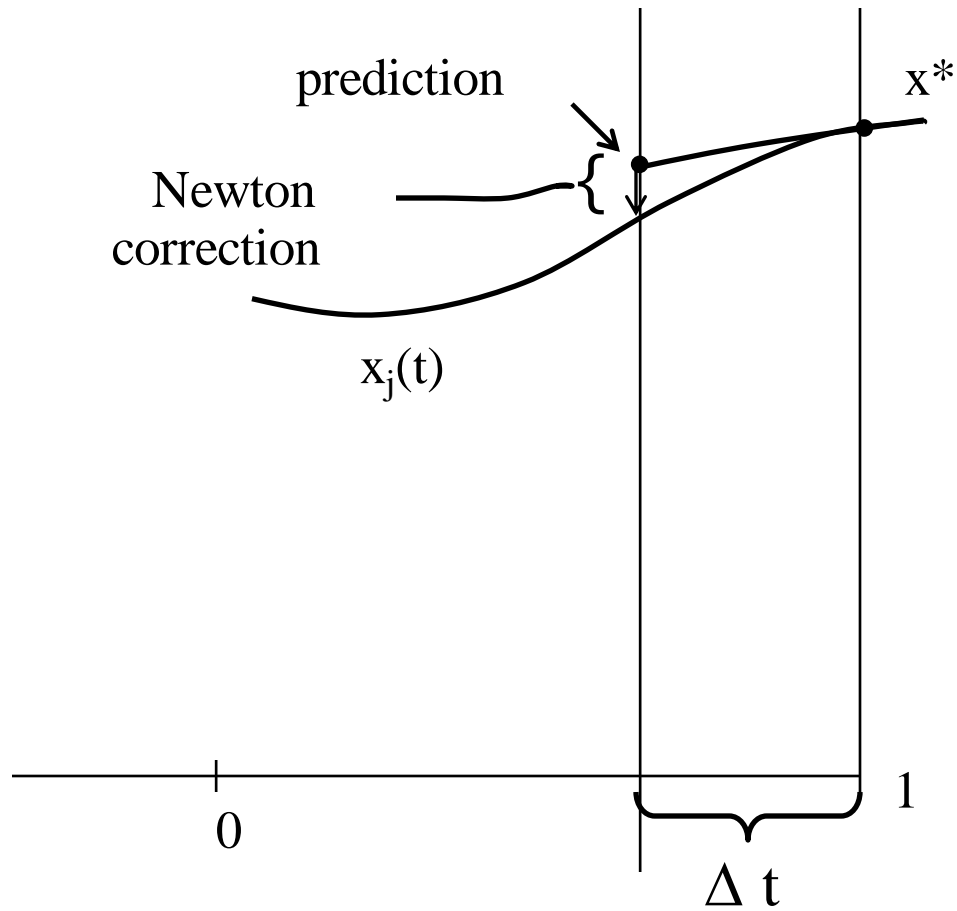
Hopefully,  $H(x, t) = 0$  defines “paths”  $x(t)$  as  $t$  runs from 1 to 0. They start at known solutions of  $g(x) = 0$  and end at the solutions of  $f(x)$  at  $t = 0$ .

- 
- The paths satisfy the Davidenko equation

$$0 = \frac{dH(\mathbf{x}(t), t)}{dt} = \sum_{i=1}^N \frac{\partial H}{\partial x_i} \frac{dx_i}{dt} + \frac{\partial H}{\partial t}$$

- To compute the paths: use ODE methods to predict and Newton's method to correct.



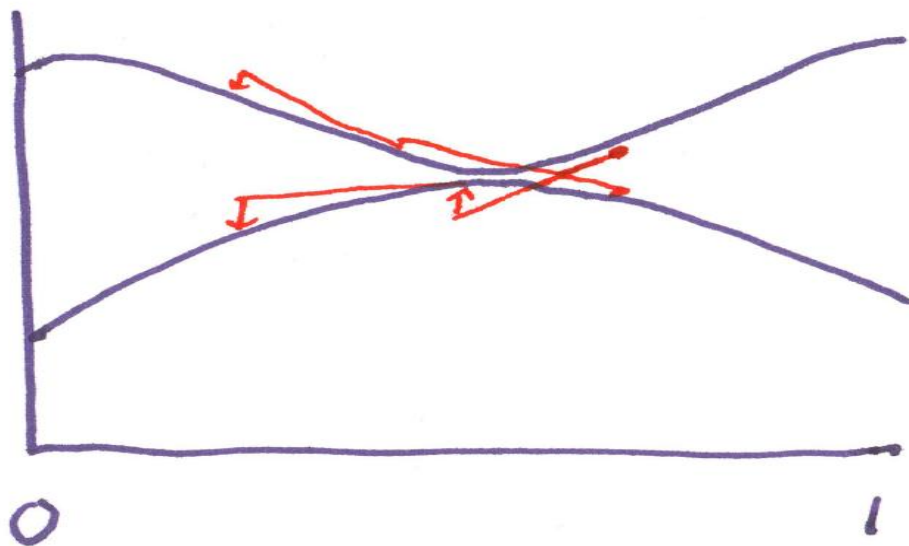


# What Continuation Computed

- 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  $f(x) = 0$ ;

$S$  was often larger than the set of isolated solutions.

Since the goal was to compute a finite set containing the solutions, path-crossing was not taken that seriously: it usually didn't prevent computing all solutions.



# 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.
  - 2006: about 10 paths a second on a single processor desktop CPU; 1000's of paths/second on moderately sized clusters.

*The inherent parallelizability has far-reaching consequences!*

# Algorithms

- **middle 80's**: Projective space was beginning to be used, but the methods were a combination of differential topology and numerical analysis.
- **Late 80's through early 90's**: algebraic geometric methods worked into the theory: great increase in security, efficiency, and speed.
- **middle 90's on**: positive dimensional sets and ever larger polynomial systems.

# Uses of algebraic geometry

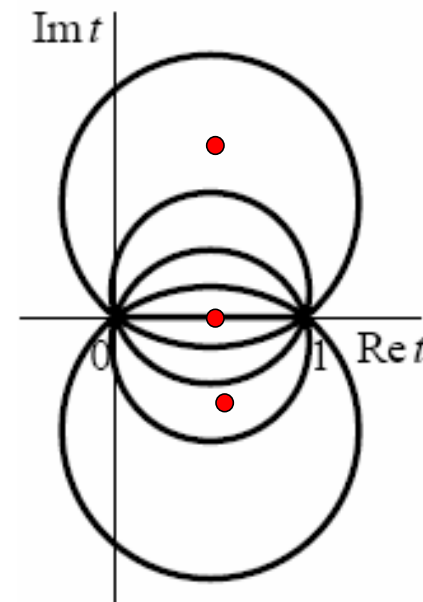
---

- Genericity and Bertini Theorems are a major source of the uses of Algebraic Geometry.

# One of the first applications

Simple but extremely useful consequence of algebraicity [A. Morgan (GM R. & D.) and S.]

- Instead of the homotopy  $H(x,t) = (1-t)f(x) + tg(x)$  use  $H(x,t) = (1-t)f(x) + \gamma tg(x)$



# Genericity giving a methodology

- Morgan + S. : if the parameter space is irreducible, solving the system at a random points simplifies subsequent solves: in practice speedups by factors of 100.
- A. Morgan and A.J. Sommese, *Coefficient-parameter polynomial continuation*, Appl. Math. Comput. 29 (1989), 123-160.

# First Major Use of the Methodology

- Kinematics Problem Posed in 1923 by Alt and solved in 1992.
  - C.W. Wampler, A. Morgan, and A.J. Sommese, **Complete solution of the nine-point path synthesis problem for four-bar linkages**, ASME Journal of Mechanical Design 114 (1992), 153-159.


# Alt's System

$$[(\hat{a} - \bar{\delta}_j)x]\gamma_j + [(a - \delta_j)\hat{x}]\hat{\gamma}_j + \delta_j(\hat{a} - \hat{x}) + \bar{\delta}_j(a - x) - \delta_j\bar{\delta}_j = 0$$

$$[(\hat{b} - \bar{\delta}_j)y]\gamma_j + [(b - \delta_j)\hat{y}]\hat{\gamma}_j + \delta_j(\hat{b} - \hat{y}) + \bar{\delta}_j(b - y) - \delta_j\bar{\delta}_j = 0$$

$$\gamma_j + \hat{\gamma}_j + \gamma_j\hat{\gamma}_j = 0$$

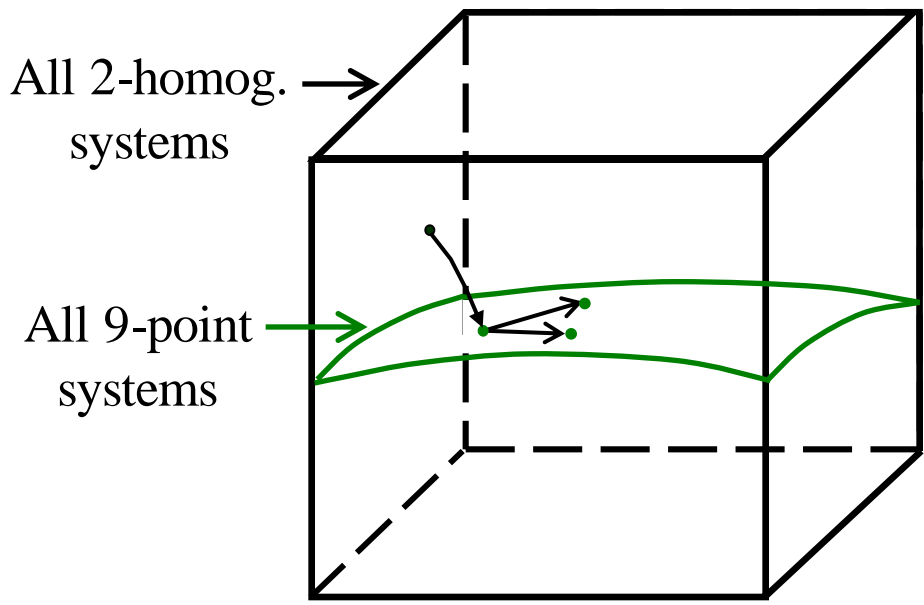
in the 24 variables  $a, b, x, y, \hat{a}, \hat{b}, \hat{x}, \hat{y}$  and  $\gamma_j, \hat{\gamma}_j$   
with  $j$  from 1 to 8.

- 
- 8 degree 2 and 16 degree 3 equations giving 11,019,960,801 paths to follow.
  - Freudenstein and Roth (early 50's): use Cramers rule and substitution on the  $\gamma$  variables, we have a system consisting of 8 equations of degree 7. In 1991, this was impractical to solve:

$$7^8 = 5,764,801 \text{ solutions.}$$



# Solve by Continuation



- “numerical reduction” to test case (done 1 time)
- synthesis program (many times)

# Nine-point Problem

## Summary

- **Analytical Reduction**
  - Initial formulation .....  $\approx 10^{10}$
  - Roth & Freudenstein ..... 5,764,801
  - Our elimination ..... 1,048,576
  - Multi-homogenization ..... 286,720
  - Symmetry ..... 143,360
- **Numerical Reduction**
  - Nondegenerate ..... 4326
  - Roberts cognates ..... 1442
- **Synthesis program tracks 1442 solution paths.**

# A point to consider

- Not all limits of paths are equal! Singular paths can be much more expensive and difficult to compute.

# A Guiding Principle

Use Special Homotopies to take advantage of sparseness

- Algorithms must be structured – when possible – to avoid paths leading to singular solutions: find a way to never follow the paths in the first place.

# Positive Dimensional Solution Sets

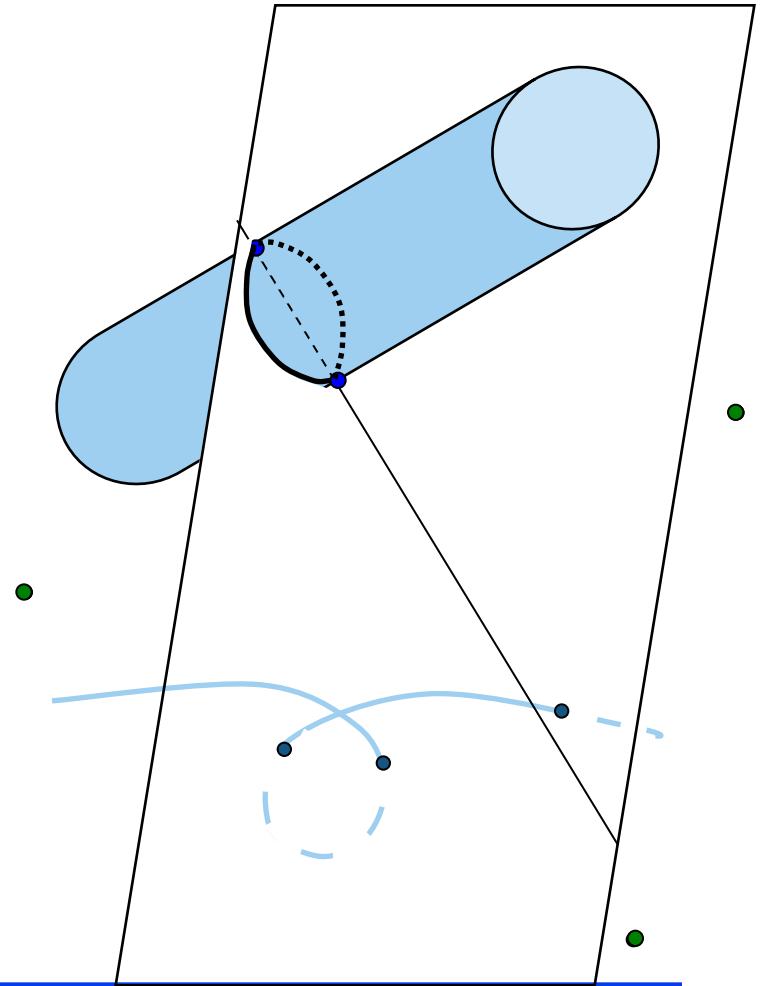
We now turn to finding the positive dimensional solution sets of a system

$$f(x) = \begin{bmatrix} f_1(x) \\ \vdots \\ f_n(x) \end{bmatrix} = 0$$

# How to represent positive dimensional components?

- S. + Wampler in '95:
  - Use the intersection of a component with generic linear space of complementary dimension.
  - By using continuation and deforming the linear space, as many points as are desired can be chosen on a component.

- Use a generic flag of affine linear spaces
- to get witness point supersets
- This approach has 19<sup>th</sup> century roots in algebraic geometry



# The Numerical Irreducible Decomposition

Carried out in a sequence of articles with Jan Verschelde (University of Illinois at Chicago) and Charles Wampler (General Motors Research and Development)

- Efficient Computation of “Witness Supersets”
  - S. and Verschelde, *Journal of Complexity* 16 (2000), 572-602.
- Numerical Irreducible Decomposition
  - S., Verschelde, and Wampler, *SIAM Journal on Numerical Analysis* 38 (2001), 2022-2046.
- An efficient algorithm using monodromy
  - S., Verschelde, and Wampler, *SIAM Journal on Numerical Analysis* 40 (2002), 2026-2046.
- Intersections of algebraic sets
  - S., Verschelde, and Wampler, *SIAM Journal on Numerical Analysis* 42 (2004), 1552-1571.



# The Core Numerical Computation

Realization slowly grew that path crossing is unacceptable and that the core numerical computation of Numerical Algebraic Geometry is:

- Given a homotopy  $H(x;q) = 0$ ; a “good” path  $q(t)$  in the  $q$ -variables defined on  $(0,1]$ ; and a point  $x^*$  satisfying  $H(x^*;q(1))=0$ , compute the limit as  $t$  goes to 0 of the path  $(x(t);q(t))$  starting with  $(x(1);q(1)) = (x^*;q(1))$  in the  $(x;q)$  space and satisfying  $H(x(t);q(t)) = 0$ .

In a nutshell: We need to compute the endpoint of  
a path!

# Numerical issues posed by multiple components

Consider a toy homotopy

$$H(x_1, x_2, t) = \begin{bmatrix} x_1^2 \\ x_2 - t \end{bmatrix} = 0$$

Continuation is a problem because the Jacobian with respect to the  $x$  variables is singular.

How do we deal with this?

# Deflation

The basic idea introduced by Ojika in 1983 is to differentiate the multiplicity away. Leykin, Verschelde, and Zhao gave an algorithm for an isolated point that they showed terminated.

Given a system  $f$ , replace it with

$$\begin{bmatrix} f(\mathbf{x}) \\ \mathbf{J}f(\mathbf{x}) \cdot \mathbf{z} \\ \mathbf{A} \cdot \mathbf{z} + \mathbf{b} \end{bmatrix} = 0$$

Repeat as necessary.



# Singular points cannot be avoided

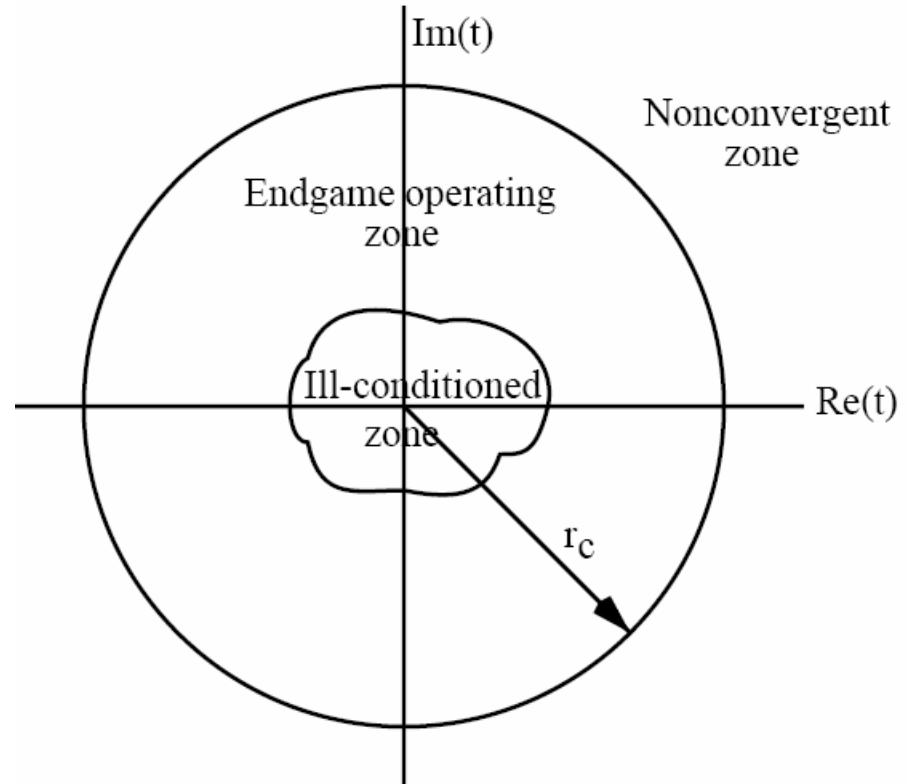
---

To make a viable algorithm for multiple components, it is necessary to make decisions on ranks of singular matrices.

# Endgames (Morgan, Wampler, and S.)

- Example:  $x(x - 1)^2 - t = 0$

We can uniformize around a solution at  $t = 0$ . Letting  $t = s^2$ , knowing the solution at  $t = 0.01$ , we can track around  $|s| = 0.1$  and use Cauchy's Integral Theorem to compute  $x$  at  $s = 0$ .



# Endgames in Non-traditional Situations

Singular solutions not a side issue, but the main object!

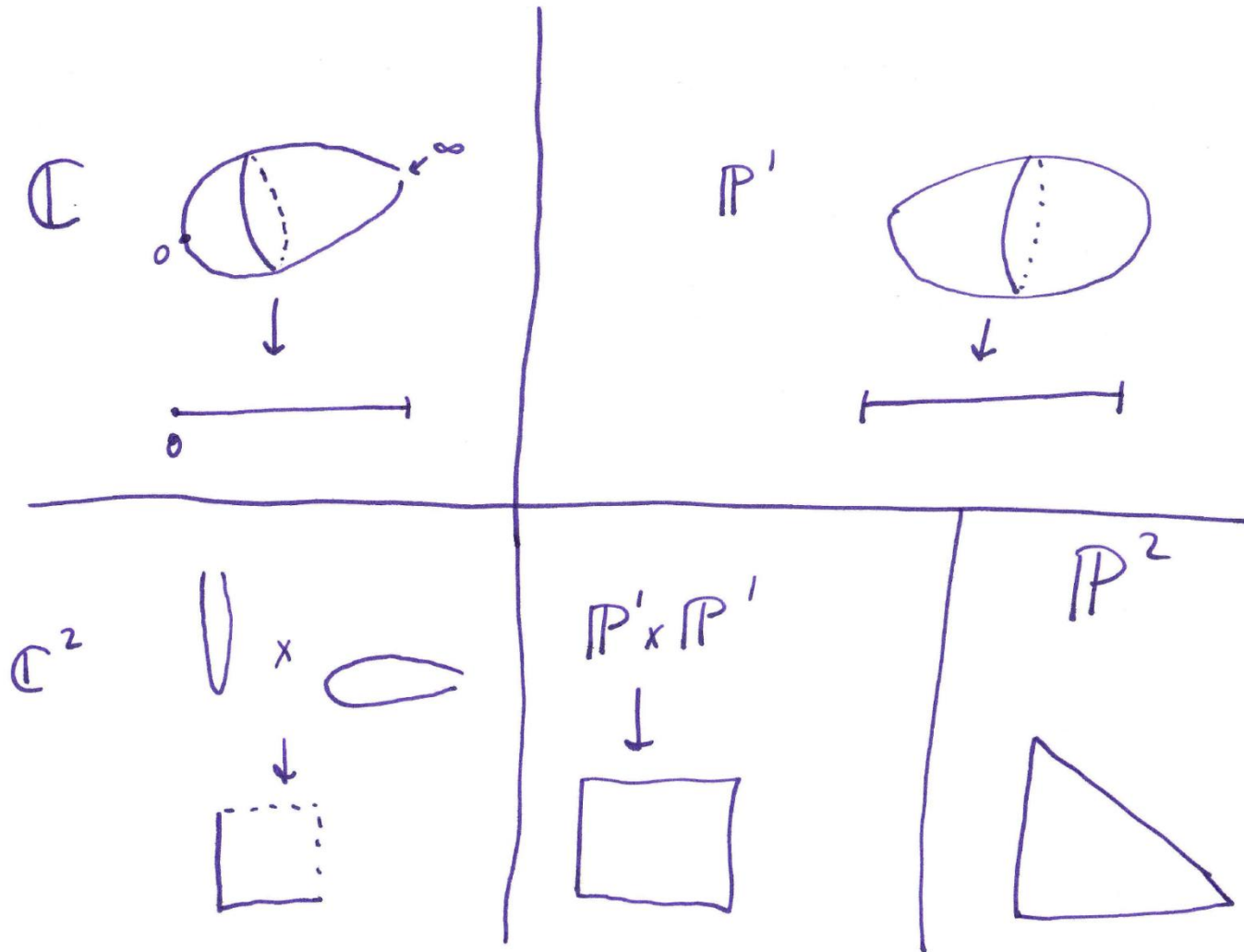
For certain classes of systems of hyperbolic PDEs and solutions with discontinuities, the Cauchy endgame gives an order of magnitude improvement over the standard time-stepping method.

- W. Hao, J.D. Hauenstein, C.-W. Shu, A.J. Sommese, Z. Xu, Y.-T. Zhang, *A homotopy method based on WENO schemes for solving steady state problems of hyperbolic conservation laws*, J. of Comp. Phys., 250 (2013), 332-346.

# Endgames

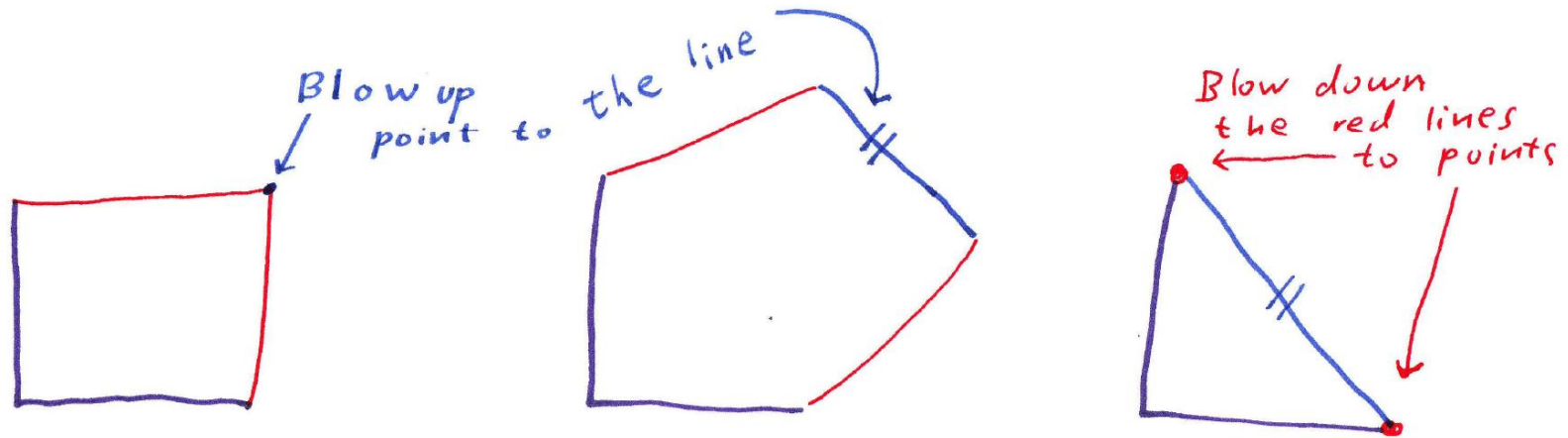
- Without multiprecision, singular points can only be computed reliably for low multiplicities.
- As a consequence, checking whether paths go to infinity is not secure!
  - Why?
  - Answer: we change coordinates to infinite endpoints are finite, but infinite endpoints often have extremely high multiplicities.

# Moment Map





# Sparse system treated as nonsparse

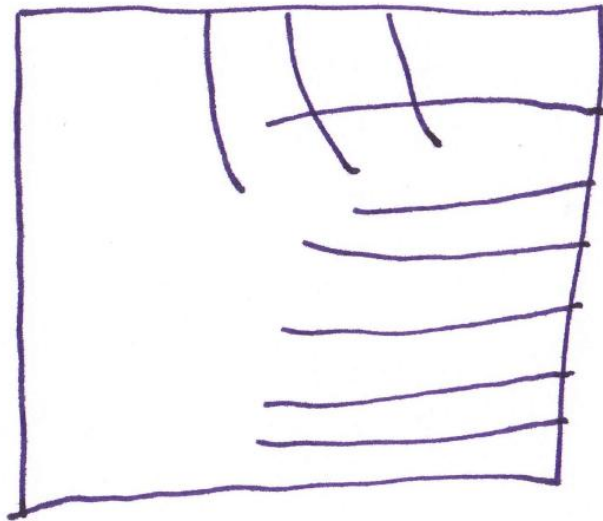


$C$  a curve of type  
3,6



$\tilde{C}$  a curve  
of degree 9  
and points of  
multiplicity 3 and 6

# Often high multiplicity solutions at infinity



# The need for multiprecision

- Why use Multiprecision?
  - to ensure that the region where an endgame works is not contained the region where the numerics break down;
  - to ensure that a polynomial is zero at a point is the same as the polynomial numerically being approximately zero at the point;
  - to prevent the linear algebra in continuation from falling apart; and
  - to prevent path crossing.

# Evaluation

$$p(z) = z^{10} - 28z^9 + 1$$

- To 15 digits of accuracy one of the roots of this polynomial is  $a = 27.999999999999999$ . Evaluating  $p(a)$  term-by-term to 15 digits, we find that  $p(a) = -2$  (or, evaluating intelligently  $p(a) = -0.05784559534077$ : this uses understanding we do not have in higher dimensions).
- Even with 17 digit accuracy, the approximate root is  $a = 27.99999999999999905$  and we still only have  $p(a) = -0.01$  (or with intell.:  $p(a) = -0.0049533155737293$ ).

# Wilkinson's Theorem from Numerical Linear Algebra

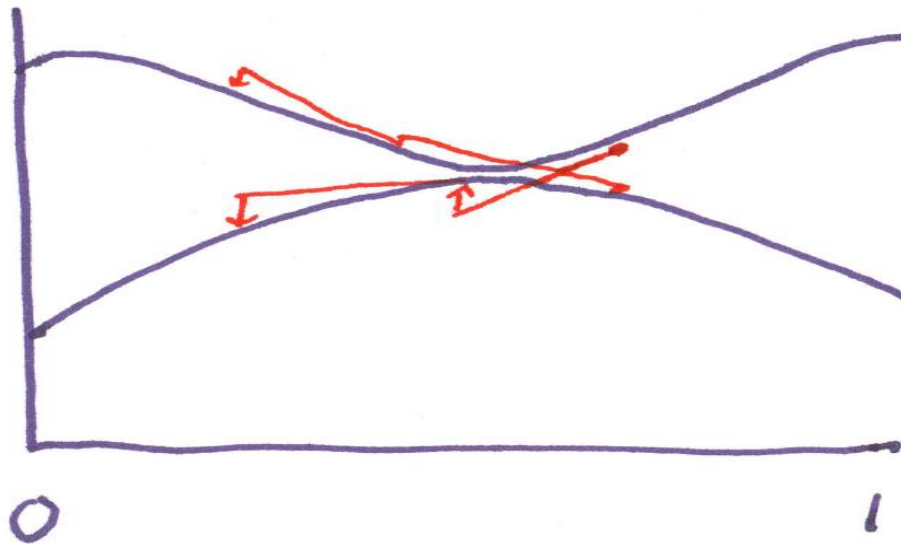
- Solving  $Az = f$ , with  $A$  an  $N$  by  $N$  matrix, we must expect to lose upto  $\log_{10}[\text{cond}(A)]$  digits of accuracy. Geometrically,  $\text{cond}(A) = \|A\| \cdot \|A^{-1}\|$  is on the order of the inverse of the distance in  $\mathcal{P}^{N \times N-1}$  from  $A$  to the set defined by  $\det(A) = 0$ .

## Double precision not enough even at nice solutions!

High condition numbers of  $10^7$  to  $10^9$  occur at nice solutions of discretizations of many systems of differential equations, e.g.,

- W. Hao, J.D. Hauenstein, B. Hu, Y. Liu, A.J. Sommese, and Y.-T. Zhang, **Continuation along bifurcation branches for a tumor model with a necrotic core**, J. Sci. Comp., 53 (2012), 395-413.
- W. Hao, J.D. Hauenstein, B. Hu, T. McCoy, and A.J. Sommese, **Computing steady-state solutions for a free boundary problem modeling tumor growth by Stokes equation**, J. Comp. and Appl. Math., 237 (2013), 326-334.
- W. Hao, B. Hu, and A.J. Sommese, **Cell cycle control and bifurcation for a free boundary problem modeling tissue growth**, J. Sci. Comp., 56 (2013), 350-365.

# Path-Crossing is dire in modern algorithms!



and isn't avoidable with only double precision

Near-singular conditions actually arise. For the current best polynomial system to solve Alt's problem:

- Out of 143,360 paths:
  - 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



# Using Higher Precision

- One approach is to simply run paths at a higher precision.

# Using Higher Precision

- One approach is to simply run paths at a higher precision.

This is computationally very expensive!

double (52 bits)	64 bits	96 bits	128 bits	256 bits	512 bits	1024 bits
2.447	32.616	35.456	35.829	50.330	73.009	124.401

From D.J. Bates, J.D. Hauenstein, A.J. Sommese, and C.W. Wampler, **Adaptive multiprecision path tracking**, SIAM Journal on Numerical Analysis 46 (2008) 722-746.




# Bertini


---

- Bertini uses data types modeled on the geometry and is designed to dynamically adjust the precision to achieve a solution with a prespecified error.
  - The current version of Bertini was developed by Daniel Bates, Jon Hauenstein, Charles Wampler, and Andrew Sommese.

# Bertini: available at [bertini.nd.edu](http://bertini.nd.edu)

- First released in Fall 2006. Written in C.
- Many strengths:
  - Very good adaptive precision tracker, which was refined and improved in successive releases.
  - Provision for user-defined homotopies.
  - Natural data structures, e.g., specification of paths in parameter spaces to track over is straightforward.


- 
- Many signs of age to some extent internally “corrected for”
    - Hard and (by today’s standards) very small limits on numbers of variables, number of equations.
    - No support for scripting or building new systems out of old systems.
    - Many different names for essentially the same functions with different inputs



Being redesigned in C++ by D.J. Bates, D.A. Brake, J.D. Hauenstein, A.J. Sommese, C.W. Wampler, and others.

Scripting facilities that allow

- concatenation of systems and building of new systems out of old without the user needing to have concern for the underlying internal evaluation scheme

- 
- Different research has led to distinct specialty versions of Bertini with special features, e.g.,
    - Support for sparse linear algebra and specification of polynomial systems of the sort arising in discretization of systems of differential equations.
    - Support for checking stability.
    - Support for studying systems depending on several parameters: for details, attend Dan Bates' talk.
    - Support for real algebraic decompositions: for details, attend Dan Brake's talk.

- 
- Designed to work with other programs.

## The article

- D.J. Bates, W. Decker, J.D. Hauenstein, C. Peterson, G. Pfister, F.-O. Schreyer, A.J. Sommese, and C.W. Wampler, **Probabilistic algorithms to analyze the components of an affine algebraic variety**, Applied Mathematics and Computation, 231 (2014), 619-633.

drove home for us that it is best to use symbolic programs for several stages in the analysis of a polynomial system.





---

Thank You!