Homotopies for **Real** Polynomial Systems

Jan Verschelde

University of Illinois at Chicago
Department of Mathematics, Statistics, and Computer Science
`http://www.math.uic.edu/~jan`
`jan@math.uic.edu`

AIM workshop on convex algebraic geometry
Palo Alto, 21-25 Sep 2009

Outline

1. Problem Statement: Real Homotopy Continuation
2. Khovanskiĭ-Rolle Continuation (Bates & Sottile)
   - Gale Duality
   - Khovanskiĭ-Rolle Theorem
3. Sweeping Algebraic Curves (Piret & Verschelde)
   - reconditioning singularities with deflation
   - detection and location of singular points
   - neural network model and symmetric Stewart-Gough platform
4. Morse-Like Representations (Lu, Bates, Sommese & Wampler)
   - definition of data structures
   - ingredients of the algorithms

Homotopy Continuation Methods

One commonly used *homotopy* to solve $f(\mathbf{x}) = \mathbf{0}$:

$$h(\mathbf{x}, t) = (1 - t)g(\mathbf{x}) + tf(\mathbf{x}) = \mathbf{0}, \quad t \in [0, 1],$$

where $g(\mathbf{x}) = \mathbf{0}$ is a good system with known start solutions.

Paths $\mathbf{x}(t)$ defined by $h(\mathbf{x}(t), t) = \mathbf{0}$ are tracked by *continuation*.

Solving over $\mathbb{C}$ has several benefits:

1. geometric interpretation: from generic to specific
2. we avoid singularities except possibly at end of the paths
3. algorithms for a numerical irreducible decomposition

For an introduction to **numerical algebraic geometry**:

A. J. Sommese and C.W. Wampler: *The Numerical Solution of Systems of Polynomials Arising in Engineering and Science.* World Scientific, 2005.

Real Problems

Solving over $\mathbb{R}$ for given system with real coefficients
means that we are mainly (or exclusively) interested in real solutions.

Some issues:

1. #real solutions $\ll$ #complex solutions
2. no longer enough genericity to avoid singularities
3. examples like $x^2 - y^2 z = 0$ complicate dimension count

Some answers:

1. Khovanskii-Rolle continuation for real isolated solutions
2. singularity detection and location when sweeping curves
3. Morse-like representation of a real algebraic curve

a new continuation algorithm

D.J. Bates and F. Sottile: *Khovanskii-Rolle Continuation for Real Solutions.* `arXiv:0908.4579v1 [math.AG] 31 Aug 2009`

On input is a square Laurent system of n equations, with $n + \ell + 1$ distinct monomials.

Two steps in the new continuation algorithm:

1. Set up master equations using Gale duality.
2. Apply the Khovanskiĭ-Rolle theorem.

Proof of concept implementation using Maple 13 and Bertini 1.1.1 for $\ell = 2$.

D.J. Bates, J.D. Hauenstein, A.J. Sommese, and C.W. Wampler: Bertini: Software for numerical algebraic geometry. Available at `http://www.nd.edu/~sommese`.

Gale Duality

$$\left\{ \begin{array}{rcl} cd & = & \gamma_{10} + \gamma_{11} be^2 + \gamma_{12} a^{-1}b^{-1}e \\ bc^{-1}e^{-2} & = & \gamma_{20} + \gamma_{21} be^2 + \gamma_{22} a^{-1}b^{-1}e \\ ab^{-1} & = & \gamma_{30} + \gamma_{31} be^2 + \gamma_{32} a^{-1}b^{-1}e \\ c^{-1}de^{-1} & = & \gamma_{40} + \gamma_{41} be^2 + \gamma_{42} a^{-1}b^{-1}e \\ bc^{-2}e & = & \gamma_{50} + \gamma_{51} be^2 + \gamma_{52} a^{-1}b^{-1}e \end{array} \right.$$

Laurent system
with few monomials :
$5 + 2 + 1$
$\gamma_{ij} \in \mathbb{R} \setminus \{0\}$

$$a^{-1}b^{-1}e \quad be^2 \quad cd \quad bc^{-1}e^{-2} \quad ab^{-1} \quad c^{-1}de^{-1} \quad bc^{-2}e$$

$$\begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} \begin{bmatrix} -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & 1 & 0 & 1 & -1 & 0 & 1 \\ 0 & 0 & 1 & -1 & 0 & 1 & -2 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 1 & 2 & 0 & -2 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} -1 & 1 \\ -2 & -3 \\ 1 & 6 \\ -2 & -2 \\ -1 & 1 \\ 1 & 6 \\ 2 & 7 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Master Functions

$$\begin{matrix} a \\ b \\ c \\ d \\ e \end{matrix} \begin{bmatrix} -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & 1 & 0 & 1 & -1 & 0 & 1 \\ 0 & 0 & 1 & -1 & 0 & 1 & -2 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 1 & 2 & 0 & -2 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} -1 & 1 \\ -2 & -3 \\ 1 & 6 \\ -2 & -2 \\ -1 & 1 \\ 1 & 6 \\ 2 & 7 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\begin{cases} (a^{-1}b^{-1}e)^{-1}\ (be^2)^{-2}\ (cd)^1\ (bc^{-1}e^{-2})^{-2}\ (ab^{-1})^{-1}\ (c^{-1}de^{-1})^1\ (bc^{-2}e)^2 = 1 \\ (a^{-1}b^{-1}e)^1\ (be^2)^{-3}\ (cd)^6\ (bc^{-1}e^{-2})^{-2}\ (ab^{-1})^1\ (c^{-1}de^{-1})^6\ (bc^{-2}e)^7 = 1 \end{cases}$$

Let $x = a^{-1}b^{-1}e$, $y = be^2$, then $cd = L_1(x, y)$, $bc^{-1}e^{-2} = L_2(x, y)$, $ab^{-1} = L_3(x, y)$, $c^{-1}de^{-1} = L_4(x, y)$, $bc^{-2}e = L_5(x, y)$.

An Equivalent System

$$\left\{ \begin{array}{l} (a^{-1}b^{-1}e)^{-1}\ (be^2)^{-2}\ (cd)^1\ (bc^{-1}e^{-2})^{-2}\ (ab^{-1})^{-1}\ (c^{-1}de^{-1})^1\ (bc^{-2}e)^2 = 1 \\ (a^{-1}b^{-1}e)^1\ (be^2)^{-3}\ (cd)^6\ (bc^{-1}e^{-2})^{-2}\ (ab^{-1})^1\ (c^{-1}de^{-1})^6\ (bc^{-2}e)^7 = 1 \end{array} \right.$$

There is a bijection between (a, b, c, d, e) and (x, y).

$$\left\{ \begin{array}{l} x^{-1}\, y^{-2}\, L_1^1\, L_2^{-2}\, L_3^{-1}\, L_4^1\, L_5^2 = 1 \\ x\, y^{-3}\, L_1^6\, L_2^{-2}\, L_3^1\, L_4^6\, L_5^7 = 1 \end{array} \right. \quad \text{or} \quad \left\{ \begin{array}{l} L_1^1\, L_4^1\, L_5^2 = x^1\, y^2\, L_2^2\, L_3^1 \\ x\, L_1^6\, L_3^1\, L_4^6\, L_5^7 = y^3\, L_2^2 \end{array} \right.$$

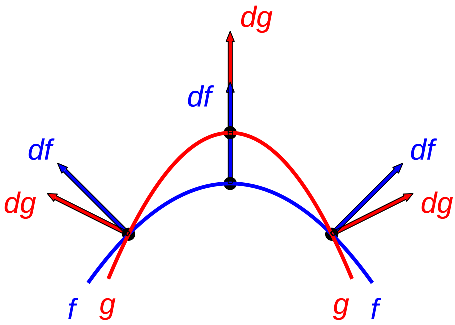
Positive solutions lie inside

$$\triangle := \{\, (x,y) \mid x > 0, y > 0, L_i(x,y) > 0 \,\}.$$

F. Bihan and F. Sottile: *Gale duality for complete intersections*.
Ann. Inst. Fourier 58(3): 877-891, 2008.

Khovanskiĭ-Rolle Theorem

Between any two zeroes of g along an arc of f, there is at least one zero of $\det(df \wedge dg)$.



A.G. Khovanskiĭ: *Fewnomials*, AMS 1991.

Khovanskiĭ-Rolle Continuation

Let f and g be the system of master functions in x and y.

1. Precomputation: solve the system $f = 0$ and $J = 0$,
   J is determinant of the Jacobian matrix.
2. Starting at solutions of $f = 0$ and $J = 0$ inside $\triangle$
   and solutions of $g = 0$ at the boundary of $\triangle$,
   trace curves to solutions of $f = 0$ and $g = 0$.

Complexity of precomputation is less than whole problem.
Every real solution is found twice.

Numerical difficulties:

- relatively high degree polynomials
- start solutions may be singular ...

a more extreme example

$$\begin{cases} 10500 - tu^{492} - 3500t^{-1}u^{463}v^5w^5 = 0 \\ 10500 - t - 3500t^{-1}u^{691}v^5w^5 = 0 \\ 14000 - 2t + tu^{492} - 3500v = 0 \\ 14000 + 2t - tu^{492} - 3500w = 0 \end{cases}$$

mixed volume: 7,663 counts #solutions in $(\mathbb{C}^*)^4$

however: only six positive real solutions

On a 2.83 Ghz computer running CentOs:

- PHCpack takes about 40 minutes to solve the system.
- Proof of concept implementation of the new continuation algorithm takes 23 seconds.

Sweeping Algebraic Curves

A **homotopy** h is a family of systems, depending on a parameter.
With **continuation** methods we track solution paths defined by h.
We distinguish between two types of parameters:

1. a natural parameter λ, for example:

$$h(\lambda, x) = \lambda^2 + x^2 - 1 = 0.$$

   As λ varies we track the unit circle: $(\lambda, x(\lambda)) \in h^{-1}(0)$.

2. an artificial parameter t, for example:

$$h(t, \lambda, x) = \begin{cases} \lambda^2 + x^2 - 1 = 0 \\ (\lambda - 2)t + (\lambda + 2)(1 - t) = 0. \end{cases}$$

   As t moves from 0 to 1, λ goes from -2 to $+2$
   and we ***sweep*** points $(\lambda(t), x(\lambda(t)))$ on the unit circle.

Reconditioning Singularities via Deflation

restoring the quadratic convergence of Newton's method

A solution $\mathbf{z}$ to $f(\mathbf{x}) = \mathbf{0}$, $f = (f_1, f_2, \ldots, f_N)$, $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, $N > n$, is **singular** if the Jacobian matrix $A(\mathbf{x}) = \left[\frac{\partial f_i}{\partial x_j} \right]$ has rank $R < n$ at $\mathbf{z}$.

Choose $\mathbf{c} \in \mathbb{C}^{R+1}$ and $\mathbf{B} \in \mathbb{C}^{n \times (R+1)}$ at random.
Introduce $R+1$ new multiplier variables $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_{R+1})$.
Apply the Gauss-Newton method to

$$\left\{ \begin{array}{rcl} f(\mathbf{x}) & = & \mathbf{0} \\ A(\mathbf{x})\mathbf{B}\boldsymbol{\mu} & = & \mathbf{0} \\ \mathbf{c}\boldsymbol{\mu} & = & 1 \end{array} \right. \qquad \begin{array}{c} \mathrm{Rank}(A(\mathbf{x})) = R \\ \Downarrow \\ \mathrm{coRank}(A(\mathbf{x})\mathbf{B}) = 1 \end{array}$$

Recurse if necessary, $\#$*deflations* $<$ multiplicity.
An efficient implementation uses algorithmic differentiation.

A. Leykin, J. Verschelde, and A. Zhao: *Newton's method with deflation for isolated singularities of polynomial systems.* Theoretical CS 2006.

Quadratic Turning Points

most common type of singularity

1. **Definition:** solution paths turn back
   when the parameter increases past a quadratic turning point.

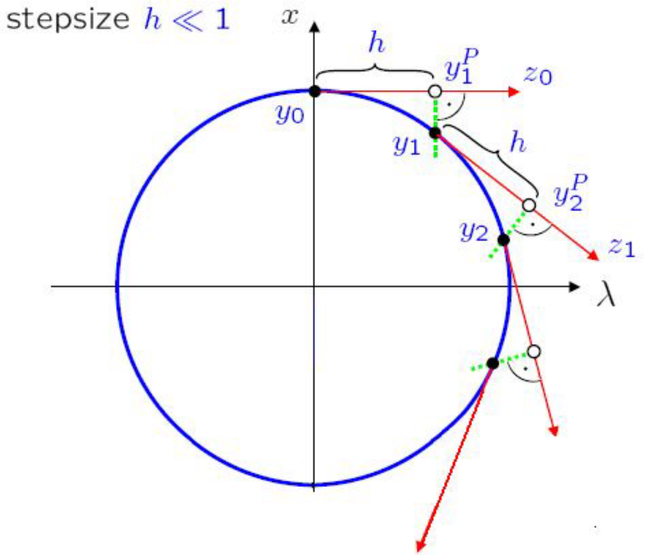
   Properties: a double solution, corank of Jacobian equals one,
   transition point: complex $\leftrightarrow$ real.

2. **Detection:** monitor orientation of tangent vectors.
   Two consecutive tangent vectors $\mathbf{v}(t_1)$ and $\mathbf{v}(t_2)$.
   Criterion: $\langle \mathbf{v}(t_1), \mathbf{v}(t_2) \rangle < 0 \Rightarrow \mathbf{v}(t) \perp t-\mathrm{axis}$ for $t \in [t_1, t_2]$.
   Tangents are simple byproduct of predictor-corrector path tracker.

3. **Location:** shooting method for step size.
   Consider $\mathbf{x}(t) = \mathbf{x}(t_1) + h\,\mathbf{v}(t_1)$, find h and t: $\mathbf{v}(t) \perp t-\mathrm{axis}$.
   Overshot turning point for $h = h_2$, at $\mathbf{x}(t_2)$ path has turned back.

T.Y. Li and Z. Zeng: *Homotopy continuation algorithm for the real nonsymmetric eigenproblem: Further development and implementation.* SIAM J. Sci. Comput. 1999.

Sweeping a Circle



Difficulties to Extend Approach

for any type of isolated singularity along a path

Detecting and locating quadratic turning points goes well.

Extending to any type of singularity has two difficulties:

1. detection: flip of tangent orientation no longer suffices
   $\rightarrow$ the path tracker glides over the singularity
2. location: higher order singularities may have corank > 1
   $\rightarrow$ the path tracker fails to converge

Solutions for these difficulties:

1. use a Jacobian criterion for detection, and
2. algebraic higher order predictor for location.

K. Piret and J. Verschelde: *Sweeping Algebraic Curves for Singular Solutions*. To appear in Journal of Comput. and Appl. Math.

Neural Network Model

a family of polynomial systems for any dimension n

V.W. Noonburg. *A neural network modeled by an adaptive Lotka-Volterra system.* SIAM J. Appl. Math. 1989.

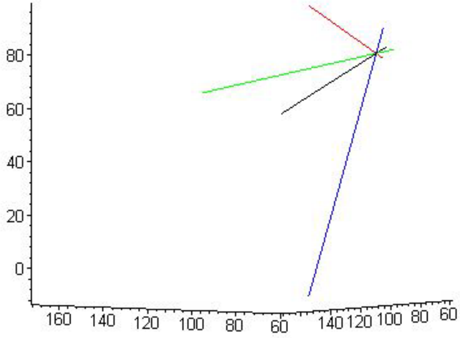
- Applying a sweep to the polynomial systems:

$$f(x,\lambda) = \left\{ \begin{array}{l} x_1x_2^2 + x_1x_3^2 - \lambda x_1 + 1 = 0 \\ x_2x_1^2 + x_2x_3^2 - \lambda x_2 + 1 = 0 \\ x_3x_1^2 + x_3x_2^2 - \lambda x_3 + 1 = 0 \\ (\lambda+1)(1-t) + (\lambda-1)t = 0 \end{array} \right.$$

- As t goes from 0 to 1, λ goes from -1 to $+1$.
- The tangent does not flip at the origin. The path tracker does not detect the quadruple point for $\lambda = 0$.

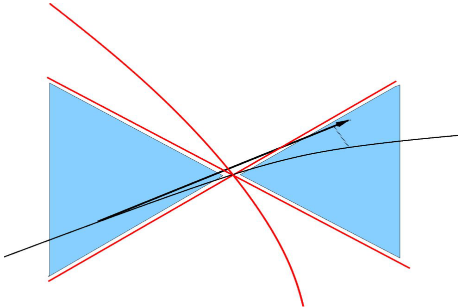
The Plot of Solution Paths for Neural Networks

the solution paths are really straight



Jumping Over Singularities

Z. Mei: *Numerical Bifurcation Analysis for Reaction-Diffusion Equations*. Springer, 2000.



The shaded blue part is the region where Newton's method converges. On straight curves, the path tracker will never cut back its step size.

Detection Algorithm Specification

Input: $h(\mathbf{x}, t) = \mathbf{0}$;	*a homotopy*
(t_1, t_2, t_3), $t_1 < t_2 < t_3$;	*consecutive samples*
$(\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3)$: $h(\mathbf{z}_i, t_i) = \mathbf{0}$, $i = 1, 2, 3$;	*with solutions*
(d_1, d_2, d_3): $d_i = \det(\partial_{\mathbf{x}} h(\mathbf{z}_i, t_i))$, $i = 1, 2, 3$;	*and determinants*
$\delta > 0$;	*tolerance on* $t_3 - t_1$
$\epsilon > 0$.	*tolerance on* $\det()$

Output: $(t^*, \mathbf{z}^*, d^*)$, $h(\mathbf{z}^*, t^*) = \mathbf{0}$;	*a solution*
$d^* = \det(\partial_{\mathbf{x}} h(\mathbf{z}^*, t^*))$, $\|d^*\| < \epsilon$;	*that is singular*
or $\emptyset$, updated $(t_i, \mathbf{z}_i, d_i)$, $i = 1, 2, 3$.	*no singular solution*

Detection Algorithm Implementation

while $(\|d_1\| > \|d_2\| < \|d_3\|)$ and $(t_3 - t_1 > \delta)$ do	*loop invariants*
$t^* := \min \mathcal{P}((t_1, t_2, t_3), (d_1, d_2, d_3))$;	*parabolic minimum*
$(z^*, d^*) := \mathrm{Newton}(h, t^*, \mathbf{z}_2)$;	*correct solution*
if $\|d^*\| < \epsilon$ then	*first stop test*
return $(t^*, \mathbf{z}^*, d^*)$;	*found singularity*
else if $\|d^*\| \geq \|d_2\|$ then	*second stop test*
return $\emptyset$;	*no singularity found*
else	*continue loop*
if $t^* < t_2$	*adjust t_1, t_2, t_3*
then $(t_3, \mathbf{z}_3, d_3) := (t_2, \mathbf{z}_2, d_2)$;	*t_2 becomes right end*
else $(t_1, \mathbf{z}_1, d_1) := (t_2, \mathbf{z}_2, d_2)$;	*t_2 becomes left end*
end if;	
$(t_2, \mathbf{z}_2, d_2) := (t^*, \mathbf{z}^*, d^*)$;	d_2 remains minimum
end if;	
end while.	

Numerical Stability

For determinant values d_1, d_2, and d_3, respectively at consecutive t_1, t_2, and t_3, $t^* := \min \mathcal{P}((t_1, t_2, t_3), (d_1, d_2, d_3))$ is subject to roundoff error. t^* is computed via

$$T = \frac{t_1^2(d_3 - d_2) + t_2^2(d_1 - d_3) + t_3^2(d_2 - d_1)}{2d_1(t_2 - t_3) + 2d_2(t_3 - t_1) + 2d_3(t_1 - t_2)}.$$

We compute $\widetilde{T}$, replacing in T d_1, d_2, and d_3 respectively by $d_1(1 + \epsilon_1)$, $d_2(1 + \epsilon_2)$, and $d_3(1 + \epsilon_3)$ for errors ϵ_1, ϵ_2, and ϵ_3.

$$\frac{\widetilde{T} - T}{T} = \frac{2\epsilon_1 d_1 t_{23} + 2\epsilon_2 d_2 t_{13} + 2\epsilon_3 d_3 t_{12}}{P}.$$

with t_{23}, t_{13}, and t_{12} constants of magnitude $> \delta$ and $P = t_1^2(d_3 - d_2) + t_2^2(d_1 - d_3) + t_3^2(d_2 - d_1)$.

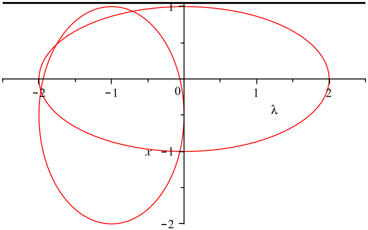
$\Rightarrow$ large relative errors only if $d_1 \approx d_2 \approx d_3$.

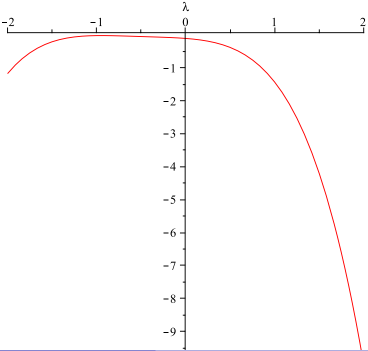
Numerical Conditioning

Worst case: straight path almost touches ellipses.

$$
h(x,\lambda,t) = \left\{ \begin{array}{rcl} (x - 1 - \epsilon)\left(\frac{\lambda^2}{4} + x^2 - 1\right) & & \\ \left(\frac{1}{4}(\lambda+1)^2 + \frac{4}{9}(x+1/2)^2 - 1\right) & = & 0 \\ (1-t)(\lambda+2) + t(\lambda-2) & = & 0 \end{array} \right. \qquad t \in [0,1].
$$

Plots for $\epsilon = 0.05$:





Polynomial Systems

the number of solutions in C^n for generic choices of parameters

Polynomial Systems	n	#Solutions
Molecular Configurations	3	16
Neural Networks	3	21
Neural Networks	4	73
Neural Networks	5	233
Neural Networks	10	59049
Neural Networks	15	14,348,907
Symmetrical Stewart-Gough Platforms	9	28 (real)

Table: Polynomial Systems which have higher-order multiple points

Molecular Configurations

applying the sweep homotopy algorithm to this system

I.Z. Emiris and B. Mourrain: *Computer algebra methods for studying and computing molecular conformations.* Algorithmica 1999.

- Applying a sweep to molecular configurations:

$$f(x,\lambda) = \left\{ \begin{array}{l} \frac{1}{2}(x_2^2 + 4x_2x_3 + x_3^2) + \lambda(x_2^2x_3^2 - 1) = 0 \\ \frac{1}{2}(x_3^2 + 4x_3x_1 + x_1^2) + \lambda(x_3^2x_1^2 - 1) = 0 \\ \frac{1}{2}(x_1^2 + 4x_1x_2 + x_2^2) + \lambda(x_1^2x_2^2 - 1) = 0 \\ (\lambda - 1)(1 - t) + (\lambda + 1)t = 0. \end{array} \right.$$

- The tangent flips at the higher-order turning point at the origin.
- For $\lambda = \pm 0.866025403780023$ on symmetrical curves of degree 6 and one of the eigenvalues of the Jacobian matrix changes signs.

Symmetrical Stewart-Gough platforms

nine quadratic polynomial equations

$$f(x, L_1) = \begin{cases} f_i = (x_i - x_{i0})^2 + (y_i - y_{i0})^2 + z_i^2 - L_i^2, i = 1, 2, \ldots, 6 \\ f_7 = (x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2 - 2R_1^2(1 - \beta)) \\ f_8 = (x_1 - x_0)^2 + (y_1 - y_0)^2 + (z_1 - z_0)^2 - R_1^2 \\ f_9 = (x_2 - x_0)^2 + (y_2 - y_0)^2 + (z_2 - z_0)^2 - R_1^2 \end{cases}$$

$$\text{where} \quad \begin{cases} x_i = w_1 x_0 + w_2^{m_1} w_3^{m_2} x_1 + w_2^{m_2} w_3^{m_1} x_2 \\ y_i = w_1 y_0 + w_2^{m_1} w_3^{m_2} y_1 + w_2^{m_2} w_3^{m_1} y_2 \\ z_i = w_1 z_0 + w_2^{m_1} w_3^{m_2} z_1 + w_2^{m_2} w_3^{m_1} z_2 \end{cases}$$

Yu Wang and Yi Wang: *Configuration Bifurcations Analysis of Six Degree-of-Freedom Symmetrical Stewart Parallel Mechanism.* Journal of Mechanical Design 2005.

Computational Results

on the symmetrical Stewart-Gough platforms

- Applying the Jacobian criterion globally leads to an augmented system with a mixed volume equal to 4,608.
  Tracking 4,608 paths in 16 variables is much more expensive than tracking 512 paths in 9 variables.
  Sweeping to find all critical points works in a more efficient setup: at most 28 paths in 9 variables.
- By fixing $L_i, i = 2, 3, \ldots, 6$, to 1.5, 2.0, and 3.0, the sweep yields four special values for the natural parameter L_1 for each L_i.
- We have replicated the results from Wang and Wang's paper with higher precision than what they reported.
  In addition, z_0 can be either positive or negative.

Morse-Like Representations of a Real Algebraic Curve

Given $f(\mathbf{x}) = \mathbf{0}$, a real polynomial system.

$$Z_1(f) = \{ \text{ all irreducible 1-dimensional solution sets in } \mathbb{C}^n \ \}$$

$$\begin{array}{rcl} Z_{1\mathbb{R}}(f) & = & Z_1(f) \cap \mathbb{R}^n \\ & = & \{ \text{ isolated real points on complex curves } \} \\ & \cup & \{ \text{ 1-dimensional real connected components } \} \end{array}$$

In addition to computing $Z_{1\mathbb{R}}(f)$, algorithms and data structures solve the membership problem:

1. does a solution belong to $Z_{1\mathbb{R}}(f)$?
2. to which real connected component does it belong to?

Y. Lu, D.J. Bates, A.J. Sommese, C.W. Wampler: *Finding all real points of a complex curve.* Contemporary Mathematics 448: 183-206, 2007.

Data Structure

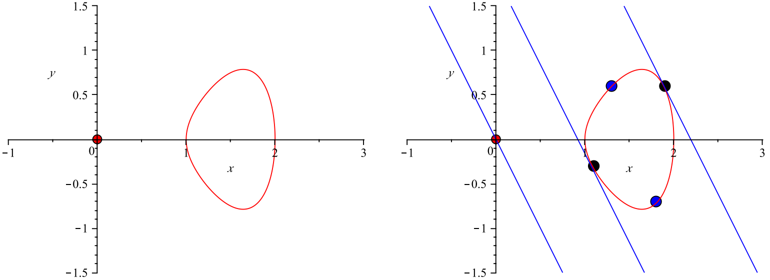
A Morse-like representation of a real algebraic curve $C_{\mathbb{R}} \subset \mathbb{R}^n$ consists of

1. a generic linear projection $\pi : \mathbb{R}^n \rightarrow \mathbb{R}$;
2. a boundary point set $\mathcal{B}_{\mathbb{R}} = \{ B_1, B_2, \ldots, B_m \}$, $B_i \in \mathbb{R}^n$ for all i;
3. an edge set $E = \{ E_1, E_2, \ldots, E_r \}$, for all $k \in \{1, 2, \ldots, r\}$:
   $E_k = (\ell_k, r_k, \mathbf{x}_k) \in (\mathbb{Z} \cup \{-\infty\}) \times (\mathbb{Z} \cup \{+\infty\}) \times \mathbb{R}^n$, where
   1. B_{ℓ_k} and B_{r_k} are left and right end points of edge e_k,
      if e_k extends to infinity to the left and/or right,
      then $\ell_k = -\infty$ and/or $r_k = +\infty$;
   2. $\mathbf{x}_k \in e_k$ over a point of $\pi(e_k)$.

an illustration

$$f(x, y) = y^2 + x^2(x-1)(x-2) = 0$$

$Z_{1\mathbb{R}}(f)$ consists of $(0,0)$ and one bounded curve.



The bounded curve of $\mathbb{Z}_{1\mathbb{R}}(f)$ is represented by two edges.

Ingredients of the Algorithms

assuming reduced complex curve

1. $Z_1(f)$ is computed via the solutions of

$$\left\{ \begin{array}{r} f(\mathbf{x}) = \mathbf{0} \\ c_0 + \mathbf{c}^T\mathbf{x} = 0 \end{array} \right. \qquad (c_0, \mathbf{c}) \in \mathbb{C}^{n+1}.$$

   The hyperplane defined by $(c_0, \mathbf{c})$ and the solutions of $f(\mathbf{x}) = \mathbf{0}$ on the hyperplane give *a witness set* W for $Z_1(f)$.

2. The boundary point set $\mathcal{B}_\mathbb{R}$ is obtained via global deflation

$$\left\{ \begin{array}{c} f(\mathbf{x}) = \mathbf{0} \\ J_f(\mathbf{x})Bz + \Lambda \left[\begin{array}{c} t_1 c_1 z \\ t_2 c_2 z \\ \vdots \\ t_{n-2} c_{n-2} z \end{array} \right] = \mathbf{0} \\ \gamma z - 1 = 0 \end{array} \right. \qquad \begin{array}{c} J_f = \left[\frac{\partial f}{\partial \mathbf{x}} \right] \\ B \in \mathbb{R}^{n \times (n-1)} \\ \Lambda \in \mathbb{C}^{(n-1)\times(n-2)} \\ \gamma \in \mathbb{C} \\ \text{cascade } t_i = 1 \rightarrow 0 \end{array}$$

computing Morse-like representations

Once boundary point set $\mathcal{B}_\mathbb{R}$ is computed, do

1. Sort $\mathcal{B}_\mathbb{R} = \{ B_1, B_2, \ldots, B_m \}$: $\pi(B_i) < \pi(B_{i+1})$, $i = 1, 2, \ldots, m-1$.
2. Starting at points in witness set W for $Z_1(f)$, compute points $\mathbf{x}_k$ on edges e_k, with homotopy

$$\left\{ \begin{array}{c} f(\mathbf{x}) = \mathbf{0} \\ (1-t)(c_0 + \mathbf{c}^T\mathbf{x}) + t(\pi(\pi_W(\mathbf{x})) - s) = 0, \quad t \in [0,1] \end{array} \right.$$

   for all midpoints $s = (\pi(B_i) + \pi(B_{i+1}))/2$, $i = 1, 2, \ldots, m-1$.
3. For every $\mathbf{x}_k$ on edge e_k, use homotopy to track to left and right end point to compute ℓ_k and r_k of E_k.

Applications and Extensions

Application to a special Griffis-Duffy platform.

- A Stewart-Gough platform with special positions of ball joints at base and end plate.
- Direct position problem described by a polynomial system of 7 homogeneous polynomial equations.
- Special scaled mechanism with multiple components.

For an extension to finding real points on surfaces, see Chapter 4 of Ye Lu: *Finding all real solutions of polynomial systems*. PhD Thesis, University of Notre Dame, 2006.

Conclusions

- computing real solutions involves searching for singularities
- many numerical challenges and complexity issues
- towards a numerical cylindrical algebraic decomposition?