Topology Determination of Algebraic Curves and Surfaces and Applications

Xiao-Shan Gao

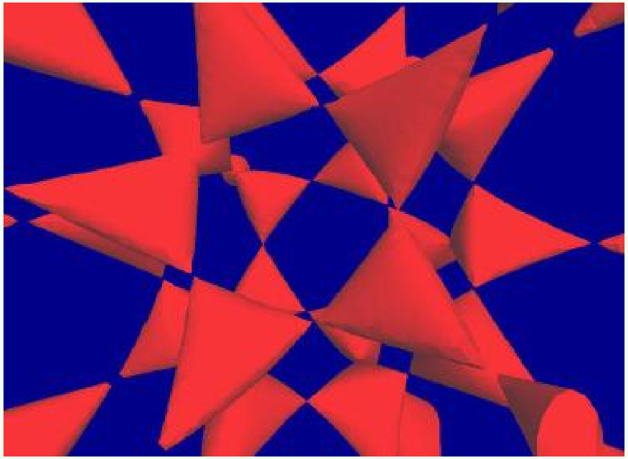
Institute of Systems Science,
Chinese Academy of Sciences

Outline

- *Introduction*
- *Algebraic curves*
- *Algebraic surfaces*
- *Root isolating for triangular systems*

1. The Problem

- Given $h(x) = 0/f(x, y) = 0/f(x, y, z) = 0$, to determine the topology of the point-set/curve/surface.
- Give trustworthy meshing or approximation to the point-set/curve/surface.
- Trustworthy: correct topology and with arbitrary precision.



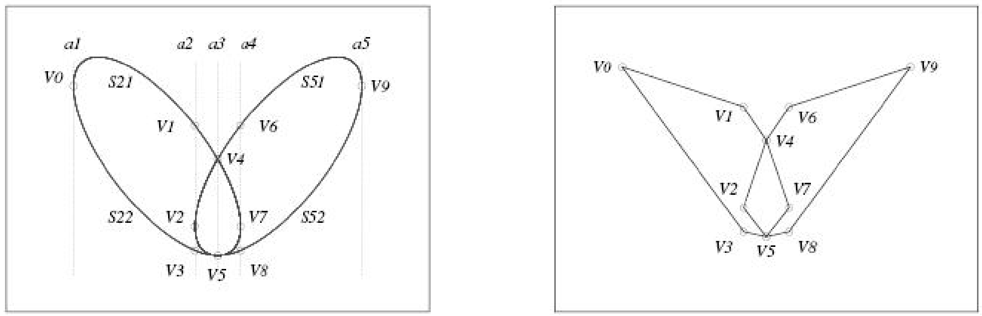
A degree six algebraic curves

Challenge: Deal with singularities

Examples of Topology Determination

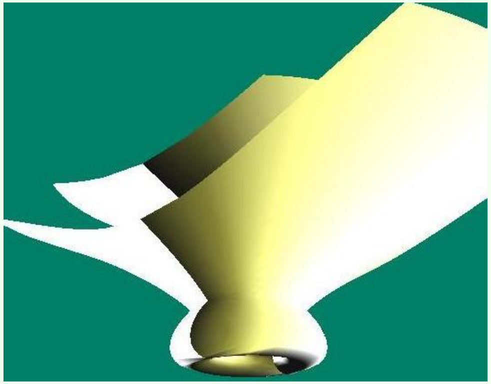
- **h(x)=0**: Real root isolation.

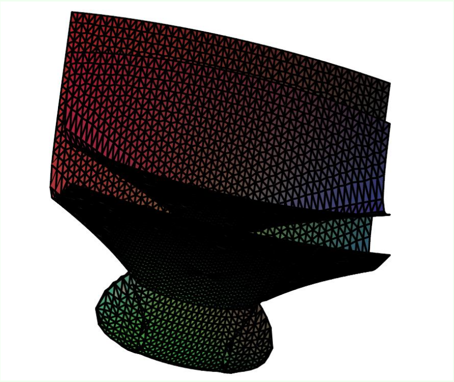
- **f(x,y)=0**: Meshing/Polygonization



A curve and its topology graph

- **g(x,y,z)=0**: Meshing/Polyhedronization





Related Work

CAD(Cylindrical Algebraic Decomposition) + Adjacency ⇒ Topology

(Collins G., Arnon P., McCallum S., Hong H., et al)

(Adjacency: Kozen and Yap, 1985; Prill, 1988; McCallum, 2002)

- **Curves:**
  - Hong, H., (MCS) 1996: plane curves
  - Gonzalez-Vega, L. and El Kahoui, M. (CAGD) 2002: plane curves
  - Gatellier G. et all, (MEGA) 2004: spatial curves
  - Alcazar J.G. and Sendra J.R., (JSC) 2005: spatial curves
  - Eigenwillig A., Kerber M. and Wolpert N., (ISSAC) 2007: plane algebraic curves.

Related Work (continued)

- **Surfaces:**
  - **Marching cube**: non-singular. Snyder, 1992; Plantinga and Vegter, 2004.
  - **Morse theory**: non-singular and orientable. Ni & Hart, 1998, 2004; Fortuna et al, 2004.
  - **Delaunay triangulation**: non-singular. Boissonnat and Oudot, 2003, Cheng, Dey et al, 2004.
  - **Projection:** general surface. Mourrain B. and Tecourt J.P., 2005.

2. Algebraic Curves

Main Steps of Projection Method

Projection. Project the "critical points" $f(x,y) = f_y(x,y) = 0$ to the x-axis.

Compute resultant $h = \text{Resl}(f, f_z)$.

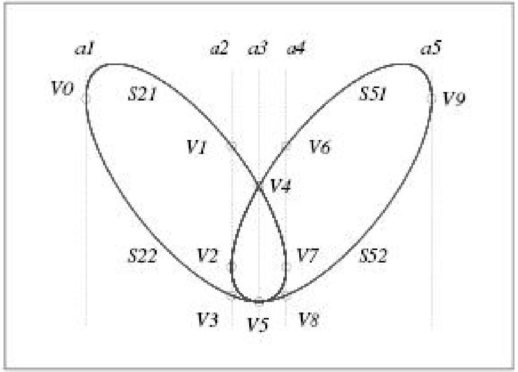
Isolate roots of $h(x) = 0$: $\alpha_0 < \alpha_1 < \cdots < \alpha_s$.

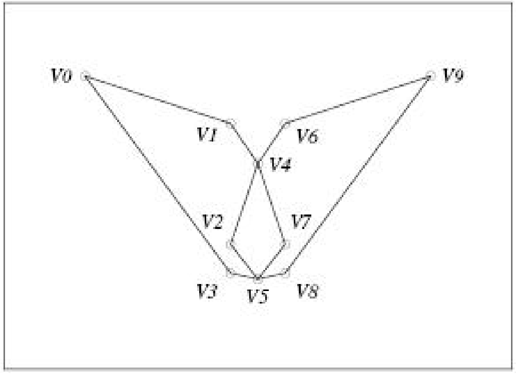
Lifting. Find the "critical points" $P_{i,j} = (\alpha_i, \beta_j)$.

Isolate roots for triangular systems: $h(x) = 0, f(x,y) = 0$.

Adjacency Determination. Determine whether $P_{i,j}$ connects $P_{i+1,k}$: $e_{i,j,k} = P_{i,j}P_{i+1,k}$.

Determine the **topology graph**: $\{P_{i,j}, e_{i,j,k}\}$.



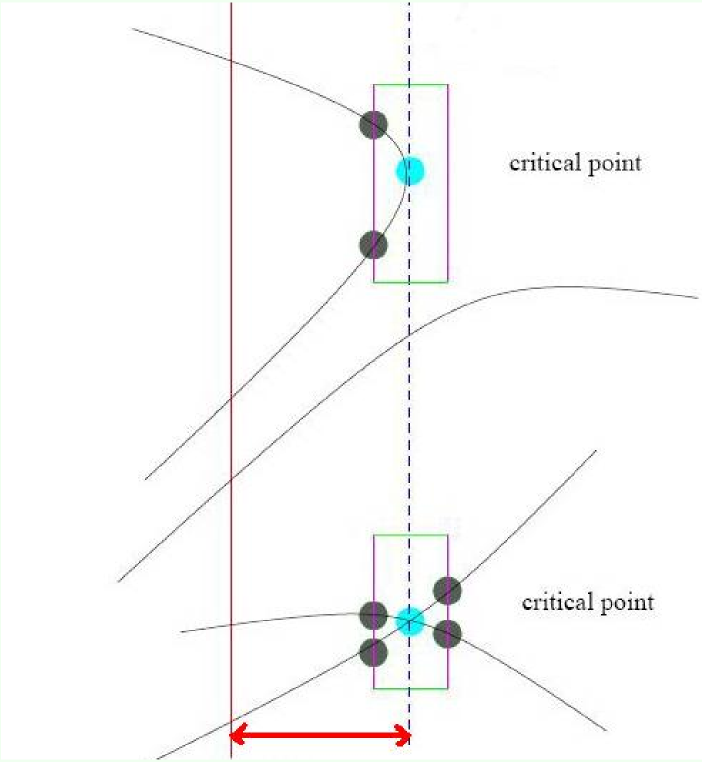


A curve and its topology graph

Key Issues

- Root Isolation for triangular systems: part 4.
- Adjacency determination: a new approach is given.

  Existing algorithms:
  generic position, Habit-Sturm like sequence computation.

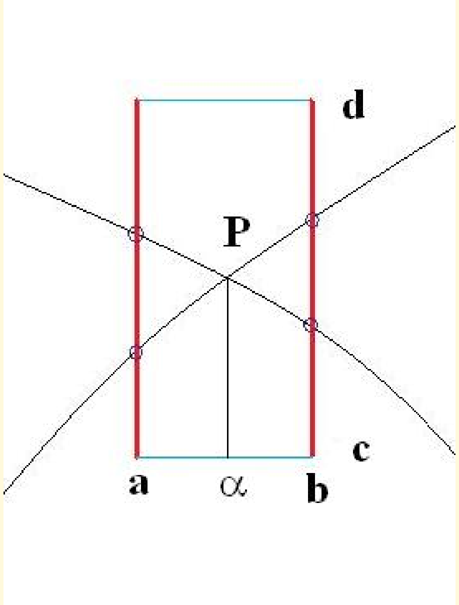


How near is enough? (Hong, 96)

Determining the Adjacency

Segregating Box: $B = [a, b] \times [c, d]$:

- B is an isolating box for P.
- Line segments $\{a \leq x \leq b, y = c\}$ and $\{a \leq x \leq b, y = d\}$ have intersections with $f(x, y) = 0$.

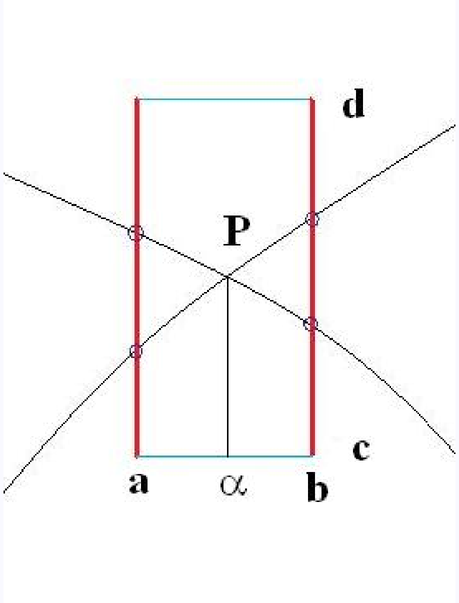


Lemma. $B = [a, b] \times [c, d]$ is a segregating box. Then
$L\#(P) = \#$ of roots $f(a, y) = 0$ inside (c, d).
$R\#(P) = \#$ of roots $f(b, y) = 0$ inside (c, d).

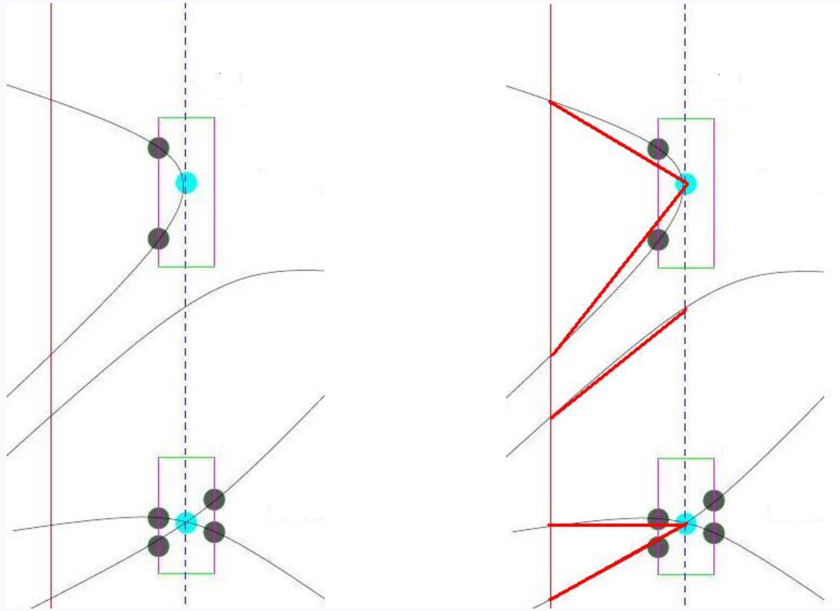
Segregating Box

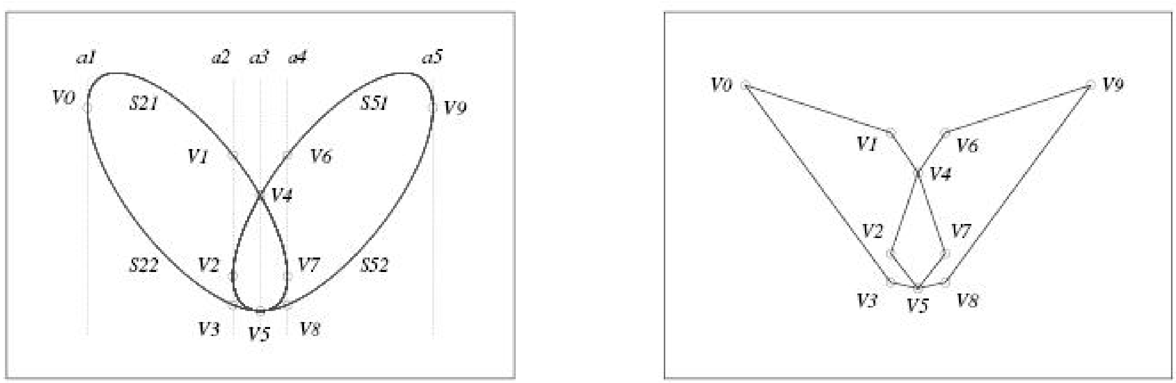
Segregating Box: $B = [a, b] \times [c, d]$:

- B is an isolating box for P.
- Line segments $\{a \le x \le b, y = c\}$ and $\{a \le x \le b, y = d\}$ have intersections with $f(x, y) = 0$.
- **Notation**: □$f(B)$
  - an interval containing $\{f(x, y) \,|\, a \le x \le b, c \le y \le d\}$.
  - □$f(B) \to 0$ when $|B| \to 0$.
- The Algorithm
  - Fix c and d.
  - While □$f([a, b], c)$ contains zero, refine the isolation interval (a, b) of the root α of $f(x) = 0$.
  - Since $f(\alpha, c) \neq 0$, this procedure will end.



Determining topology without changing to generic positions





A curve and its topology graph

Application: Rational Quadratic Approximation of Algebraic Curves

Implicitization: rational curves $\Longrightarrow$ implicit curves.

Methods: Groebner base, CS, resultant and moving curves and surfaces

Problems: large coefficients, self-intersecion, etc

Solution: approximate implicitization

Parametrization: implicit curves $\Longrightarrow$ rational curves.

Possible only for genus $g = 0$.

Solution: approximate parametrization

Rational Conics: low degree and well known.

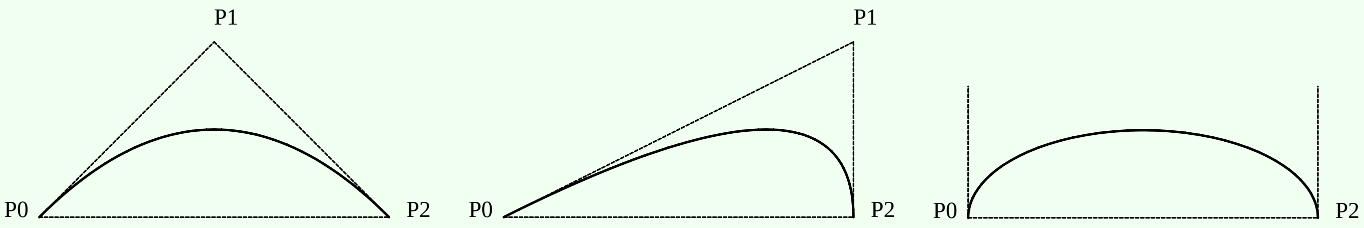
Our Approach. Rational (Quadratic) Approximation with correct topology.

I. Curve Division

To divide the curve into **triangle convex segment**

Triangle convex: (P_0P_2, S) is convex.

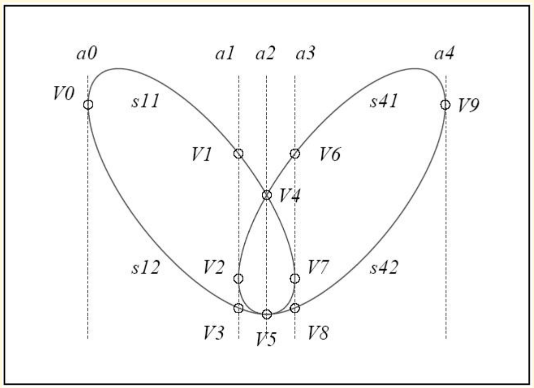
Shoulder point S_P: maximal distance to P_0P_2.

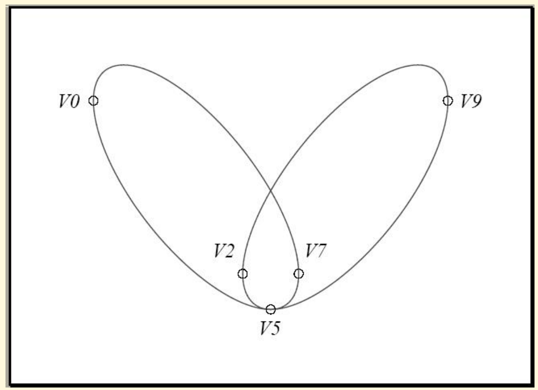


1. Topology Determination

2. Segments Combination

Delete Simple Points and Ordinary Singularities





3. Further Division at Flexes

Flex: non-singular, $H(f) = 0$.

4. Tangent Directions Computation

Flexes, Vertical Points, Singularities: (Zeng, 2003)

II. Segments Approximation with RQBC

Approximation Triangle Convex Segments with RQBC

1. Shoulder point computation (Newton method)

2. Shoulder point approximation

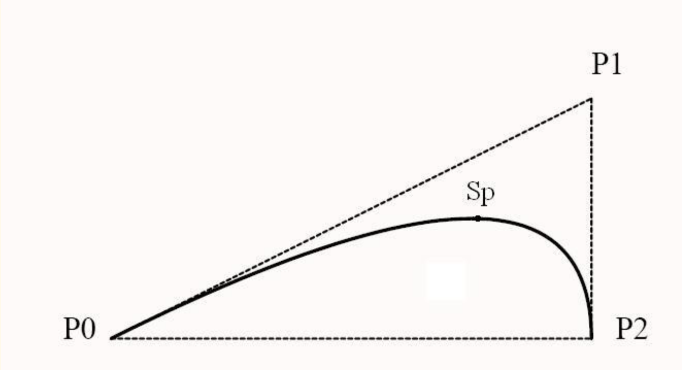
Triangle convex curve segment $S[P_0, T_0, P_2, T_2]$

2.1 $P_1 = (P_0, T_0) \bigcap (P_1, T_1)$

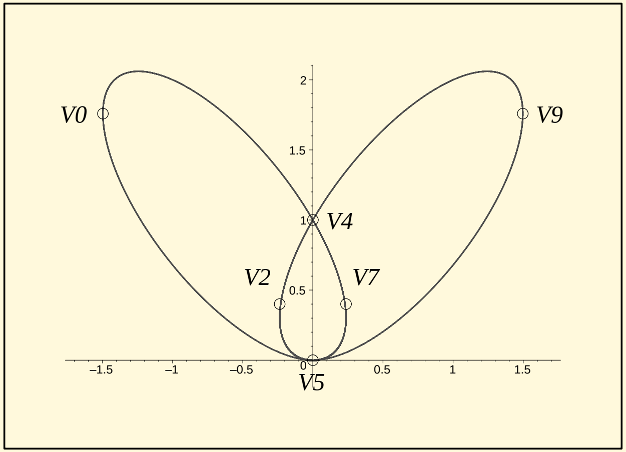
2.2 $B(t) = \frac{P_0\phi_0(t)+\omega P_1\phi_1(t)+P_2\phi_2(t)}{\phi_0(t)+\omega\phi_1(t)+\phi_2(t)}, 0 \le t \le 1$

2.3 ω : $\min_\omega d(S_p, B(1/2))$ (close form solution)

2.4 if $e(\mathcal{C}, B(t)) > \delta$, divide at S_P and repeat.



III. Curve Tracing and B-spline Generation

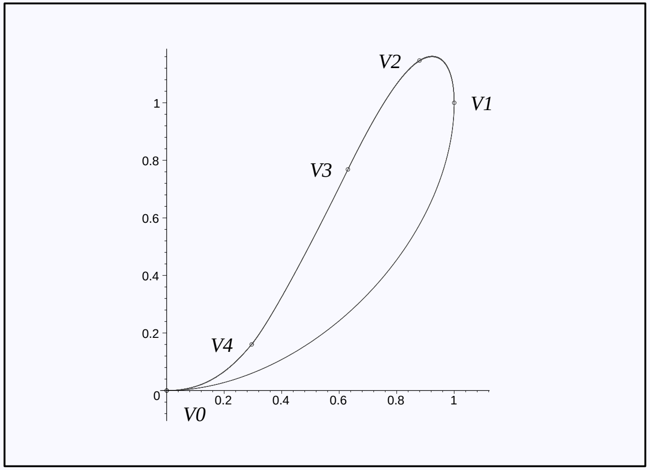


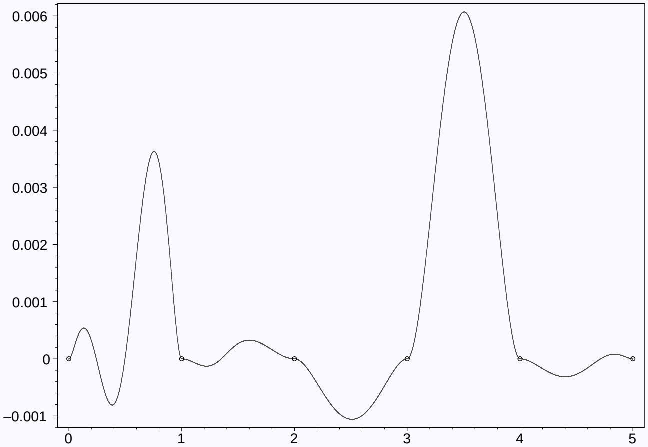
Tracing order: $V_5V_0V_4V_7V_5V_2V_4V_9V_5$

B-spline generation: knot selection, global C^1

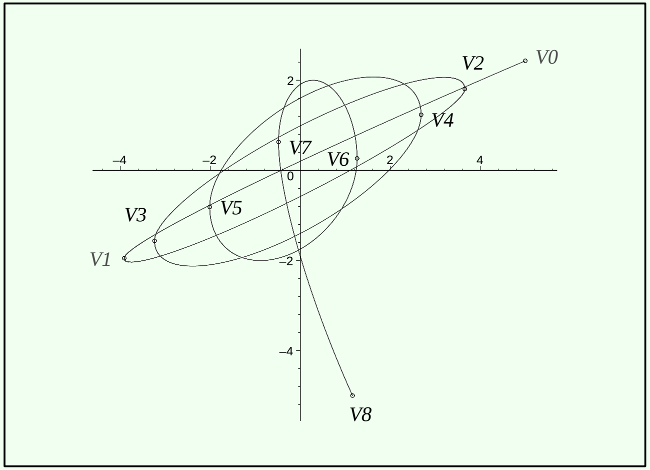
Experimental Results

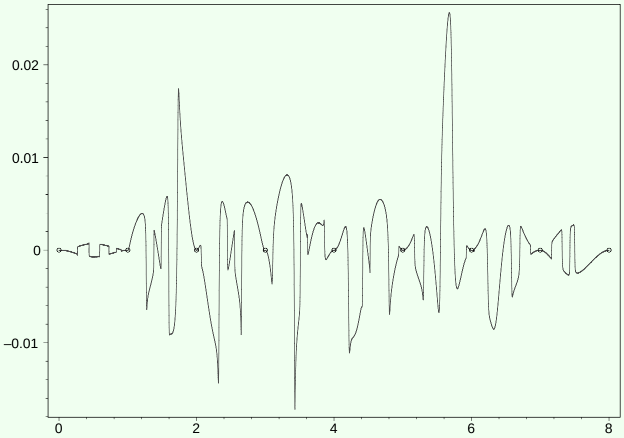
$$x^4 + x^2y^2 - 2x^2y - xy^2 + y^2 = 0$$





$$y^8 + y^7 - (8 + 7x)y^6 - (7 - 21x^2)y^5 - (-20 - 35x + 35x^3)y^4 -$$
$$(-14 + 70x^2 - 35x^4)y^3 - (16 + 42x - 70x^3 + 21x^5)y^2 -$$
$$(7 - 42x^2 + 35x^4 - 7x^6)y + 7x - 14x^3 + 7x^5 - x^7 = 0,$$





Experimental Results

		Our results			BX		
Ex.	deg	d-app	error	p-num	d-app	error	p-num
C_0	4	(2,2)	0.003	8	(2,1)	0.1	34
C_2	4	(2,2)	0.005	5	(3,3)	0.1	12
C_3	4	(2,2)	0.005	9	(2,1)	0.09	27
C_4	6	(2,2)	0.003	12	(2,1)	0.1	28

Comparison of our results with results in Bajaj and Xu (97)

Results.

1. High accuracy with a few segments
2. Maintenance of sharp points and with correct topology

3. Algebraic Surfaces

Represent the topology of $\mathcal{S} : f(x, y, z) = 0$ by a polyhedron: ($\mathcal{V}, \mathcal{E}, \mathcal{F}$)

- **Vertices($\mathcal{V}$):** Singular points, boundary points and some auxiliary points (regular points).
- **Edges($\mathcal{E}$):** Singular curve segments, boundary curve segments, and some auxiliary curve segments whose endpoints are inside $\mathcal{V}$.
- **Surfaces($\mathcal{F}$):** Surface Patch of $\mathcal{S}$ without singularities except for their boundaries, whose boundaries are inside $\mathcal{E}$.

Adjacency Information:

- **Vertices-Edges:** How many curve segments connect to a vertex?
- **Edges-Surfaces:** How many surface patches connect to a curve segments?

Basic Steps

- PROJECTION

  $f(x, y, z) \longmapsto g(x, y) = \text{Resl}(f, f_z, z)$

- TOPOLOGY DETERMINATION of plane curve $g(x, y) = 0$ and TRIANGULATION of plane region.

- LIFTING

  plane points $\longmapsto$ spatial points;

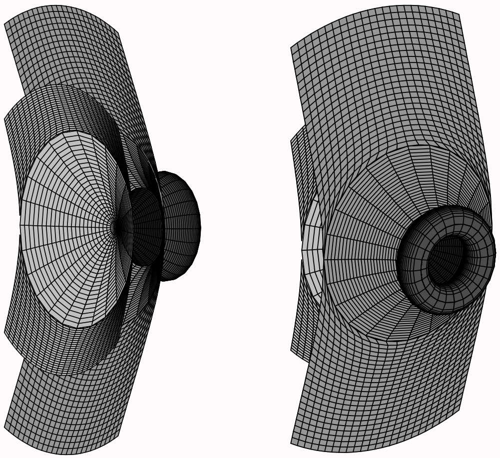
  plane curve segments $\longmapsto$ spatial curve segments;

  plane regions (cells) $\longmapsto$ spatial surface patches

- ADJACENCY INFORMATION
  - **Vertices-Edges:** Add (P, C) to the topology polyhedron.
  - **Edges-Surfaces:** Add (C, S) to the topology polyhedron.

Example. Consider the surface:

$$\mathcal{S} : f(x, y, z) = (y^2 + z^2 - x^2 + 1/2 \cdot x^3 - 4)^2 - 16 \cdot x^2 + 8 \cdot x^3 = 0.$$



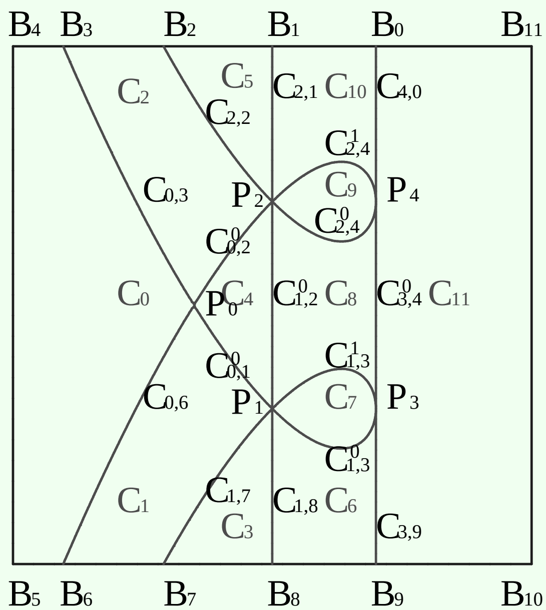
Project $\mathcal{S}$ to the XY-plane to obtain the projection curve $\mathcal{C} : g(x, y) = 0$.

Determine the topology of $\mathcal{C}$.

Points: $\{P_0, P_1 \cdots\}$

Curve Segments: $\{C_{0,1}, C_{1,3}, \cdots\}$

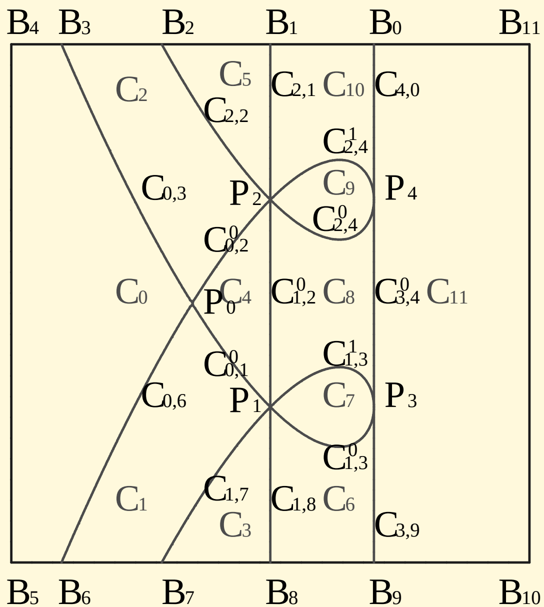
Cells: $\{E_0, E_1, \cdots\}$

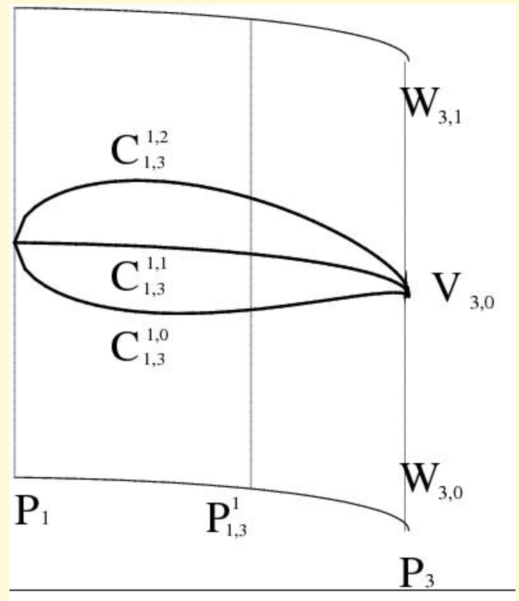


Lifting the plane critical points to spatial critical points

Determine the spatial curve segment

Example: Spatial curve lifted from P_3P_1.



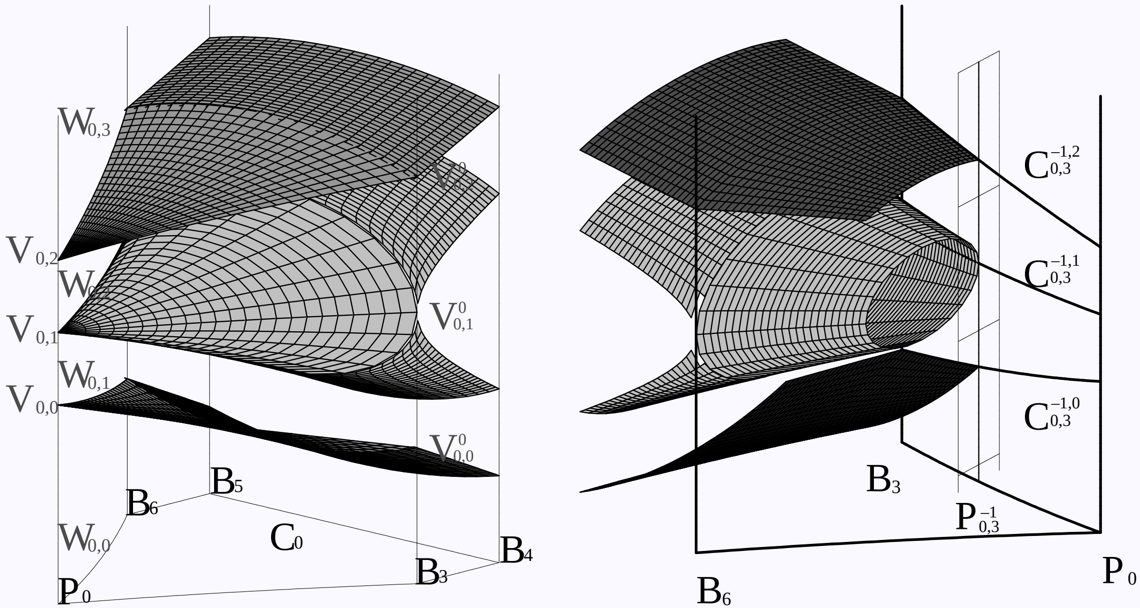


# of curves originating from $V_{3,0}$ = # of intersections of line $(x, y) = P^1_{1,3}$ and S.

If $P^1_{1,3}$ is close enough to P_3?

Using segregating box.

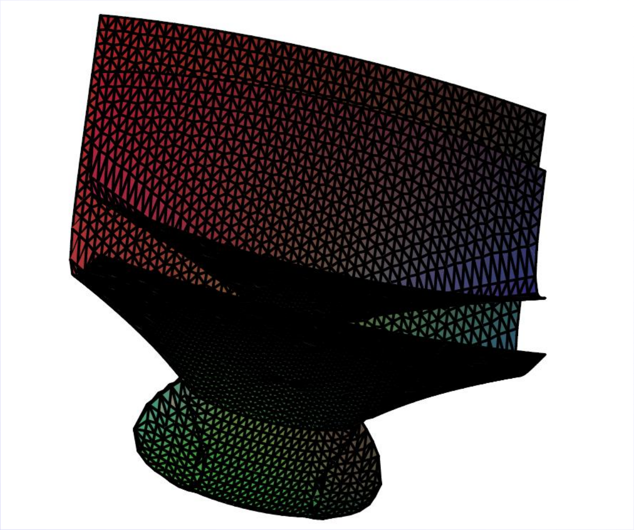
Determine the surface patch of $\mathcal{S}$.

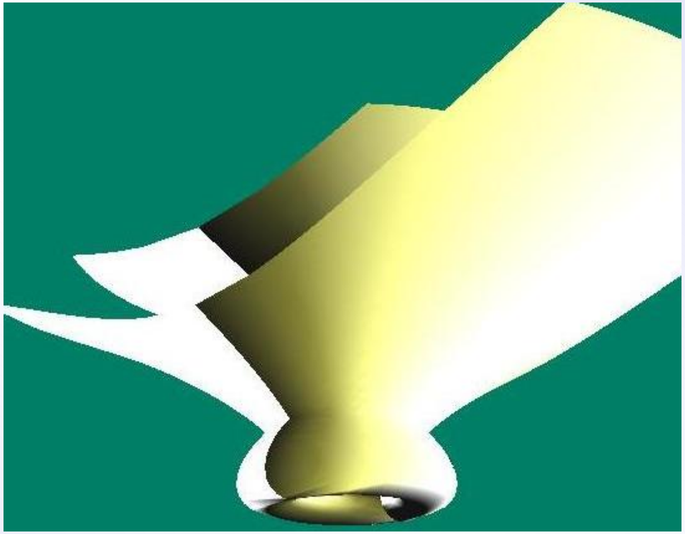


# of surfaces originating from P_0B_3 = # of intersections of line $(x, y) = P^{-1}_{1,3}$ and S.

Mesh the surface with correct topology

Subdivide until the precision is satisfied

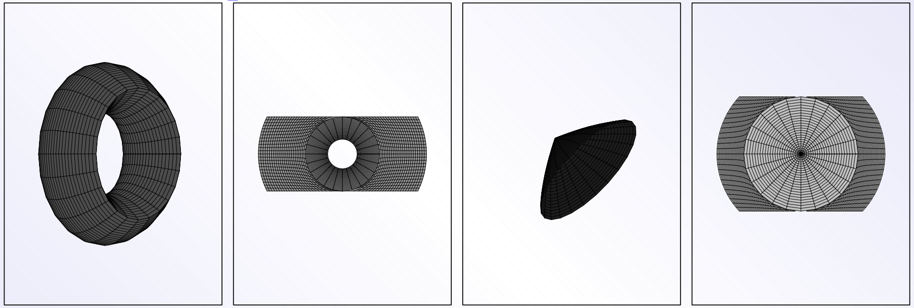




Intrinsic topology of surfaces

Combine any two surface patches which have a common non-singular intersection.

Example. The four finial surface patches of $\mathcal{S}$.



Euler characteristic: $\chi_1 = 0$, $\chi_2 = 0$, $\chi_3 = 1$, $\chi_4 = 1$.

Number of the close boundaries: $\beta_1 = 0$, $\beta_1 = 2$, $\beta_1 = 1$, $\beta_1 = 1$.

Connection Information: $V_{0,1}$: the isolated singular point of $\mathcal{S}$. L_1: the singular curve of $\mathcal{S}$.
B_1, B_2: the closed boundary curve segments of $\mathcal{S}$.

4. Root-isolating for Triangular Systems

Problem:

Find an isolating box $I = [a_1, b_1] \times \cdots \times [a_n, b_n]$ under a given precision ϵ for each real root ξ of a triangular system F_n:

$$f_1(x_1) = 0, \ldots, f_n(x_1, \ldots, x_n) = 0$$

- $\max_{j=0}^{n}\{b_j - a_j\} \leq \epsilon$.
- $\xi \in I$ and ξ is the unique root of F_n in I.
- $I \cap J = \emptyset$, I, J both are isolating boxes of real roots of F_n.

Motivation:

In CAD computation and topology determination of algebraic surfaces, we need to find zeros of triangular systems with multiple roots (Lazard, Mega 2007; Mourrain, Mega 2007).

Related Work

- For univariate polynomials with coefficients in $\mathbb{R}$
  - C.B. Soh and C.S. Berger, Strict aperiodic-property of polynomials with perturbed coefficients, *IEEE Transactions on Automatic Control*, 1989.
  - A. Eigenwillig et al, A descartes algorithm for polynomials with bit stream coefficients, CASC 2005, LNCS 3718, 2005.
- For triangular systems
  - G.E. Collins et al, interval arithmetic methods and Descartes' method, *Journal of Symbolic Computation*, 2002.
  - B. Xia and L. Yang, resultant computation, *Journal of Symbolic Computation*, 2002.
  - Z. Lu et al, sleeve polynomials, Proceedings of International Workshop on Symbolic-Numeric Computation, 2005.

These methods require the system to be square-free, regular. And they can not deal with multiple roots directly.

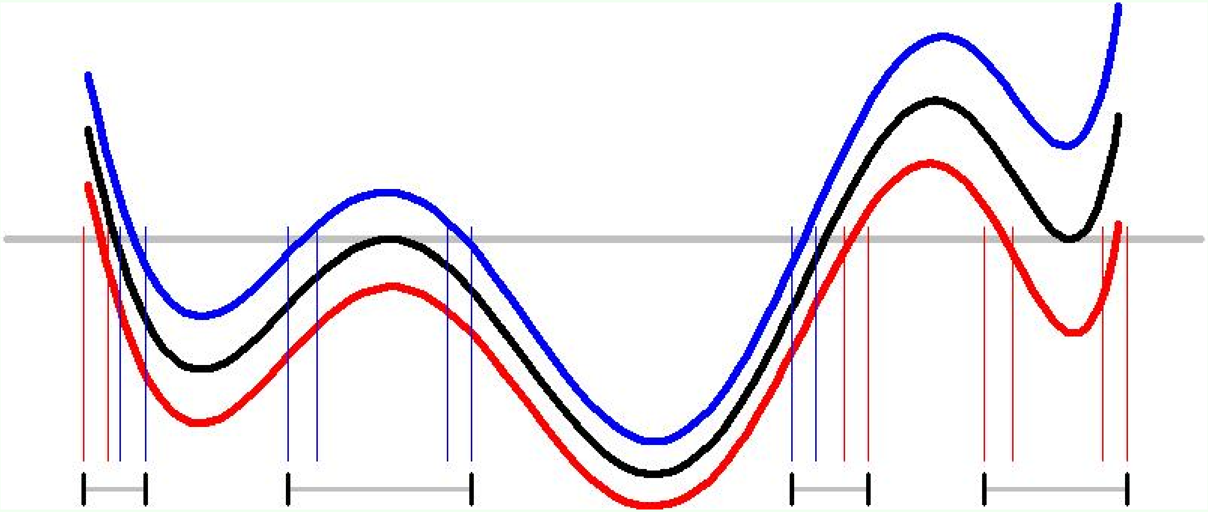
Our method can deal with any zero-dimensional triangular system, including the system with multiple roots, directly.

The Idea

- Solving triangular system $\Rightarrow$ Solving univariate polynomial in $\mathbb{R}[x]$
  Given triangular system $F_n := \{f_1, \ldots, f_n\}$
  For each root $(\xi_1, \ldots, \xi_i)$ of F_i, solve x_{i+1} with $f_{i+1}(\xi_1, \ldots, \xi_i, x_{i+1}) = 0 (i = 1, \ldots, n-1)$
- Solving univariate polynomial in $\mathbb{R}[x]$ $\Rightarrow$ Solving univariate polynomial in $\mathbb{Q}[x]$
  For $f(x) \in \mathbb{R}[x]$, construct sleeve (Lu, et al) $f^u(x), f^d(x) \in \mathbb{Q}[x]$ such that

$$f^d(t) < f(t) < f^u(t).$$

Real roots isolation of $f \Rightarrow$ Real roots isolation of f^d, f^u.



Sleeve-Evaluation Inequality

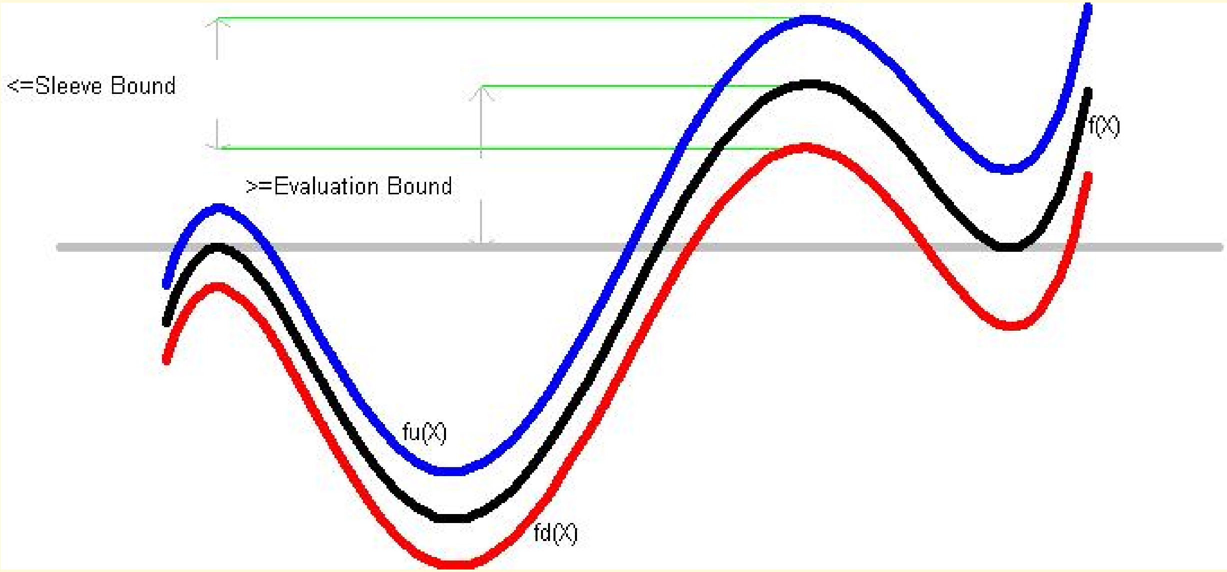
Sleeve Bound: $SB_I(f^u, f^d) \triangleq \sup\{f^u(x) - f^d(x) : x \in I = [a, b]\}$.

Evaluation Bound: $EB_I(f) := \inf\{|f(x)| : x \in zero(f') \cup \{a, b\} \setminus zero(f)\}, x \in I\}$,where $zero(g) := \{x : g(x) = 0\}$.

Lemma. When the sleeve-evaluation inequality holds:

$$SB_I(f^u, f^d) < EB_I(f),$$

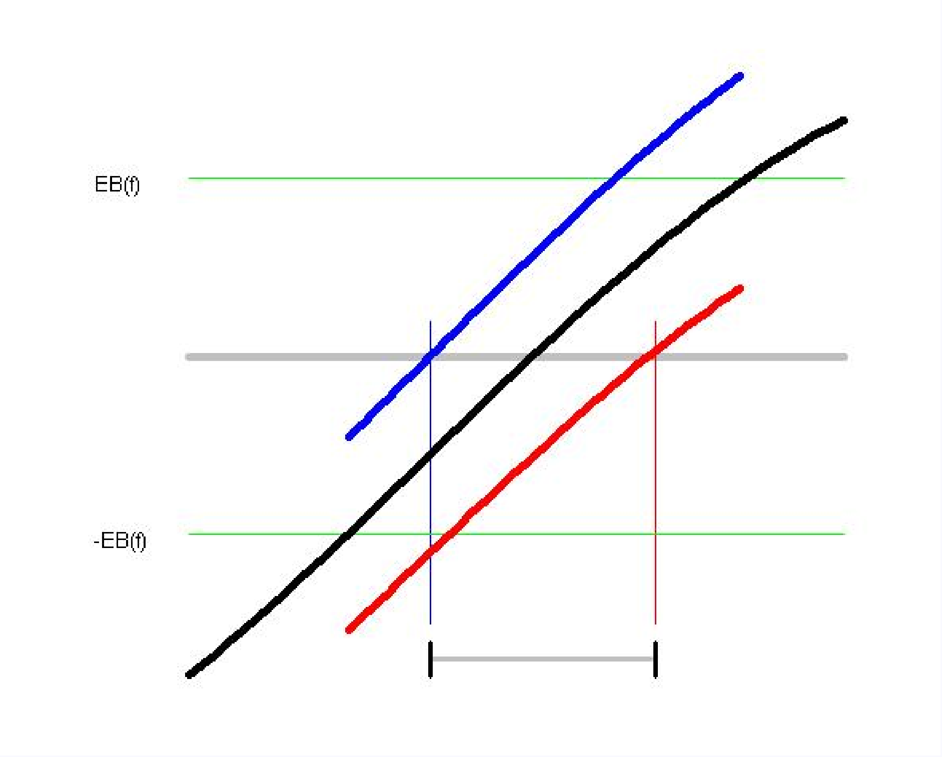
all the real roots of f are isolated by the real roots of f^u or f^d.



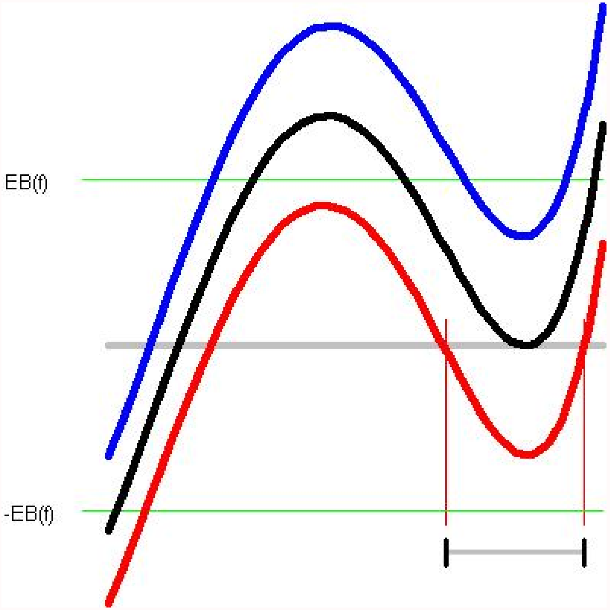
Odd Root Isolation

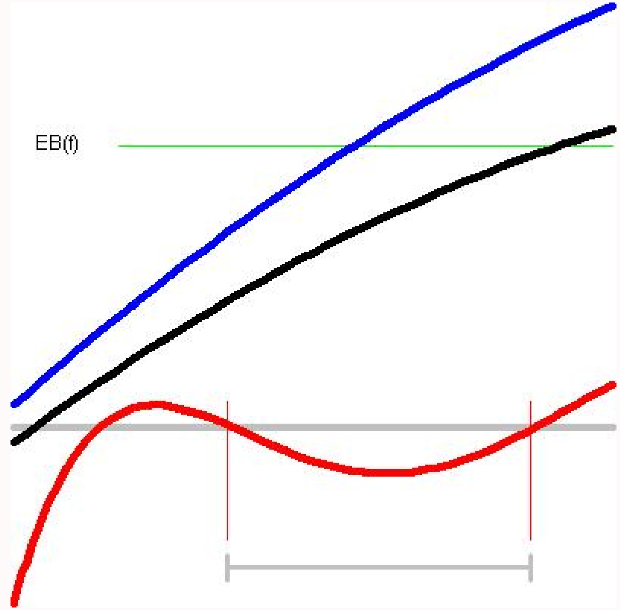
Lemma. Interval $J_j = (t_{2j}, t_{2j+1})$ is an isolating interval of f when

$t_{2j} \in zero(f^u)$ and $t_{2j+1} \in zero(f^d)$.



Problems for Even Roots Isolation



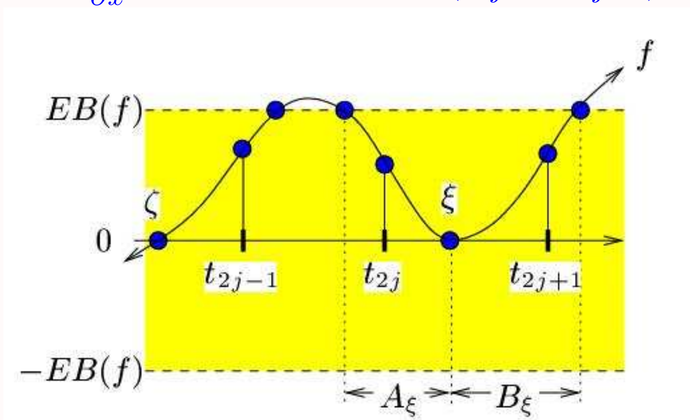


For even root, we need **Monotonicity Property:**

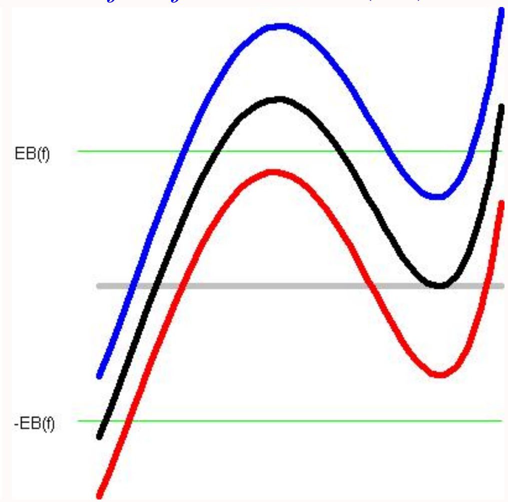
$$\frac{\partial f^u}{\partial x} \geq \frac{\partial f}{\partial x} \geq \frac{\partial f^d}{\partial x} \textbf{ holds in } I$$

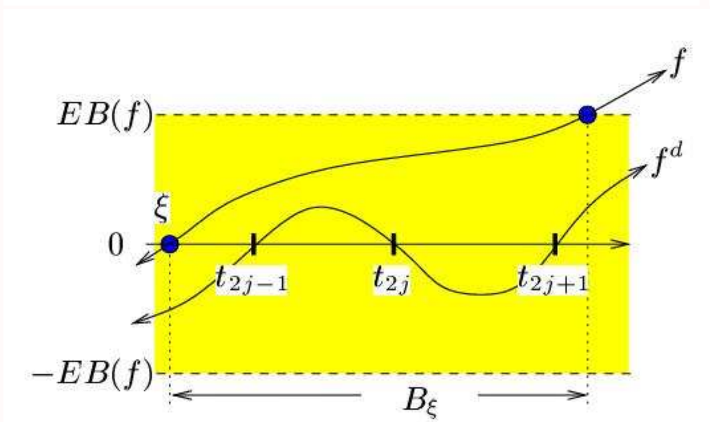
Even Root Isolation

Lemma. Interval (t_{2j}, t_{2j+1}) is an isolating interval of f iff
$\frac{\partial f^u}{\partial x}$ has a real zero in (t_{2j-1}, t_{2j+1}) when $t_{2j}, t_{2j+1} \in zero(f^d)$.

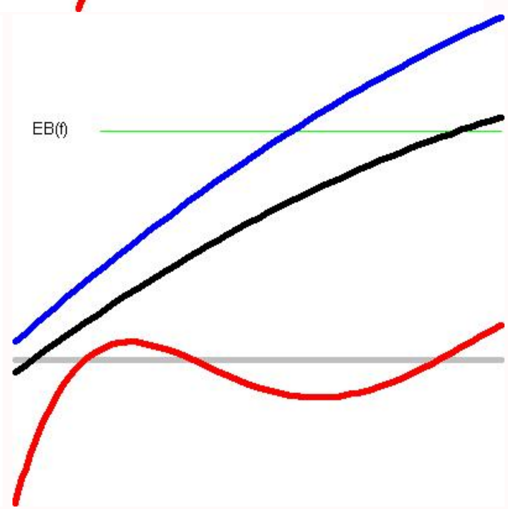


(a)





(b)



Triangular System

1. How to construct sleeve?
2. How to compute sleeve bound and evaluation bound?
3. How to ensure monotonicity property?

Constructing Sleeve & Monotonicity Property

Let $f \in \mathbb{Z}[x_1, \ldots, x_n]$, then $f = f^+ - f^-$, where $f^+, f^- \in \mathbb{Z}_+[x_1, \ldots, x_n]$.
$J_\xi = [a_1, b_1] \times \cdots \times [a_{n-1}, b_{n-1}]$ is an isolating box for the zero $\xi = (\xi_1, \ldots, \xi_{n-1})$ of $F_{n-1} = 0$, when $a_i \geq 0, b_i \geq 0$, we have the sleeve (Lu, et al)

$$f^u(x) := f_n^u(J_\xi, x) = f_n^+(b_1, \ldots, b_{n-1}, x) - f_n^-(a_1, \ldots, a_{n-1}, x),$$

$$f^d(x) := f_n^d(J_\xi, x) = f_n^+(a_1, \ldots, a_{n-1}, x) - f_n^-(b_1, \ldots, b_{n-1}, x)$$

for

$$f(x) := f_n(\xi, x) = f_n(\xi_1, \ldots, \xi_{n-1}, x).$$

- Monotonicity property $\frac{\partial f^u}{\partial x}(x) \geq \frac{\partial f}{\partial x}(x) \geq \frac{\partial f^d}{\partial x}(x)$ is true directly.
- $f^u(x) - f^d(x)$ is monotonously increasing, so, when $I = [a, b]$,

$$SB_I(f^u, f^d) = f^u(b) - f^d(b).$$

Computing Evaluation Bound

Let $\bar{F}_n := \{f_1, \ldots, f_{n-1}, \frac{\partial f_n(x_1,\ldots,x_{n-1},x_n)}{\partial x_n}, Y - f_n(x_1, \ldots, x_{n-1}, x_n)\}$.

- Yap's bound:

  The evaluation bound $EB(F_n)$ of $F_n = \{f_1, \ldots, f_n\}$:

  $$EB(F_n) \geq (2^{3/2} NK)^{-D}\, 2^{-(n+1)d_1 \cdots d_n},$$

  where $K := \max\{\sqrt{n+2}, \|f_1\|_2, \ldots, \|f_{n-1}\|_2, \|Y - f_n\|_2, \|\frac{\partial f_n}{\partial x_n}\|_2\}$,

  $$N := \binom{1 + \sum_{i=1}^{n+1} d_i}{n+1}, \quad D := (1 + \sum_{i=1}^{n+1} \frac{1}{d_i}) \prod_{i=1}^{n+1} d_i,$$

  $d_i = \deg(f_i)$ for $i = 1, \ldots, n-1$, $d_n = \deg(\frac{\partial f_n}{\partial x_n})$, $d_{n+1} = \deg(Y - f_n)$.

- Resultant Computation:

  $$e_n = res_X(Y - f_n, \frac{\partial f_n}{\partial X}), \;\; e_i = res_{x_i}(e_{i+1}, f_i), i = n-1, \ldots, 1,$$

  where $res_x(p, q)$ is the resultant of p and q relative to x. Thus $e_1 \in \mathbb{Z}[Y]$, we have

  $$EB(F_n) := \min\{|z| : e_1(z) = 0, z \neq 0\}.$$

Experimental Results

Triangular system $(f_1, f_2, \ldots, f_n)$, precision is 2^{-10}.

Timings collected on a PC with a 3.2G CPU and 512M memory.

TYPE	NT	TIME	NS	NE
(3, 3)	100	0.04862	2.04	(4, 10)
(9, 7)	100	0.52717	3.99	(10, 10)
(21, 21)	20	108.9115	5.45	(10, 10)
(3, 3, 3)	100	0.15783	3.48	(4, 10, 10)
(9, 7, 5)	100	16.20573	8.36	(10, 10, 10)
(3, 3, 3, 3)	100	1.69115	5.64	(4, 10, 10, 10)
(3, 3, 3, 3, 3)	10	159.1199	8.0	(4, 10, 10, 10, 10)

Timings for solving randomly generated sparse triangular systems

TYPE:= $(deg_{x_1}(f_1), deg_{x_2}(f_2), \ldots, deg_{x_n}(f_n))$.

NT is the number of tested triangular systems.

TIME is the average running time for each triangular system in seconds.

NS is the average number of real solutions for each triangular system.

NE is the average number of terms in each polynomial.

TYPE	NT	TIME	NS	NE
(3, 3)	100	0.05355	1.91	(3.99, 8.02)
(9, 8)	100	1.87486	4.26	(9.94, 43.98)
(11, 11)	80	8.78255	4.5	(11.975, 72.5)
(16, 14)	100	50.22294	6.0	(16.9, 127.13)
(21, 15)	100	164.23443	6.22	(21.91, 176.8)
(3, 3, 3)	100	0.38702	2.91	(3.99, 7.77, 13.01)
(5, 4, 4)	100	2.97011	4.88	(5.99, 14.72, 24.24)
(5, 5, 5)	80	33.225275	5.6125	(5.9625, 17.775, 42.1375)
(8, 7, 6)	10	592.1848	7.6	(8.9, 36.0, 79.8)
(3, 3, 3, 3)	50	119.94042	6.96	(4.0, 8.12, 12.82, 20.92)
(5, 5, 5, 3)	10	551.4401	3.4	(6.0, 32.1, 42.3, 21.5)

Timings for solving randomly generated dense triangular systems

TYPE	NT	TIME	NS	NM	NE
(5, 5)	100	0.71251	3.71	1.57	(5.97, 34.47)
(9, 8)	100	0.60408	3.1	3.1	(9.94, 18.92)
(13, 11)	100	32.44376	6.55	3.92	(13.94, 107.68)
(23, 21)	20	466.0289	6.15	3.75	(24.0, 183.4)
(3, 3, 3)	100	3.21342	5.59	3.24	(3.99, 13.08, 31.71)
(9, 7, 5)	20	425.95055	12.95	8.15	(9.95, 60.85, 100.35)
(3, 3, 3, 3)	20	130.617	11.15	6.1	(4.0, 12.2, 33.7, 62.95)

Timings for solving dense triangular systems with multiple roots

The column NM gives the average number of multiple roots for the tested triangular systems.

Without Using Evaluation Bound

Lemma. Let (f^u, f^d) be a sleeve of f, $[a, b]$ an odd candidate interval of f, then $[a, b]$ is an isolating interval of f if f^u, f^d both are monotonous in $[a, b]$.
Furthermore, if f is squarefree, then for each root ξ of $f = 0$, there exists an interval I containg ξ such that f^u, f^d are monotonous in I.

Monotonous of a function $f(x)$ on an interval I can be efficiently checked with $\square f(I) > 0$. This avoids the computation of Sturm sequence.

TYPE	TIME	NS	NT	NE
(9, 8)	0.03282	4.39	100	(9.9, 44.67)
(21, 15)	0.09391	5.75	100	(21.85, 135.37)
(25, 21)	0.19454	6.66	100	(25.88, 251.81)
(51, 41)	0.93559	9.08	100	(51.78, 898.49)
(119, 70)	4.33518	11.77	100	(119.38, 2543.39)
(219, 180)	54.33796	15.33	100	(218.7, 16387.83)
(8, 7, 6)	0.12077	8.08	100	(8.96, 35.86, 83.64)
(19, 17, 14)	1.22715	14.58	100	(19.94, 170.11, 676.38)
(39, 37, 31)	19.44103	24.63	100	(39.82, 737.1, 5954.7)
(139, 77, 41)	70.086375	36.25	40	(139.25, 3066.475, 13177.05)
(9, 7, 5, 3)	0.14828	8.05	100	(9.95, 35.85, 55.73, 34.8)
(19, 17, 15, 13)	9.49561	29.81	100	(19.96, 170.14, 811.96, 2368.19)
(59, 37, 25, 23)	71.62045	28.1	20	(59.45, 737.0, 3259.15, 17458.95)
(11, 9, 8, 7, 5, 3, 3)	3.41567	41.31	100	(11.9, 54.7, 163.31, 328.42, 250.94, 83.57, 119.38)

Conclusion

1. We give an algorithm to determine a topologically correct meshing for algebraic curves and surfaces without changing them to generic positions.
2. We can approximate algebraic curves with RQBS curve which has the same topology as the original curve.
3. The main computation tool: a complete algorithm for root isolation of triangular systems using interval arithmetics.

References

1. X.S. Gao and M. Li, Rational Quadratic Approximation to Real Algebraic Curves, *Computer Aided Geometric Design*, 2004.
2. M. Li, X.S. Gao, and S.C. Chou, Quadratic Approximation to Plane Parametric Curves and Applications in Approximate Implicitization, *Visual Computers*, 2006.
3. J.S. Cheng, X.S. Gao, and C.K. Yap, Complete Numerical Isolation of Real Zeros in General Triangular Systems, *ISSAC 2007*, 2007.
4. J.S. Cheng, X.S. Gao, and M. Li, Determine the Topology of Real Algebraic Surfaces, *Mathematics of Surfaces*, Springer, 2005.
5. J.S. Cheng and X.S. Gao, Topology Determination and Triangular Meshing for Algebraic Curves and Surfaces, Preprints, 2008.

Thanks!