Geometric Computing and Root Isolation

Kurt Mehlhorn

Max Planck Institute for Informatics and Saarland University

September 20, 2010

Outline

CGAL

Computational Geometry Algorithms Library

- a comprehensive library for geometric computing
- joint effort of INRIA Sophia Antipolis, Tel Aviv, Berlin, ETH, Groningen, MPI-INF, and many others

- algs in CGAL are exact, complete and efficient

this requires new theory

An Arrangement of Algebraic Curves

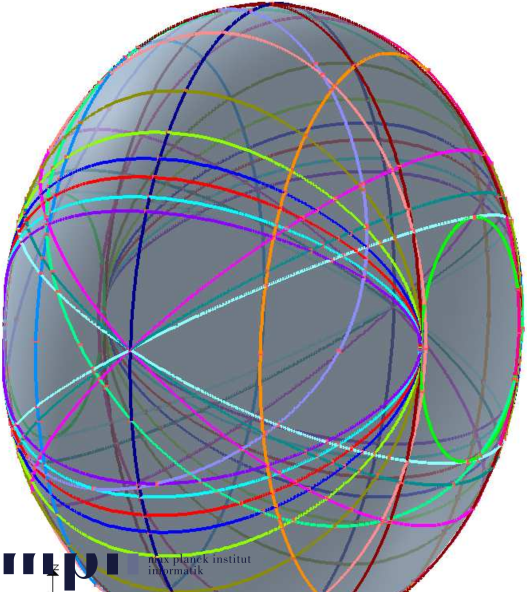
input: a set of algebraic curves

output: their arrangement (= a planar embedded graph)

alg is exact and handles any input

Eigenwillig, Kerber, Wolpert

The Intersection of Quadric Surfaces



input: a set of quadrics $S_0, S_1, \ldots$

output: the arrangement of their intersection curves with S_0

alg is exact and handles any input

Berberich, Fogel, Halperin, M, Wein

An all-important primitive

Intersecting two algebraic curves

see also talk by F. Rouillier

Intersecting Two Lines

intersect $5x + 7y - 1 = 0$ and $3x - 6y + 4 = 0$

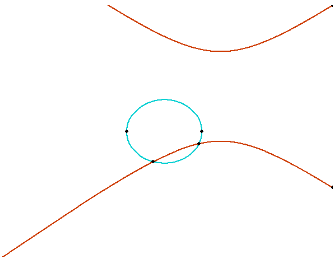
eliminate a variable, say y, and obtain $51x + 22 = 0$

solve for x and obtain $x = -\frac{22}{51}$

substitute into one of the equations and obtain $-\frac{110}{51} + 7y - 1 = 0$

solve for y and obtain $y = \frac{23}{51}$

Intersecting Two Algebraic Curves



intersect $5x^2 + 7y^2 - 1 = 0$ and $3x^2 - 4x - 6y^2 + 5y + 2 = 0$

eliminate a variable, say y, and obtain $1601x^4 - 2656x^3 + \dots$

solve for x and obtain $x_1 = 0.399\dots$, $x_2 = -0.1475\dots$, $x_3 = \dots$, $x_4 = \dots$

substitute x_1 into one of the equations and obtain $7y^2 + 5(0.399\dots)^2 - 1 = 0$

solve for y and obtain $y_{ij} =$

select the right y_{ij}

eliminate y from $5x^2+7y^2-1=0$ and $3x^2-4x-6y^2+5y+2=0$

$$p(x)=\begin{vmatrix} 7 & 0 & 5x^2-1 & 0 \\ 0 & 7 & 0 & 5x^2-1 \\ 6 & 5 & 3x^2-4x+2 & 0 \\ 0 & 6 & 5 & 3x^2-4x+2 \end{vmatrix} = 1601x^4-2656x^3+\ldots$$

- Sylvester resultant
- roots of $p(x)$ are the x-coordinates of the intersections
- Emeliyanenko ('10): evaluate $p(x)$ at five values in parallel (GPU) and interpolate

Root Isolation

Input: a polynomial p given through its coefficient sequence

Output: isolating intervals for the real roots

Isolating Interval

an interval $[a, b]$ is isolating if it contains exactly one root of p and is disjoint from other isolating intervals

isolating intervals are easily refined (Newton iteration or Abott's method)

Coefficients

- integral, e.g. 27, or **bitstreams**, e.g., $\pi = 3.14\ldots$
- bitstreams are potentially infinite; we can ask for additional bits

Root Separation: A Measure of Difficulty

Root Separation

- $x_1, \ldots, x_n$, the complex roots of p
- $\sigma(p) = \min\{\,|x_i - x_j|;\, i \neq j\,\}$, the root separation of p
- intuition: the smaller $\sigma(p)$, the harder it is to isolate the roots
- remark: $\sigma(p)$ is zero, if p has multiple roots

Example

- $p = x^2 - 2$
- roots $x_1 = -\sqrt{2}$, $x_2 = +\sqrt{2}$
- $\sigma(p) = 2\sqrt{2}$
- isolating intervals, e.g., $(-2,-1)$ and $(1,2)$

Root Separation: A Measure of Difficulty

Root Separation

- $x_1, \ldots, x_n$, the complex roots of p
- $\sigma(p) = \min\{\,|x_i - x_j|;\, i \neq j\,\}$, the root separation of p
- intuition: the smaller $\sigma(p)$, the harder it is to isolate the roots
- remark: $\sigma(p)$ is zero, if p has multiple roots

Example

- $p = x^2 - 2$
- roots $x_1 = -\sqrt{2}$, $x_2 = +\sqrt{2}$
- $\sigma(p) = 2\sqrt{2}$
- isolating intervals, e.g., $(-2, -1)$ and $(1, 2)$

Root Isolation is well-studied with a 200 year history

two kinds of papers

- algorithms without a convergence guarantee
- algorithms with a guarantee
  - Simple Bisection Methods: Descartes, Gauss, Vincent, Uspensky, Ostrowski, Collins/Loos, Krandick/Mehlhorn, Rouillier/Zimmermann, Mourrain/Roy/Rouillier, Emiris/Tsigaridis, Mehlhorn/Sagraloff, Eigenwillig/Sharma/Yap. . .
  - Advanced Methods: Henrici, Schönhage, Pan, Smale, . . .
- Pan's algorithm is the asymptotically fastest
- but, in his own words:

  > *The algorithm is quite involved, and would require non-trivial implementation work. No implementation was attempted yet.*

- open problem: is Pan's alg competitive in practice?

Root Isolation is well-studied with a 200 year history

two kinds of papers

- algorithms without a convergence guarantee
- algorithms with a guarantee
  - Simple Bisection Methods: Descartes, Gauss, Vincent, Uspensky, Ostrowski, Collins/Loos, Krandick/Mehlhorn, Rouillier/Zimmermann, Mourrain/Roy/Rouillier, Emiris/Tsigaridis, Mehlhorn/Sagraloff, Eigenwillig/Sharma/Yap...
  - Advanced Methods: Henrici, Schönhage, Pan, Smale, ...
- Pan's algorithm is the asymptotically fastest
- but, in his own words:

  The algorithm is quite involved, and would require non-trivial implementation work. No implementation was attempted yet.

- open problem: is Pan's alg competitive in practice?

Sign Variations $var(q)$ in a sequence $q = (q_0, \ldots, q_n)$ of reals

$var(q)$ is the number of pairs (i,j) of integers with $0 \le i < j \le n$ and $q_i q_j < 0$ and $q_{i+1} = \ldots = q_{j-1} = 0$ $\qquad var(3,0,-2,2,-1) = 3.$

Descartes' Rule of Signs:

- Let $q(x) = \sum_{i=0}^{n} q_i x^i$. Then

$$var(q) = \#\text{ of positive real roots} + 2k \quad \text{for some } k \in \mathbb{N}_0$$

- $var(q) = 0 \quad \Rightarrow \quad q$ has no positive real root
- $var(q) = 1 \quad \Rightarrow \quad q$ has exactly one positive real root
- extension to arbitrary intervals
  - zeros of p in $I = (c,d)$: consider $q_I(x) := (1+x)^n \cdot p\left(\frac{cx+d}{x+1}\right)$
  - roots of p in I correspond to positive roots of q_I
  - define $var(p,I) := var(q_I)$

Sign Variations $var(q)$ in a sequence $q = (q_0, \ldots, q_n)$ of reals

$var(q)$ is the number of pairs (i,j) of integers with $0 \le i < j \le n$ and $q_i q_j < 0$ and $q_{i+1} = \ldots = q_{j-1} = 0$ $\qquad var(3,0,-2,2,-1) = 3$.

Descartes' Rule of Signs:

- Let $q(x) = \sum_{i=0}^{n} q_i x^i$. Then

$$var(q) = \#\text{ of positive real roots} + 2k \quad \text{for some } k \in \mathbb{N}_0$$

- $var(q) = 0 \quad \Rightarrow \quad q$ has no positive real root
- $var(q) = 1 \quad \Rightarrow \quad q$ has exactly one positive real root
- extension to arbitrary intervals
  - zeros of p in $I = (c,d)$: consider $q_I(x) := (1+x)^n \cdot p\left(\frac{cx+d}{x+1}\right)$
  - roots of p in I correspond to positive roots of q_I
  - define $var(p,I) := var(q_I)$

A Recursive Algorithm

Rouillier/Zimmermann

Root Bound for $p(x) = \sum_{1 \leq i \leq n} p_i x^i$

real roots have absolute value bounded by $1 + \max_i p_i / p_n$

Task: isolate real roots of $p(x)$

initialize $I = (c, d)$ according to root bound

if $var(p, I) = 0$ return;

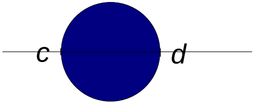
if $var(p, I) = 1$, return and report (c, d) as an isolating interval

otherwise. Let $m = (c + d)/2$.

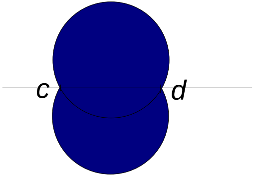
- If $p(m) = 0$, report $[m, m]$ as an isolating interval.
- recurse on both sub-intervals (c, m) and (m, d)

The Descartes Test: Partial Converses

Landau proved the following partial converses: Let $I = (c, d)$

if 

contains no root, then $var(p, I) = 0$.

if 

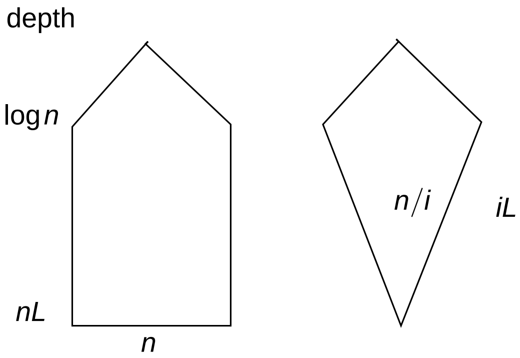
contains exactly one root, then $var(p, I) = 1$

if $w(I) \leq \sigma(p)$, then $var(p, I) \leq 1$

Analysis for L-Bit Integer Coefficients

- stopping criterium applies at intervals of length $\sigma(p)$.
- recursion depth = $\log(M/\sigma(p))$ where M =length of start interval
- $\log M = O(L)$ and $\log(1/\sigma(p)) = \widetilde{O}(nL)$
  thus recursion depth = $\widetilde{O}(nL)$
- numbers grow by n bits in every node of the recursion tree
- so numbers grow to $L + n\log(M/\sigma(p)) = \widetilde{O}(n^2L)$ bits
- $\widetilde{O}(n)$ arithmetic operations in every node
- width of tree is $O(n)$ since *var* is subadditive over intervals
- bit-complexity = $\widetilde{O}(n \cdot nL \cdot n \cdot n^2L) = \widetilde{O}(n^5L^2)$
- this assumes fast integer multiplication and Taylor shift

Improved Analysis (Krandick (95), Krandick/Mehlhorn (06), Eigenwillig/Sharma/Yap (06))



consequence: running time is $\widetilde{O}(n^4L^2)$

Continued Fraction Method (Vincent, Akritas)

Find Zeros of p in $[0,\infty]$

- if $p(0) = 0$, replace p by p/x and recurse
- find a (large) integer $b \leq$ any positive real root of p;
- recurse on $[b, b+1)$ and $[b+1, \infty)$

(recursion involves a Taylor shift)

Analysis (Sharma (08))

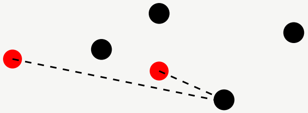
- recursion tree (depth, growth of coefficients, arithmetic operations) has similar properties (this assumes a good b), but
- time to compute a good b was $O(n^2)$
- time bound $\widetilde{O}(n^5 L^2)$

Hong's bound for $p = \sum_{0 \le i \le n} a_i x^i$

$$H(p) = \max_{i,\ a_i<0} \left(\min_{j>i,\ a_j>0} \left(\frac{|a_i|}{|a_j|} \right)^{1/(j-i)} \right)$$ is a good b

Geometry helps Algebra (Mehlhorn/Ray (09))

- let's take logarithms
  $\log H(p) = -\max_{i,\ a_i<0} \left(\min_{j>i,\ a_j>0} (\log|a_j| - \log|a_i|)/(j-i) \right)$
- define points $q_i = (i, \log|a_i|)$



red = "$a_i < 0$

black = "$a_j > 0$"

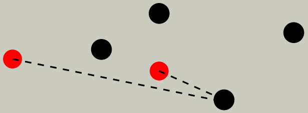
- computation of $H(p)$ reduces to dynamic convex hull problem: $O(n)$ instead of $O(n^2)$

Hong's bound for $p = \sum_{0 \le i \le n} a_i x^i$

$$H(p) = \max_{i,\ a_i<0} \left(\min_{j>i,\ a_j>0} \left(\frac{|a_i|}{|a_j|} \right)^{1/(j-i)} \right)$$ is a good b

Geometry helps Algebra (Mehlhorn/Ray (09))

- let's take logarithms
  $\log H(p) = -\max_{i,\ a_i<0} \left(\min_{j>i,\ a_j>0}(\log|a_j| - \log|a_i|)/(j-i)\right)$
- define points $q_i = (i, \log|a_i|)$



red = "$a_i < 0$

black = "$a_j > 0$"

- computation of $H(p)$ reduces to dynamic convex hull problem: $O(n)$ instead of $O(n^2)$

Bitstream Coefficients

Definition

- how about more complex coefficients, e.g., $\sqrt{2}$, π, $\ln 2$, $\sin(\pi/19)$
- in principle: use exact arithmetic in domain of coefficients
- better: approximate coefficients, i.e., coefficients are given by their binary representation (= potentially infinite bitstream): $\pi = 3.14\ldots$
- can ask for approximations of arbitrary precision
- we assume: $p(x) = \sum_{0 \le i \le n} p_i x^i$, a polynomial of degree n
- $p_n \ge 1$, $p_i \le 2^\tau$ for all i — τ bits before binary point
- $\sigma(p)$, the root separation of p

Theorem (Mehlhorn/Sagraloff (09))

Theorem: Isolating intervals can be computed in time polynomial in n and $\tau + \log 1/\sigma(p)$.

more precisely, $\tilde{O}(n^2(\tau + \log(1/\sigma(p))) \cdot n(\tau + \log(1/\sigma(p))))$ bit operations

Sagraloff (2010) improves upon this (see below)

Experimental Experience

$p(x)$, a polynomial with integer coefficients

running times on $p(x)$, $\pi \cdot p(x)$, and $\sqrt{2} \cdot p(x)$ are essentially the same

running time depends on "geometry of the polynomial", but not on the representation of the polynomial

Real Coefficients: Approach I

Interval Coefficients (Collins/Johnson/Krandick (02))

- replace coefficients by intervals
- then run alg on interval polynomials
- very successful in practice: Rouillier's solver RS (Maple, CGAL) even on integer polynomials with large coefficients
- two problems:
  - not every interval has a sign
  - quality of approximation, width of intervals
- Eigenwillig/Kettner/Krandick/M/Schmitt/Wolpert (2005) use randomization to make approach complete

Real Coefficients: Approach II

Isolate Roots of an Approximation p^* (M/Sagraloff (09)

- roots depend continuously on coefficients
  - therefore, isolate the roots of a suitable approximation p^*
  - return slightly enlarged intervals
- difficulties
  - how good must approximation be?
  - how can we make sure that enlarged intervals are disjoint?

Roots Depend Continuously on Coefficients

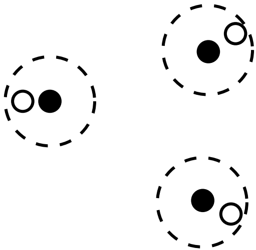
- **Theorem (Schönhage, 85)** Let p and p^* polynomials of degree n, z_i roots of p, z_i^* roots of p^*, $|z_i| < 1$

  $$\mu \leq 2^{-7n} \quad \text{and} \quad |p - p^*| < \mu |p| \ .$$

  Then up to a permutation of the indices of the z_i^*

  $$|z_i^* - z_i| < 9\sqrt[n]{\mu} \ .$$

- apply with $9\sqrt[n]{\mu} \ll \sigma(p)$
- real roots correspond to real roots
- nonreal roots correspond to . . .
- $\sigma(p^*) \approx \sigma(p)$
- it suffices to enlarge intervals by $9\sqrt[n]{\mu}$
- but we do not know $\sigma(p)$



Roots Depend Continuously on Coefficients

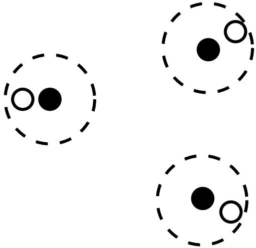
- **Theorem (Schönhage, 85)** Let p and p^* polynomials of degree n, z_i roots of p, z_i^* roots of p^*, $|z_i| < 1$

$$\mu \leq 2^{-7n} \quad \text{and} \quad |p - p^*| < \mu |p| \ .$$

Then up to a permutation of the indices of the z_i^*

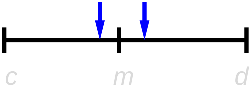
$$|z_i^* - z_i| < 9 \sqrt[n]{\mu} \ .$$

- apply with $9\sqrt[n]{\mu} \ll \sigma(p)$
- real roots correspond to real roots
- nonreal roots correspond to ...
- $\sigma(p^*) \approx \sigma(p)$
- it suffices to enlarge intervals by $9\sqrt[n]{\mu}$
- but we do not know $\sigma(p)$



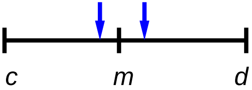
A Modified Algorithm for Isolating Roots in $I = (c, d)$

- let $I^+ = (c - 2(d-c), d + 2(d-c))$.
- if $var(p, I) = 0$ return;
- if $var(p, I) = 1$ and $var(p, I^+) = 1$ return and report (c, d)
- Let $m = (c+d)/2$ and if $p(m) = 0$ report $[m, m]$
- recurse on sub-intervals (c, m) and (m, d)
- Properties:
  - generates well-separated isolating intervals separation $\geq \sigma(p)/10$
  - refuses to be lucky, i.e, shortest interval generated has length $\approx \sigma(p)$ (assume $=$)



A Modified Algorithm for Isolating Roots in $I = (c, d)$

- let $I^+ = (c - 2(d-c), d + 2(d-c))$.
- if $var(p, I) = 0$ return;
- if $var(p, I) = 1$ and $var(p, I^+) = 1$ return and report (c, d)
- Let $m = (c+d)/2$ and if $p(m) = 0$ report $[m, m]$
- recurse on sub-intervals (c, m) and (m, d)
- Properties:
  - generates well-separated isolating intervals   separation $\geq \sigma(p)/10$
  - refuses to be lucky, i.e, shortest interval generated has length $\approx \sigma(p)$ (assume $=$)



The Master Algorithm

- let $\mu := 2^{-7n}$ so that Schönhage applies
- while true
  - let p^* be such that $|p - p^*| \le \mu |p|$
    roots move by at most $9\sqrt[n]{\mu}$ and hence $\sigma(p^*) \ge \sigma(p) - O(\sqrt[n]{\mu})$
    we want $9\sqrt[n]{\mu} \le \sigma(p^*)/10$
  - run modified algorithm on p^* shortest generated interval has length $\sigma(p^*)$
  - if alg produces an interval of length less than $\sqrt[n]{\mu}/90$
    then $\sigma(p^*) < \sqrt[n]{\mu}/90$, approximation not good enough)
  - stop alg, square μ and repeat
  - else exit from the loop
- alg ends with $\log \sqrt[n]{\mu} \approx \log \sigma(p)$

Analysis

- at termination: $\log \sqrt[n]{\mu} \approx \log \sigma(p)$ or $\log 1/\mu = n \log 1/\sigma(p)$
- recursion depth = $\log(M/\sigma(p))$ where $M =$length of start interval
- $\log M = O(\tau)$, thus depth = $O(\tau + \log 1/\sigma(p))$
- numbers grow by n bits in every node of the recursion tree
- so numbers grow to $\tau + \log 1/\mu + n \log(M/\sigma(p)) = \widetilde{O}(n(\tau + \log 1/sep(p)))$ bits
- $\widetilde{O}(n)$ arithmetic operations in every node
- width of tree is $O(n)$ since *var* is subadditive over intervals
- bit-complexity = $\widetilde{O}(n \cdot (\tau + \log 1/\sigma(p)) \cdot n \cdot n(\tau + \log 1/\sigma(p))) = \widetilde{O}(n^3(\tau + \log 1/\sigma(p))^2)$
- this assumes fast integer multiplication and Taylor shift

Experiments

- on polynomials with integer coefficients running time of standard Descartes and our version is about the same (give or take a factor of two)

- the big win: running time on $p(x)$ and $\pi \cdot p(x)$ is about the same, i.e.,
- running time depends on the geometry of the problem (distribution of roots in the plane) and not on the idiosyncrasy of the representation

Sagraloff's Improvements (2010)

- so far: $\tilde{O}(n(n\tau + n\log(1/\sigma(p)))^2)$ bit complexity.
- Sagraloff's new algorithm works with $\sum_\xi \log(1/\sigma(\xi))$ instead of $n\log 1/\sigma(p)$.
- bit complexity becomes $\tilde{O}(n(n\tau + \sum_\xi \log(1/\sigma(\xi)))^2)$
- for integer polynomials, this yields bit complexity $\tilde{O}(n^3\tau^2)$, an improvement by a factor of n
- for details, talk to Michael

Summary

- exact geometric computing has made a big step forward in the last decade
  - mature algorithms and software for 2d
  - first steps for 3d
- improved methods for isolating roots of real polynomials (bitstream coefficients) played a big rule.
- open problems:
  - improved bounds: see Sagraloff's new work (10)
  - Pan's method
  - 3d geometry