New Bounds in the Analysis of Subdivision Algorithms for Real Root Isolation

Chee Yap
Courant Institute of Mathematical Sciences
New York University

*Joint work with Vikram Sharma, Zilin Du, Chris Wu, Michael Burr

ABSTRACT

An important class of algorithms for isolating real roots of polynomials is based on the paradigm of subdivision, basically a form of binary search. This includes the Sturm algorithm and the Descartes method. A variant of the latter is applicable for polynomials in the Bernstein basis. Algorithms based on the Descartes method is very efficient in practice. In this talk, we introduce a third class of such algorithms, based on polynomial evaluation and interval arithmetic, derived from the work of Plantinga and Vegter on meshing surfaces. Almost no previous complexity analysis of such algorithms are known.

We describe three recent results in bounding the complexity of such algorithms:

(1) Simplified approach to the efficient evaluation of Sturm sequences. (2) Almost optimal bounds in the Descartes method. (3) Complexity analysis of an evaluation-based root isolation algorithm.

Papers

- Work is based on:
- "Amortized Bound for Root Isolation via Sturm Sequences"
  - ∗ Z.Du, V.Sharma, C.Yap.
    Workshop on Symbolic-Numeric Computation (SNC 2006)
- "Almost Tight Recursion Tree Bounds for the Descartes Method"
  - ∗ A.Eigenwilig, V.Sharma, C.Yap.
    (to appear) ISSAC 2006
- "Exact Evaluation-Based Root Isolation"
  - ∗ M.Burr, C.Wu, V.Sharma, C.Yap.
    Preprint, 2006

TALK OVERVIEW

- Subdivision Algorithms for Root Isolation
- Variations on Mahler-Davenport Bound
- Simplified Approach to Sturm Method
- Almost Optimal Bound for Descartes Method
- Evaluation Based Isolation Method
- Conclusion

TALK OVERVIEW

- Subdivision Algorithms for Root Isolation
- Variations on Mahler-Davenport Bound
- Simplified Approach to Sturm Method
- Almost Optimal Bound for Descartes Method
- Evaluation Based Isolation Method
- Conclusion

TALK OVERVIEW

- Subdivision Algorithms for Root Isolation
- Variations on Mahler-Davenport Bound
- Simplified Approach to Sturm Method
- Almost Optimal Bound for Descartes Method
- Evaluation Based Isolation Method
- Conclusion

TALK OVERVIEW

- Subdivision Algorithms for Root Isolation
- Variations on Mahler-Davenport Bound
- Simplified Approach to Sturm Method
- Almost Optimal Bound for Descartes Method
- Evaluation Based Isolation Method
- Conclusion

TALK OVERVIEW

- Subdivision Algorithms for Root Isolation
- Variations on Mahler-Davenport Bound
- Simplified Approach to Sturm Method
- Almost Optimal Bound for Descartes Method
- Evaluation Based Isolation Method
- Conclusion

TALK OVERVIEW

- Subdivision Algorithms for Root Isolation
- Variations on Mahler-Davenport Bound
- Simplified Approach to Sturm Method
- Almost Optimal Bound for Descartes Method
- Evaluation Based Isolation Method
- Conclusion

TALK OVERVIEW

- Subdivision Algorithms for Root Isolation
- Variations on Mahler-Davenport Bound
- Simplified Approach to Sturm Method
- Almost Optimal Bound for Descartes Method
- Evaluation Based Isolation Method
- Conclusion

I. Subdivision Methods in Root Isolation

The Real Root Isolation Problem

- PROBLEM: Isolate all the real zeros of f in I.
  - Input: interval $I = [a, b]$ and a function $f : \mathbb{R} \longrightarrow \mathbb{R}$
  - Output: set of isolating intervals, $J_1, J_2, \ldots, J_m$
  - Each $J_i \subseteq I$ contains a unique zero of f

- Complexity of Root Isolation
  - Integer polynomial $f(X)$
  - $I =$ interval containing all real zeros of $f(X)$
  - $d =$ degree, $L =$ maximum coefficient bit sizes

- Schönhage ('82) : approximate linear factorization approach
  - Implies root isolation in time $\widetilde{O}(d^3L)$
  - Classical/Numerical Analysis literature: Weierstrass, Weyl, Aberth, Traub, etc.
  - Complexity literature: Renegar'87, Kim-Sutherland'94, Neff-Reif'96, Pan'96, etc.

  - No improvement on Schönhage's bound

The Real Root Isolation Problem

- PROBLEM: Isolate all the real zeros of f in I.
  - **Input:** interval $I = [a, b]$ and a function $f : \mathbb{R} \longrightarrow \mathbb{R}$
  - **Output:** set of isolating intervals, $J_1, J_2, \ldots, J_m$
  - Each $J_i \subseteq I$ contains a unique zero of f

- Complexity of Root Isolation
  - Integer polynomial $f(X)$
  - $I =$ interval containing all real zeros of $f(X)$
  - $d =$ degree, $L =$ maximum coefficient bit sizes

- Schönhage ('82) : approximate linear factorization approach
  - Implies root isolation in time $\widetilde{O}(d^3L)$
  - Classical/Numerical Analysis literature: Weierstrass, Weyl, Aberth, Traub, etc.
  - Complexity literature: Renegar'87, Kim-Sutherland'94, Neff-Reif'96, Pan'96, etc.

  - No improvement on Schönhage's bound

The Real Root Isolation Problem

- PROBLEM: Isolate all the real zeros of f in I.
  - ∗ **Input:** interval $I = [a, b]$ and a function $f : \mathbb{R} \longrightarrow \mathbb{R}$
  - ∗ **Output:** set of isolating intervals, $J_1, J_2, \ldots, J_m$
  - ∗ Each $J_i \subseteq I$ contains a unique zero of f

- Complexity of Root Isolation
  - ∗ Integer polynomial $f(X)$
  - ∗ $I =$ interval containing all real zeros of $f(X)$
  - ∗ $d =$ degree, $L =$ maximum coefficient bit sizes

- Schönhage ('82) : approximate linear factorization approach
  - ∗ Implies root isolation in time $\widetilde{O}(d^3 L)$
  - ∗ Classical/Numerical Analysis literature: Weierstrass, Weyl, Aberth, Traub, etc.
  - ∗ Complexity literature: Renegar'87, Kim-Sutherland'94, Neff-Reif'96, Pan'96, etc.

  - ∗ No improvement on Schönhage's bound

The Real Root Isolation Problem

- PROBLEM: Isolate all the real zeros of f in I.
  - **Input:** interval $I = [a, b]$ and a function $f : \mathbb{R} \longrightarrow \mathbb{R}$
  - **Output:** set of isolating intervals, $J_1, J_2, \ldots, J_m$
  - Each $J_i \subseteq I$ contains a unique zero of f

- Complexity of Root Isolation
  - Integer polynomial $f(X)$
  - $I =$ interval containing all real zeros of $f(X)$
  - $d =$ degree, $L =$ maximum coefficient bit sizes

- Schönhage ('82) : approximate linear factorization approach
  - Implies root isolation in time $\widetilde{O}(d^3 L)$
  - Classical/Numerical Analysis literature: Weierstrass, Weyl, Aberth, Traub, etc.
  - Complexity literature: Renegar'87, Kim-Sutherland'94, Neff-Reif'96, Pan'96, etc.

  - No improvement on Schönhage's bound

Subdivision Algorithms

- Assume a Root Estimation Procedure, $EST_f(I)$
  - Let $\#_f(J)$ be number of zeros of f in J
  - EST returns $\#(J) = 0$ or $\#(J) = 1$ or $\#(J) = wha??$

- Generic Subdivision Algorithm (on input I, f)
  - Initialize queue Q with I
  - While Q is non-empty
  - $J \leftarrow Q.remove()$
  - Call procedure $EST_f(J)$
  - If $\#(J) = 0$, discard J
  - Else if $\#(J) = 1$, output J
  - Else split J into (J_0, J_1), and put J_0, J_1 into Q

- When is this an algorithm for root isolation?
  - Only termination is at issue

Subdivision Algorithms

- Assume a Root Estimation Procedure, $EST_f(I)$
  - Let $\#_f(J)$ be number of zeros of f in J
  - EST returns $\#(J) = 0$ or $\#(J) = 1$ or $\#(J) = wha??$

- Generic Subdivision Algorithm (on input I, f)
  - Initialize queue Q with I
  - While Q is non-empty
  - $J \leftarrow Q.remove()$
  - Call procedure $EST_f(J)$
  - If $\#(J) = 0$, discard J
  - Else if $\#(J) = 1$, output J
  - Else split J into (J_0, J_1), and put J_0, J_1 into Q

- When is this an algorithm for root isolation?
  - Only termination is at issue

Subdivision Algorithms

- Assume a Root Estimation Procedure, $EST_f(I)$
  - Let $\#_f(J)$ be number of zeros of f in J
  - EST returns $\#(J) = 0$ or $\#(J) = 1$ or $\#(J) = wha??$
- Generic Subdivision Algorithm (on input I, f)
  - Initialize queue Q with I
  - While Q is non-empty
  - $J \leftarrow Q.remove()$
  - Call procedure $EST_f(J)$
  - If $\#(J) = 0$, discard J
  - Else if $\#(J) = 1$, output J
  - Else split J into (J_0, J_1), and put J_0, J_1 into Q
- When is this an algorithm for root isolation?
  - Only termination is at issue

Subdivision Algorithms

- Assume a Root Estimation Procedure, $EST_f(I)$
  - Let $\#_f(J)$ be number of zeros of f in J
  - EST returns $\#(J) = 0$ or $\#(J) = 1$ or $\#(J) = wha??$

- Generic Subdivision Algorithm (on input I, f)
  - Initialize queue Q with I
  - While Q is non-empty
  - $J \leftarrow Q.remove()$
  - Call procedure $EST_f(J)$
  - If $\#(J) = 0$, discard J
  - Else if $\#(J) = 1$, output J
  - Else split J into (J_0, J_1), and put J_0, J_1 into Q

- When is this an algorithm for root isolation?
  - Only termination is at issue

Analysis of Three Algorithms

- Three examples of Procedure EST
  - Sturm procedure: returns $\#(J)$
  - Descartes procedure: returns $\#(J) = 0, 1$ or $\#(J) + 2k$ (unknown k)
  - Evaluation procedure: return $\#(J) = 0$ or $\#(J) = 1$ or $\#(J) = wha??$
- Bernstein basis variant of Descartes
  - Lane-Riesenfeld'81, Mourrain-Rouillier-Roy'05, Emiris-Mourrain-Tsigaridas'06
- THIS TALK : the complexity analysis of these 3 algorithms
  - Common Themes: amortized analysis, Davenport-Mahler Bounds

Analysis of Three Algorithms

- Three examples of Procedure EST
  - Sturm procedure: returns $\#(J)$
  - Descartes procedure: returns $\#(J) = 0, 1$ or $\#(J) + 2k$ (unknown k)
  - Evaluation procedure: return $\#(J) = 0$ or $\#(J) = 1$ or $\#(J) = wha??$
- Bernstein basis variant of Descartes
  - Lane-Riesenfeld'81, Mourrain-Rouillier-Roy'05, Emiris-Mourrain-Tsigaridas'06
- THIS TALK : the complexity analysis of these 3 algorithms
  - Common Themes: amortized analysis, Davenport-Mahler Bounds

Analysis of Three Algorithms

- Three examples of Procedure EST
  - Sturm procedure: returns $\#(J)$
  - Descartes procedure: returns $\#(J) = 0, 1$ or $\#(J) + 2k$ (unknown k)
  - Evaluation procedure: return $\#(J) = 0$ or $\#(J) = 1$ or $\#(J) = wha??$
- Bernstein basis variant of Descartes
  - Lane-Riesenfeld'81, Mourrain-Rouillier-Roy'05, Emiris-Mourrain-Tsigaridas'06
- THIS TALK : the complexity analysis of these 3 algorithms
  - Common Themes: amortized analysis, Davenport-Mahler Bounds

Analysis of Three Algorithms

- Three examples of Procedure EST
  - Sturm procedure: returns $\#(J)$
  - Descartes procedure: returns $\#(J) = 0, 1$ or $\#(J) + 2k$ (unknown k)
  - Evaluation procedure: return $\#(J) = 0$ or $\#(J) = 1$ or $\#(J) = wha??$
- Bernstein basis variant of Descartes
  - Lane-Riesenfeld'81, Mourrain-Rouillier-Roy'05, Emiris-Mourrain-Tsigaridas'06
- THIS TALK : the complexity analysis of these 3 algorithms
  - Common Themes: amortized analysis, Davenport-Mahler Bounds

REMARKS

- Relative Advantages of Methods:
  - Root approximation [Schonhage, Aberth] – simultaneously find all zeros
  - Subdivision approach – find zeros in a region
  - Adaptivity – inversely proportional to the cost at each node of the recursion tree

- Empirically: Descartes is superior to Sturm
  - Collins-Akritas'76,Johnson'98,Rouillier-Zimmermann'01
  - But Descartes is harder to analyze than Sturm [Davenport'85,Krandick'95]
  - So until recently, bound for Descartes is worst than for Sturm

- No previous complexity analysis of Evaluation based method known

REMARKS

- Relative Advantages of Methods:
  - Root approximation [Schonhage, Aberth] – simultaneously find all zeros
  - Subdivision approach – find zeros in a region
  - Adaptivity – inversely proportional to the cost at each node of the recursion tree

- Empirically: Descartes is superior to Sturm
  - Collins-Akritas'76,Johnson'98,Rouillier-Zimmermann'01
  - But Descartes is harder to analyze than Sturm [Davenport'85,Krandick'95]
  - So until recently, bound for Descartes is worst than for Sturm

- No previous complexity analysis of Evaluation based method known

REMARKS

- Relative Advantages of Methods:
  - ∗ Root approximation [Schonhage, Aberth] – simultaneously find all zeros
  - ∗ Subdivision approach – find zeros in a region
  - ∗ Adaptivity – inversely proportional to the cost at each node of the recursion tree

- Empirically: Descartes is superior to Sturm
  - ∗ Collins-Akritas'76,Johnson'98,Rouillier-Zimmermann'01
  - ∗ But Descartes is harder to analyze than Sturm [Davenport'85,Krandick'95]
  - ∗ So until recently, bound for Descartes is worst than for Sturm

- No previous complexity analysis of Evaluation based method known

REMARKS

- Relative Advantages of Methods:
  - ∗ Root approximation [Schonhage, Aberth] – simultaneously find all zeros
  - ∗ Subdivision approach – find zeros in a region
  - ∗ Adaptivity – inversely proportional to the cost at each node of the recursion tree
- Empirically: Descartes is superior to Sturm
  - ∗ Collins-Akritas'76,Johnson'98,Rouillier-Zimmermann'01
  - ∗ But Descartes is harder to analyze than Sturm [Davenport'85,Krandick'95]
  - ∗ So until recently, bound for Descartes is worst than for Sturm
- No previous complexity analysis of Evaluation based method known

II. Variations on Davenport-Mahler Bound

A Generalized Form

- THEOREM: Let $f(X) \in \mathbb{Z}[X]$ be square-free, of degree d. If $\alpha_1 < \beta_1 \leq \alpha_2 < \beta_2 \leq \alpha_3 < \cdots < \alpha_k \leq \beta_k$ are real roots of $f(X)$, then

$$\prod_{i=1}^{k} |\alpha_k - \beta_k| \begin{cases} \leq & M(f) \\ \geq & M(f)^{-d+1} d^{-d/2} (\sqrt{3}/d)^k \end{cases}$$

- IDEA of applications
  - E.g., To bound the size of recursion tree,
  - express the size as a log of the product of distances among suitable pairs of roots
  - E.g., for Evaluation Method, also look at critical points

- Upper Bound Proof
  - CASE A: Suppose exists an $h = 1, \ldots, k$ such that $\alpha_h < 0 < \beta_h$.

$$\prod_{i=1}^{k} |\beta_i - \alpha_i| \quad = \quad \left(\prod_{i=1}^{h-1} (\beta_i - \alpha_i) \right) \cdot |\alpha_h - \beta_h| \cdot \left(\prod_{i=h+1}^{k} (\beta_i - \alpha_i) \right)$$

A Generalized Form

- THEOREM: Let $f(X) \in \mathbb{Z}[X]$ be square-free, of degree d. If $\alpha_1 < \beta_1 \leq \alpha_2 < \beta_2 \leq \alpha_3 < \cdots < \alpha_k \leq \beta_k$ are real roots of $f(X)$, then

$$\prod_{i=1}^{k} |\alpha_k - \beta_k| \begin{cases} \leq & M(f) \\ \geq & M(f)^{-d+1} d^{-d/2} (\sqrt{3}/d)^k \end{cases}$$

- IDEA of applications
  - E.g., To bound the size of recursion tree,
  - express the size as a log of the product of distances among suitable pairs of roots
  - E.g., for Evaluation Method, also look at critical points

- Upper Bound Proof
  - CASE A: Suppose exists an $h = 1, \ldots, k$ such that $\alpha_h < 0 < \beta_h$.

$$\prod_{i=1}^{k} |\beta_i - \alpha_i| \quad = \quad \left(\prod_{i=1}^{h-1} (\beta_i - \alpha_i) \right) \cdot |\alpha_h - \beta_h| \cdot \left(\prod_{i=h+1}^{k} (\beta_i - \alpha_i) \right)$$

A Generalized Form

- THEOREM: Let $f(X) \in \mathbb{Z}[X]$ be square-free, of degree d. If $\alpha_1 < \beta_1 \leq \alpha_2 < \beta_2 \leq \alpha_3 < \cdots < \alpha_k \leq \beta_k$ are real roots of $f(X)$, then

$$\prod_{i=1}^{k} |\alpha_k - \beta_k| \begin{cases} \leq & M(f) \\ \geq & M(f)^{-d+1} d^{-d/2} (\sqrt{3}/d)^k \end{cases}$$

- IDEA of applications
  - E.g., To bound the size of recursion tree,
  - express the size as a log of the product of distances among suitable pairs of roots
  - E.g., for Evaluation Method, also look at critical points

- Upper Bound Proof
  - CASE A: Suppose exists an $h = 1, \ldots, k$ such that $\alpha_h < 0 < \beta_h$.

$$\prod_{i=1}^{k} |\beta_i - \alpha_i| = \left(\prod_{i=1}^{h-1} (\beta_i - \alpha_i)\right) \cdot |\alpha_h - \beta_h| \cdot \left(\prod_{i=h+1}^{k} (\beta_i - \alpha_i)\right)$$

A Generalized Form

- THEOREM: Let $f(X) \in \mathbb{Z}[X]$ be square-free, of degree d. If $\alpha_1 < \beta_1 \le \alpha_2 < \beta_2 \le \alpha_3 < \cdots < \alpha_k \le \beta_k$ are real roots of $f(X)$, then

$$\prod_{i=1}^{k} |\alpha_k - \beta_k| \begin{cases} \le & M(f) \\ \ge & M(f)^{-d+1} d^{-d/2} (\sqrt{3}/d)^k \end{cases}$$

- IDEA of applications
  - $*$ E.g., To bound the size of recursion tree,
  - $*$ express the size as a log of the product of distances among suitable pairs of roots
  - $*$ E.g., for Evaluation Method, also look at critical points

- Upper Bound Proof
  - $*$ CASE A: Suppose exists an $h = 1, \ldots, k$ such that $\alpha_h < 0 < \beta_h$.

$$\prod_{i=1}^{k} |\beta_i - \alpha_i| = \left(\prod_{i=1}^{h-1} (\beta_i - \alpha_i) \right) \cdot |\alpha_h - \beta_h| \cdot \left(\prod_{i=h+1}^{k} (\beta_i - \alpha_i) \right)$$

A Generalized Form

- THEOREM: Let $f(X) \in \mathbb{Z}[X]$ be square-free, of degree d. If $\alpha_1 < \beta_1 \leq \alpha_2 < \beta_2 \leq \alpha_3 < \cdots < \alpha_k \leq \beta_k$ are real roots of $f(X)$, then

$$\prod_{i=1}^{k} |\alpha_k - \beta_k| \begin{cases} \leq & M(f) \\ \geq & M(f)^{-d+1} d^{-d/2} (\sqrt{3}/d)^k \end{cases}$$

- IDEA of applications
  - E.g., To bound the size of recursion tree,
  - express the size as a log of the product of distances among suitable pairs of roots
  - E.g., for Evaluation Method, also look at critical points

- Upper Bound Proof
  - CASE A: Suppose exists an $h = 1, \ldots, k$ such that $\alpha_h < 0 < \beta_h$.

$$\prod_{i=1}^{k} |\beta_i - \alpha_i| = \left(\prod_{i=1}^{h-1} (\beta_i - \alpha_i)\right) \cdot |\alpha_h - \beta_h| \cdot \left(\prod_{i=h+1}^{k} (\beta_i - \alpha_i)\right)$$

$$\leq \left(\prod_{i=1}^{h-1} |\alpha_i|\right) \cdot (|\alpha_h| + |\beta_h|) \cdot \left(\prod_{i=h+1}^{k} |\beta_i|\right)$$

$$\leq (\prod_{i=1}^{d} \max\{1, |\gamma_i|\}|) \qquad \text{(the } \gamma_i\text{'s are distinct } \alpha_j, \beta_j\text{'s)}$$

$$\leq M(A).$$

- Lower Bound Proof (Davenport)

$*$ First assume $A(X)$ is monic. Then $\sqrt{|\mathrm{disc}(A)|} = \pm \det V$ for a Vandermond matrix V.

$*$ The jth column of V has form $(1, \gamma, \gamma^2, \ldots, \gamma^{d-1})^T$.

$*$ Extract the factors $(\alpha_i - \beta_i)$ from determinant by subtracting column of α_i from column of β_i.

$*$ Column of β_i is modified but its 2-norm can be bounded by $\sqrt{d^3/3}\max\{1, |\beta_i|\}$

$*$ Finally, use Hadamard's bound. Mahler measure $M(A)$ reappears as above.

$$
\begin{aligned}
&\leq \left(\prod_{i=1}^{h-1} |\alpha_i|\right) \cdot (|\alpha_h| + |\beta_h|) \cdot \left(\prod_{i=h+1}^{k} |\beta_i|\right) \\
&\leq \left(\prod_{i=1}^{d} \max\{1, |\gamma_i|\}|\right) \qquad \text{(the } \gamma_i\text{'s are distinct } \alpha_j, \beta_j\text{'s)} \\
&\leq M(A).
\end{aligned}
$$

- Lower Bound Proof (Davenport)

  $*$ First assume $A(X)$ is monic. Then $\sqrt{|\mathrm{disc}(A)|} = \pm \det V$ for a Vandermond matrix V.

  $*$ The jth column of V has form $(1, \gamma, \gamma^2, \ldots, \gamma^{d-1})^T$.

  $*$ Extract the factors $(\alpha_i - \beta_i)$ from determinant by subtracting column of α_i from column of β_i.

  $*$ Column of β_i is modified but its 2-norm can be bounded by $\sqrt{d^3/3}\max\{1, |\beta_i|\}$

  $*$ Finally, use Hadamard's bound. Mahler measure $M(A)$ reappears as above.

$$\leq \left(\prod_{i=1}^{h-1} |\alpha_i|\right) \cdot (|\alpha_h| + |\beta_h|) \cdot \left(\prod_{i=h+1}^{k} |\beta_i|\right)$$

$$\leq (\prod_{i=1}^{d} \max\{1, |\gamma_i|\}|) \qquad \text{(the } \gamma_i\text{'s are distinct } \alpha_j, \beta_j\text{'s)}$$

$$\leq M(A).$$

- Lower Bound Proof (Davenport)

  ∗ First assume $A(X)$ is monic. Then $\sqrt{|\mathrm{disc}(A)|} = \pm \det V$ for a Vandermond matrix V.

  ∗ The jth column of V has form $(1, \gamma, \gamma^2, \ldots, \gamma^{d-1})^T$.

  ∗ Extract the factors $(\alpha_i - \beta_i)$ from determinant by subtracting column of α_i from column of β_i.

  ∗ Column of β_i is modified but its 2-norm can be bounded by $\sqrt{d^3/3}\max\{1, |\beta_i|\}$

  ∗ Finally, use Hadamard's bound. Mahler measure $M(A)$ reappears as above.

$$\leq \left(\prod_{i=1}^{h-1} |\alpha_i|\right) \cdot (|\alpha_h| + |\beta_h|) \cdot \left(\prod_{i=h+1}^{k} |\beta_i|\right)$$

$$\leq (\prod_{i=1}^{d} \max\{1, |\gamma_i|\}|) \qquad \text{(the } \gamma_i\text{'s are distinct } \alpha_j, \beta_j\text{'s)}$$

$$\leq M(A).$$

- Lower Bound Proof (Davenport)

  $*$ First assume $A(X)$ is monic. Then $\sqrt{|\mathrm{disc}(A)|} = \pm \det V$ for a Vandermond matrix V.

  $*$ The jth column of V has form $(1, \gamma, \gamma^2, \ldots, \gamma^{d-1})^T$.

  $*$ Extract the factors $(\alpha_i - \beta_i)$ from determinant by subtracting column of α_i from column of β_i.

  $*$ Column of β_i is modified but its 2-norm can be bounded by $\sqrt{d^3/3}\max\{1, |\beta_i|\}$

  $*$ Finally, use Hadamard's bound. Mahler measure $M(A)$ reappears as above.

$$
\begin{aligned}
&\le \left(\prod_{i=1}^{h-1} |\alpha_i|\right) \cdot (|\alpha_h| + |\beta_h|) \cdot \left(\prod_{i=h+1}^{k} |\beta_i|\right) \\
&\le (\prod_{i=1}^{d} \max\{1, |\gamma_i|\}|) \qquad \text{(the } \gamma_i\text{'s are distinct } \alpha_j, \beta_j\text{'s)} \\
&\le M(A).
\end{aligned}
$$

- Lower Bound Proof (Davenport)

  $*$ First assume $A(X)$ is monic. Then $\sqrt{|\mathrm{disc}(A)|} = \pm \det V$ for a Vandermond matrix V.

  $*$ The jth column of V has form $(1, \gamma, \gamma^2, \ldots, \gamma^{d-1})^T$.

  $*$ Extract the factors $(\alpha_i - \beta_i)$ from determinant by subtracting column of α_i from column of β_i.

  $*$ Column of β_i is modified but its 2-norm can be bounded by $\sqrt{d^3/3}\max\{1, |\beta_i|\}$

  $*$ Finally, use Hadamard's bound. Mahler measure $M(A)$ reappears as above.

$$\leq \left(\prod_{i=1}^{h-1} |\alpha_i|\right) \cdot (|\alpha_h| + |\beta_h|) \cdot \left(\prod_{i=h+1}^{k} |\beta_i|\right)$$

$$\leq (\prod_{i=1}^{d} \max\{1, |\gamma_i|\}|) \qquad \text{(the } \gamma_i\text{'s are distinct } \alpha_j, \beta_j\text{'s)}$$

$$\leq M(A).$$

- Lower Bound Proof (Davenport)

  $*$ First assume $A(X)$ is monic. Then $\sqrt{|\mathrm{disc}(A)|} = \pm \det V$ for a Vandermond matrix V.

  $*$ The jth column of V has form $(1, \gamma, \gamma^2, \ldots, \gamma^{d-1})^T$.

  $*$ Extract the factors $(\alpha_i - \beta_i)$ from determinant by subtracting column of α_i from column of β_i.

  $*$ Column of β_i is modified but its 2-norm can be bounded by $\sqrt{d^3/3}\max\{1, |\beta_i|\}$

  $*$ Finally, use Hadamard's bound. Mahler measure $M(A)$ reappears as above.

$$\leq \left(\prod_{i=1}^{h-1} |\alpha_i|\right) \cdot (|\alpha_h| + |\beta_h|) \cdot \left(\prod_{i=h+1}^{k} |\beta_i|\right)$$

$$\leq (\prod_{i=1}^{d} \max\{1, |\gamma_i|\}|) \qquad \text{(the } \gamma_i\text{'s are distinct } \alpha_j, \beta_j\text{'s)}$$

$$\leq M(A).$$

- Lower Bound Proof (Davenport)

$*$ First assume $A(X)$ is monic. Then $\sqrt{|\mathrm{disc}(A)|} = \pm \det V$ for a Vandermond matrix V.

$*$ The jth column of V has form $(1, \gamma, \gamma^2, \ldots, \gamma^{d-1})^T$.

$*$ Extract the factors $(\alpha_i - \beta_i)$ from determinant by subtracting column of α_i from column of β_i.

$*$ Column of β_i is modified but its 2-norm can be bounded by $\sqrt{d^3/3}\max\{1, |\beta_i|\}$

$*$ Finally, use Hadamard's bound. Mahler measure $M(A)$ reappears as above.

III. Simplified Approach to Sturm Method

Analysis of Sturm Method

- Classic Approach [Collin-Loos 1983]

- Complexity Analysis in 3 parts
  - ∗ (A) Computing the Sturm sequence $S_f(X)$ of $f(X)$
  - ∗ (B) Size of the Recursion Tree
  - ∗ (C) Complexity of Evaluation $S_f(X)$ at a dL-bit rational $X = x_0$

- Best known bounds
  - ∗ (A) $\widetilde{O}(d^2L)$ [Reischert'97,Lickteig-Roy'01]
  - ∗ (B) $O(d(L + \log d))$ [Davenport'85,Du-Sharma-Y'05]
  - ∗ (C) $\widetilde{O}(d^3L)$ [Reischert'97,Lickteig-Roy'01,Du-Sharma-Y'05]
  - ∗ Overall: (A)+(B)(C)= $\widetilde{O}(d^4L^2)$

- Polynomial Remainder Sequences
  - ∗ GIVEN: $f(X) = f_0, g(X) = f_1$
  - ∗ COMPUTE: polynomial remainder sequence $PRS(f, g) = (f_0, f_1, f_2, \ldots, f_k)$
  - ∗ Where $\beta_i f_{i+1} = \alpha_i f_{i-1} - Q_i f_i$
  - ∗ Classic Subresultant PRS Algorithm [Collins,Brown-Traub]
  - ∗ With appropriate signs attached, we assume $PRS(f, g)$ is a Sturm sequence

Analysis of Sturm Method

- Classic Approach [Collin-Loos 1983]

- Complexity Analysis in 3 parts
  - (A) Computing the Sturm sequence $S_f(X)$ of $f(X)$
  - (B) Size of the Recursion Tree
  - (C) Complexity of Evaluation $S_f(X)$ at a dL-bit rational $X = x_0$

- Best known bounds
  - (A) $\widetilde{O}(d^2L)$ [Reischert'97,Lickteig-Roy'01]
  - (B) $O(d(L + \log d))$ [Davenport'85,Du-Sharma-Y'05]
  - (C) $\widetilde{O}(d^3L)$ [Reischert'97,Lickteig-Roy'01,Du-Sharma-Y'05]
  - Overall: (A)+(B)(C)= $\widetilde{O}(d^4L^2)$

- Polynomial Remainder Sequences
  - GIVEN: $f(X) = f_0, g(X) = f_1$
  - COMPUTE: polynomial remainder sequence $PRS(f, g) = (f_0, f_1, f_2, \ldots, f_k)$
  - Where $\beta_i f_{i+1} = \alpha_i f_{i-1} - Q_i f_i$
  - Classic Subresultant PRS Algorithm [Collins,Brown-Traub]
  - With appropriate signs attached, we assume $PRS(f, g)$ is a Sturm sequence

Analysis of Sturm Method

- Classic Approach [Collin-Loos 1983]

- Complexity Analysis in 3 parts
  - (A) Computing the Sturm sequence $S_f(X)$ of $f(X)$
  - (B) Size of the Recursion Tree
  - (C) Complexity of Evaluation $S_f(X)$ at a dL-bit rational $X = x_0$

- Best known bounds
  - (A) $\widetilde{O}(d^2L)$ [Reischert'97,Lickteig-Roy'01]
  - (B) $O(d(L + \log d))$ [Davenport'85,Du-Sharma-Y'05]
  - (C) $\widetilde{O}(d^3L)$ [Reischert'97,Lickteig-Roy'01,Du-Sharma-Y'05]
  - Overall: (A)+(B)(C)= $\widetilde{O}(d^4L^2)$

- Polynomial Remainder Sequences
  - GIVEN: $f(X) = f_0, g(X) = f_1$
  - COMPUTE: polynomial remainder sequence $PRS(f, g) = (f_0, f_1, f_2, \ldots, f_k)$
  - Where $\beta_i f_{i+1} = \alpha_i f_{i-1} - Q_i f_i$
  - Classic Subresultant PRS Algorithm [Collins,Brown-Traub]
  - With appropriate signs attached, we assume $PRS(f, g)$ is a Sturm sequence

Analysis of Sturm Method

- Classic Approach [Collin-Loos 1983]

- Complexity Analysis in 3 parts
  - (A) Computing the Sturm sequence $S_f(X)$ of $f(X)$
  - (B) Size of the Recursion Tree
  - (C) Complexity of Evaluation $S_f(X)$ at a dL-bit rational $X = x_0$

- Best known bounds
  - (A) $\widetilde{O}(d^2L)$ [Reischert'97,Lickteig-Roy'01]
  - (B) $O(d(L + \log d))$ [Davenport'85,Du-Sharma-Y'05]
  - (C) $\widetilde{O}(d^3L)$ [Reischert'97,Lickteig-Roy'01,Du-Sharma-Y'05]
  - Overall: (A)+(B)(C)= $\widetilde{O}(d^4L^2)$

- Polynomial Remainder Sequences
  - GIVEN: $f(X) = f_0, g(X) = f_1$
  - COMPUTE: polynomial remainder sequence $PRS(f, g) = (f_0, f_1, f_2, \ldots, f_k)$
  - Where $\beta_i f_{i+1} = \alpha_i f_{i-1} - Q_i f_i$
  - Classic Subresultant PRS Algorithm [Collins,Brown-Traub]
  - With appropriate signs attached, we assume $PRS(f, g)$ is a Sturm sequence

Analysis of Sturm Method

- Classic Approach [Collin-Loos 1983]

- Complexity Analysis in 3 parts
  - (A) Computing the Sturm sequence $S_f(X)$ of $f(X)$
  - (B) Size of the Recursion Tree
  - (C) Complexity of Evaluation $S_f(X)$ at a dL-bit rational $X = x_0$

- Best known bounds
  - (A) $\widetilde{O}(d^2L)$ [Reischert'97,Lickteig-Roy'01]
  - (B) $O(d(L + \log d))$ [Davenport'85,Du-Sharma-Y'05]
  - (C) $\widetilde{O}(d^3L)$ [Reischert'97,Lickteig-Roy'01,Du-Sharma-Y'05]
  - Overall: (A)+(B)(C)= $\widetilde{O}(d^4L^2)$

- Polynomial Remainder Sequences
  - GIVEN: $f(X) = f_0, g(X) = f_1$
  - COMPUTE: polynomial remainder sequence $PRS(f, g) = (f_0, f_1, f_2, \ldots, f_k)$
  - Where $\beta_i f_{i+1} = \alpha_i f_{i-1} - Q_i f_i$
  - Classic Subresultant PRS Algorithm [Collins,Brown-Traub]
  - With appropriate signs attached, we assume $PRS(f, g)$ is a Sturm sequence

Quotient Remainder Sequences

- Quotient Remainder Sequences [Strassen]
  - Represent $PRS(f,g)$ as $QRS(f,g) = (f_0, f_1, Q_1, Q_2, \ldots, Q_{k-1})$
  - Suffices to $QRS(f,g)$ at $X = x_0$
  - Number of arithmetic operations: $O(d)$ instead of $O(d^2)$

- What about bit complexity?
  - FACT: The coefficient bit sizes of Q_i's can be $\Omega(d^2 L)$
  - E.g., $f(X) = aX^{3d}$ and $g(X) = bX^{2d} + c$
  - So, naively, the complexity bound for (C) is $\widetilde{O}(d^4 L)$
  - SOLUTION 1: Reischert uses half-GCD idea
  - SOLUTION 2: Lickteig-Roy uses rational arithmetic

Quotient Remainder Sequences

- Quotient Remainder Sequences [Strassen]
  - Represent $PRS(f, g)$ as $QRS(f, g) = (f_0, f_1, Q_1, Q_2, \ldots, Q_{k-1})$
  - Suffices to $QRS(f, g)$ at $X = x_0$
  - Number of arithmetic operations: $O(d)$ instead of $O(d^2)$

- What about bit complexity?
  - FACT: The coefficient bit sizes of Q_i's can be $\Omega(d^2 L)$
  - E.g., $f(X) = aX^{3d}$ and $g(X) = bX^{2d} + c$
  - So, naively, the complexity bound for (C) is $\widetilde{O}(d^4 L)$
  - SOLUTION 1: Reischert uses half-GCD idea
  - SOLUTION 2: Lickteig-Roy uses rational arithmetic

Quotient Remainder Sequences

- Quotient Remainder Sequences [Strassen]
  - Represent $PRS(f, g)$ as $QRS(f, g) = (f_0, f_1, Q_1, Q_2, \ldots, Q_{k-1})$
  - Suffices to $QRS(f, g)$ at $X = x_0$
  - Number of arithmetic operations: $O(d)$ instead of $O(d^2)$

- What about bit complexity?
  - FACT: The coefficient bit sizes of Q_i's can be $\Omega(d^2 L)$
  - E.g., $f(X) = aX^{3d}$ and $g(X) = bX^{2d} + c$
  - So, naively, the complexity bound for (C) is $\widetilde{O}(d^4 L)$
  - SOLUTION 1: Reischert uses half-GCD idea
  - SOLUTION 2: Lickteig-Roy uses rational arithmetic

Simplified Approach to Best Sturm Bound

- Amortized Bound to the rescue
  - Bit size of $Q_i(X)$ is $O(\delta_i dL)$ where $\delta_i = \deg(Q_{i-1}) - \deg(Q_i)$
  - Thus, $\sum_i \delta_i = d$
- THEOREM: Using the Subresultant QRS for $QRS(f,g)$ and evaluate using Horner's rule, the bit complexity for (A) and (B) is $\widetilde{O}(d^3L)$.
  - NOTE: we bumped up the cost of (A) as the price of simplicity
- We achieved the current best bound using "straightforward" and "old" (ca. 1980) algorithms, but new analysis
  - Cf. Reischert, Lickteig-Roy

Simplified Approach to Best Sturm Bound

- Amortized Bound to the rescue
  - Bit size of $Q_i(X)$ is $O(\delta_i d L)$ where $\delta_i = \deg(Q_{i-1}) - \deg(Q_i)$
  - Thus, $\sum_i \delta_i = d$

- THEOREM: Using the Subresultant QRS for $QRS(f,g)$ and evaluate using Horner's rule, the bit complexity for (A) and (B) is $\widetilde{O}(d^3 L)$.
  - NOTE: we bumped up the cost of (A) as the price of simplicity

- We achieved the current best bound using "straightforward" and "old" (ca. 1980) algorithms, but new analysis
  - Cf. Reischert, Lickteig-Roy

Simplified Approach to Best Sturm Bound

- Amortized Bound to the rescue
  - Bit size of $Q_i(X)$ is $O(\delta_i dL)$ where $\delta_i = \deg(Q_{i-1}) - \deg(Q_i)$
  - Thus, $\sum_i \delta_i = d$

- THEOREM: Using the Subresultant QRS for $QRS(f, g)$ and evaluate using Horner's rule, the bit complexity for (A) and (B) is $\widetilde{O}(d^3 L)$.
  - NOTE: we bumped up the cost of (A) as the price of simplicity

- We achieved the current best bound using "straightforward" and "old" (ca. 1980) algorithms, but new analysis
  - Cf. Reischert, Lickteig-Roy

Simplified Approach to Best Sturm Bound

- Amortized Bound to the rescue
  - $*$ Bit size of $Q_i(X)$ is $O(\delta_i dL)$ where $\delta_i = \deg(Q_{i-1}) - \deg(Q_i)$
  - $*$ Thus, $\sum_i \delta_i = d$

- THEOREM: Using the Subresultant QRS for $QRS(f, g)$ and evaluate using Horner's rule, the bit complexity for (A) and (B) is $\widetilde{O}(d^3 L)$.
  - $*$ NOTE: we bumped up the cost of (A) as the price of simplicity

- We achieved the current best bound using "straightforward" and "old" (ca. 1980) algorithms, but new analysis
  - $*$ Cf. Reischert, Lickteig-Roy

IV. Almost Optimal Bounds for Descartes Method

Descartes Rule of Sign

- Descartes Rule of Signs:
  - If $f(X) = \sum_{i=0}^{d} a_i X^i$, then $\#(0, \infty)$ is equal to

$$Var(a_0, a_1, \ldots, a_d) - 2k, \qquad (k \geq 0)$$

  - Here, $Var(a_0, \ldots, a_d)$ is number of sign variations in the sequence
  - So, if $Var(a_0, \ldots, a_d) = 0$ or 1, then $k = 0$ (i.e., the estimate is exact!)

- Analogue in the Bernstein form: let $J = [u, v]$
  - Let $f_J(X) = \sum_{i=0}^{d} b_i B_i^d[u, v](X)$
  - where $B_i^d[u, v](X) = \binom{d}{i} \frac{(X-u)^i (v-X)^{d-i}}{(v-u)^d}$
  - Then the number of roots of $f(X)$ in J is equal to $Var(b_0, \ldots, b_d) - 2k$.
  - Note: Variation Diminishing Property of Bezier curves

Descartes Rule of Sign

- Descartes Rule of Signs:
  - If $f(X) = \sum_{i=0}^{d} a_i X^i$, then $\#(0, \infty)$ is equal to

$$Var(a_0, a_1, \ldots, a_d) - 2k, \qquad (k \geq 0)$$

  - Here, $Var(a_0, \ldots, a_d)$ is number of sign variations in the sequence
  - So, if $Var(a_0, \ldots, a_d) = 0$ or 1, then $k = 0$ (i.e., the estimate is exact!)

- Analogue in the Bernstein form: let $J = [u, v]$
  - Let $f_J(X) = \sum_{i=0}^{d} b_i B_i^d[u,v](X)$
  - where $B_i^d[u,v](X) = \binom{d}{i} \frac{(X-u)^i (v-X)^{d-i}}{(v-u)^d}$
  - Then the number of roots of $f(X)$ in J is equal to $Var(b_0, \ldots, b_d) - 2k$.
  - Note: Variation Diminishing Property of Bezier curves

Descartes Rule of Sign

- Descartes Rule of Signs:
  - ∗ If $f(X) = \sum_{i=0}^{d} a_i X^i$, then $\#(0, \infty)$ is equal to

$$Var(a_0, a_1, \ldots, a_d) - 2k, \qquad (k \geq 0)$$

  - ∗ Here, $Var(a_0, \ldots, a_d)$ is number of sign variations in the sequence
  - ∗ So, if $Var(a_0, \ldots, a_d) = 0$ or 1, then $k = 0$ (i.e., the estimate is exact!)

- Analogue in the Bernstein form: let $J = [u, v]$
  - ∗ Let $f_J(X) = \sum_{i=0}^{d} b_i B_i^d[u, v](X)$
  - ∗ where $B_i^d[u, v](X) = \binom{d}{i} \frac{(X-u)^i (v-X)^{d-i}}{(v-u)^d}$
  - ∗ Then the number of roots of $f(X)$ in J is equal to $Var(b_0, \ldots, b_d) - 2k$.
  - ∗ Note: Variation Diminishing Property of Bezier curves

Basis-Free Descartes Method

- How to implement $EST_f(I)$:

  ∗ (1) Transform f to $f_I(X)$ so that

$$|Zeros(f) \cap I| = \begin{cases} |Zeros(f_I) \cap (0, \infty)| & \text{in power basis} \\ |Zeros(f_I) \cap I| & \text{in Berstein basis} \end{cases}$$

  ∗ (2) Compute $Var(f_I)$

  ∗ NOTE: in Bernstein basis, f_I is as the same polynomial as f, but in different Bernstein basis

- Basis-free framework for Descartes Method

  ∗ For efficiency, we may store incremental transformations of f, etc

- Correctness (i.e., termination): [Ostrowski'50,Krandick-Mehlhorn'06]

  ∗ One Circle Theorem: If the disc $D_I \subseteq \mathbb{C}$ with diameter I does not contain any zeros of f (complex or real) then $EST(I) = 0$.

  ∗ Two Circle Theorem: If the union of two discs, $D_I^+ \cup D_I^-$ contains exact one real root of f and no other roots (complex or real) then $EST(I) = 1$.

  ∗ Corollary: every path terminates

Basis-Free Descartes Method

- How to implement $EST_f(I)$:
  - $*$ (1) Transform f to $f_I(X)$ so that

$$|Zeros(f) \cap I| = \begin{cases} |Zeros(f_I) \cap (0, \infty)| & \text{in power basis} \\ |Zeros(f_I) \cap I| & \text{in Berstein basis} \end{cases}$$

  - $*$ (2) Compute $Var(f_I)$
  - $*$ NOTE: in Bernstein basis, f_I is as the same polynomial as f, but in different Bernstein basis

- Basis-free framework for Descartes Method
  - $*$ For efficiency, we may store incremental transformations of f, etc

- Correctness (i.e., termination): [Ostrowski'50,Krandick-Mehlhorn'06]
  - $*$ One Circle Theorem: If the disc $D_I \subseteq \mathbb{C}$ with diameter I does not contain any zeros of f (complex or real) then $EST(I) = 0$.
  - $*$ Two Circle Theorem: If the union of two discs, $D_I^+ \cup D_I^-$ contains exact one real root of f and no other roots (complex or real) then $EST(I) = 1$.
  - $*$ Corollary: every path terminates

Basis-Free Descartes Method

- How to implement $EST_f(I)$:
  - (1) Transform f to $f_I(X)$ so that

$$|Zeros(f) \cap I| = \begin{cases} |Zeros(f_I) \cap (0, \infty)| & \text{in power basis} \\ |Zeros(f_I) \cap I| & \text{in Berstein basis} \end{cases}$$

  - (2) Compute $Var(f_I)$
  - NOTE: in Bernstein basis, f_I is as the same polynomial as f, but in different Bernstein basis

- Basis-free framework for Descartes Method
  - For efficiency, we may store incremental transformations of f, etc

- Correctness (i.e., termination): [Ostrowski'50,Krandick-Mehlhorn'06]
  - One Circle Theorem: If the disc $D_I \subseteq \mathbb{C}$ with diameter I does not contain any zeros of f (complex or real) then $EST(I) = 0$.
  - Two Circle Theorem: If the union of two discs, $D_I^+ \cup D_I^-$ contains exact one real root of f and no other roots (complex or real) then $EST(I) = 1$.
  - Corollary: every path terminates

Basis-Free Descartes Method

- How to implement $EST_f(I)$:

  $*$ (1) Transform f to $f_I(X)$ so that

$$|Zeros(f) \cap I| = \begin{cases} |Zeros(f_I) \cap (0, \infty)| & \text{in power basis} \\ |Zeros(f_I) \cap I| & \text{in Berstein basis} \end{cases}$$

  $*$ (2) Compute $Var(f_I)$

  $*$ NOTE: in Bernstein basis, f_I is as the same polynomial as f, but in different Bernstein basis

- Basis-free framework for Descartes Method

  $*$ For efficiency, we may store incremental transformations of f, etc

- Correctness (i.e., termination): [Ostrowski'50,Krandick-Mehlhorn'06]

  $*$ One Circle Theorem: If the disc $D_I \subseteq \mathbb{C}$ with diameter I does not contain any zeros of f (complex or real) then $EST(I) = 0$.

  $*$ Two Circle Theorem: If the union of two discs, $D_I^+ \cup D_I^-$ contains exact one real root of f and no other roots (complex or real) then $EST(I) = 1$.

  $*$ Corollary: every path terminates

Almost Optimal Bound on Recursion Tree Size

- Previous analysis:
  - $*$ Depth is $O(d(L + \log d)$. Hence size is exponential
  - $*$ Width is d. Hence size is $O(d^2(L + \log d)$
  - $*$ Size is $O(d^2(L + \log d) \log d)$ [Krandick]

- THEOREM : Recursion tree T size has $|T| = O(d(L + \log d)$
  - $*$ SKETCH: A leaf of T is type i if it has i roots $(i = 0, 1)$
  - $*$ Prune leaves which have non-leaf siblings
  - $*$ If two leaves are siblings, prune one of them (prefer to prune type 0 over type 1)
  - $*$ Let T' be pruned tree: $|T'| > |T|/2$
  - $*$ If $U =$ leaves of T', then $|U| \leq d$ and $|T'| \leq \sum_{u \in U} \log \frac{w(I_0)}{w(I_u)}$

- Reduce to Mahler-Davenport bound:
  - $*$ Type 1 node u: there is a real root α in I_u. Find another root β as its partner
  - $*$ Type 0 node v: similarly find a pair α, β of roots, perhaps complex
  - $*$ The $\{\alpha, \beta\}$ pairs are disjoint among the type 0, and among the type 1 nodes
  - $*$ Further, $w(I_u) \geq |\alpha - \beta|/2$
  - $*$ Thus, $|T'| \leq \log(MD) + d \log |I_0| = O(d(L + \log d))$

Almost Optimal Bound on Recursion Tree Size

- Previous analysis:
  - ∗ Depth is $O(d(L + \log d)$. Hence size is exponential
  - ∗ Width is d. Hence size is $O(d^2(L + \log d)$
  - ∗ Size is $O(d^2(L + \log d) \log d)$ [Krandick]

- THEOREM : Recursion tree T size has $|T| = O(d(L + \log d)$
  - ∗ SKETCH: A leaf of T is type i if it has i roots $(i = 0, 1)$
  - ∗ Prune leaves which have non-leaf siblings
  - ∗ If two leaves are siblings, prune one of them (prefer to prune type 0 over type 1)
  - ∗ Let T' be pruned tree: $|T'| > |T|/2$
  - ∗ If $U =$ leaves of T', then $|U| \le d$ and $|T'| \le \sum_{u \in U} \log \frac{w(I_0)}{w(I_u)}$

- Reduce to Mahler-Davenport bound:
  - ∗ Type 1 node u: there is a real root α in I_u. Find another root β as its partner
  - ∗ Type 0 node v: similarly find a pair α, β of roots, perhaps complex
  - ∗ The $\{\alpha, \beta\}$ pairs are disjoint among the type 0, and among the type 1 nodes
  - ∗ Further, $w(I_u) \ge |\alpha - \beta|/2$
  - ∗ Thus, $|T'| \le \log(MD) + d \log |I_0| = O(d(L + \log d))$

Almost Optimal Bound on Recursion Tree Size

- Previous analysis:
  - $*$ Depth is $O(d(L + \log d)$. Hence size is exponential
  - $*$ Width is d. Hence size is $O(d^2(L + \log d)$
  - $*$ Size is $O(d^2(L + \log d) \log d)$ [Krandick]

- THEOREM : Recursion tree T size has $|T| = O(d(L + \log d)$
  - $*$ SKETCH: A leaf of T is type i if it has i roots $(i = 0, 1)$
  - $*$ Prune leaves which have non-leaf siblings
  - $*$ If two leaves are siblings, prune one of them (prefer to prune type 0 over type 1)
  - $*$ Let T' be pruned tree: $|T'| > |T|/2$
  - $*$ If U = leaves of T', then $|U| \leq d$ and $|T'| \leq \sum_{u \in U} \log \frac{w(I_0)}{w(I_u)}$

- Reduce to Mahler-Davenport bound:
  - $*$ Type 1 node u: there is a real root α in I_u. Find another root β as its partner
  - $*$ Type 0 node v: similarly find a pair α, β of roots, perhaps complex
  - $*$ The $\{\alpha, \beta\}$ pairs are disjoint among the type 0, and among the type 1 nodes
  - $*$ Further, $w(I_u) \geq |\alpha - \beta|/2$
  - $*$ Thus, $|T'| \leq \log(MD) + d \log |I_0| = O(d(L + \log d))$

Almost Optimal Bound on Recursion Tree Size

- Previous analysis:
  - Depth is $O(d(L + \log d)$. Hence size is exponential
  - Width is d. Hence size is $O(d^2(L + \log d)$
  - Size is $O(d^2(L + \log d) \log d)$ [Krandick]

- THEOREM : Recursion tree T size has $|T| = O(d(L + \log d)$
  - SKETCH: A leaf of T is type i if it has i roots $(i = 0, 1)$
  - Prune leaves which have non-leaf siblings
  - If two leaves are siblings, prune one of them (prefer to prune type 0 over type 1)
  - Let T' be pruned tree: $|T'| > |T|/2$
  - If U = leaves of T', then $|U| \le d$ and $|T'| \le \sum_{u \in U} \log \frac{w(I_0)}{w(I_u)}$

- Reduce to Mahler-Davenport bound:
  - Type 1 node u: there is a real root α in I_u. Find another root β as its partner
  - Type 0 node v: similarly find a pair α, β of roots, perhaps complex
  - The $\{\alpha, \beta\}$ pairs are disjoint among the type 0, and among the type 1 nodes
  - Further, $w(I_u) \ge |\alpha - \beta|/2$
  - Thus, $|T'| \le \log(MD) + d \log |I_0| = O(d(L + \log d))$

- This bound is achieved for $L \geq \log d$
  * Use variations of an example from Mignotte

- THEOREM: Worst case complexity for real root isolation of square-free polynomials via Descartes method is $\widetilde{O}(d^4 L^2)$.
  * [Emiris-Mourrain-Tsigaridas'06] showed that the same bound applies to the non-square free case, by first removing the square-free part

- This bound is achieved for $L \geq \log d$
  - $*$ Use variations of an example from Mignotte

- THEOREM: Worst case complexity for real root isolation of square-free polynomials via Descartes method is $\widetilde{O}(d^4L^2)$.

  $*$ [Emiris-Mourrain-Tsigaridas'06] showed that the same bound applies to the non-square free case, by first removing the square-free part

- This bound is achieved for $L \geq \log d$
  - $*$ Use variations of an example from Mignotte

- THEOREM: Worst case complexity for real root isolation of square-free polynomials via Descartes method is $\widetilde{O}(d^4 L^2)$.

  $*$ [Emiris-Mourrain-Tsigaridas'06] showed that the same bound applies to the non-square free case, by first removing the square-free part

V. Evaluation Based Isolation Method

Meshing according to Vegter-Plantinga

- Meshing of an implicit curve $C : f(X, Y) = 0$
  - ∗ Want a polygonal curve C' that is ε-isotopic to C
  - ∗ I.e., C, C' are isotopic and there is a homeomorphism $h : C \to C'$ such that $\|h(p) - p'\| \le \varepsilon$ (for all p)
  - ∗ Previous work: [Stander-Hart'97] in SIGGRAPH, etc
  - ∗ Vegter-Plantinga'05 gave the first correct solution

- Vegter-Plantinga Model (Refinement of Marching Cube Model)
  - ∗ Use subdivision of space into a (non-uniform) grid
  - ∗ Primitive (1): (sign) evaluation of $f(X, Y)$ at grid points
  - ∗ Primitive (2): interval evaluation of $f(X, Y)$ and first derivatives at any rectangular box

- Let $\square\mathbb{R}$ be the set of (closed) intervals, $(\square\mathbb{R})^2 = \square\mathbb{R}^2$ be set of boxes
  - ∗ For any set $S \subseteq \mathbb{R}$, let $f(S) = \{f(p) : p \in S\}$
  - ∗ The interval analogue of f is $\square f : \square\mathbb{R}^2 \to \square\mathbb{R}$
  - ∗ Property (1): $f(B) \subseteq \square f(B)$
  - ∗ Property (2): If $B_i \to p \in \mathbb{R}^d$ as $i \to \infty$, then $\square f(B_i) \to f(p)$.

Meshing according to Vegter-Plantinga

- Meshing of an implicit curve $C : f(X,Y) = 0$
  - $*$ Want a polygonal curve C' that is ε-isotopic to C
  - $*$ I.e., C, C' are isotopic and there is a homeomorphism $h : C \to C'$ such that $\|h(p) - p'\| \le \varepsilon$ (for all p)
  - $*$ Previous work: [Stander-Hart'97] in SIGGRAPH, etc
  - $*$ Vegter-Plantinga'05 gave the first correct solution

- Vegter-Plantinga Model (Refinement of Marching Cube Model)
  - $*$ Use subdivision of space into a (non-uniform) grid
  - $*$ Primitive (1): (sign) evaluation of $f(X,Y)$ at grid points
  - $*$ Primitive (2): interval evaluation of $f(X,Y)$ and first derivatives at any rectangular box

- Let $\square\mathbb{R}$ be the set of (closed) intervals, $(\square\mathbb{R})^2 = \square\mathbb{R}^2$ be set of boxes
  - $*$ For any set $S \subseteq \mathbb{R}$, let $f(S) = \{f(p) : p \in S\}$
  - $*$ The interval analogue of f is $\square f : \square\mathbb{R}^2 \to \square\mathbb{R}$
  - $*$ Property (1): $f(B) \subseteq \square f(B)$
  - $*$ Property (2): If $B_i \to p \in \mathbb{R}^d$ as $i \to \infty$, then $\square f(B_i) \to f(p)$.

Meshing according to Vegter-Plantinga

- Meshing of an implicit curve $C : f(X,Y) = 0$
  - ∗ Want a polygonal curve C' that is ε-isotopic to C
  - ∗ I.e., C, C' are isotopic and there is a homeomorphism $h : C \to C'$ such that $\|h(p) - p'\| \le \varepsilon$ (for all p)
  - ∗ Previous work: [Stander-Hart'97] in SIGGRAPH, etc
  - ∗ Vegter-Plantinga'05 gave the first correct solution

- Vegter-Plantinga Model (Refinement of Marching Cube Model)
  - ∗ Use subdivision of space into a (non-uniform) grid
  - ∗ Primitive (1): (sign) evaluation of $f(X,Y)$ at grid points
  - ∗ Primitive (2): interval evaluation of $f(X,Y)$ and first derivatives at any rectangular box

- Let $\square\mathbb{R}$ be the set of (closed) intervals, $(\square\mathbb{R})^2 = \square\mathbb{R}^2$ be set of boxes
  - ∗ For any set $S \subseteq \mathbb{R}$, let $f(S) = \{f(p) : p \in S\}$
  - ∗ The interval analogue of f is $\square f : \square\mathbb{R}^2 \to \square\mathbb{R}$
  - ∗ Property (1): $f(B) \subseteq \square f(B)$
  - ∗ Property (2): If $B_i \to p \in \mathbb{R}^d$ as $i \to \infty$, then $\square f(B_i) \to f(p)$.

Meshing according to Vegter-Plantinga

- Meshing of an implicit curve $C : f(X,Y) = 0$
  - $*$ Want a polygonal curve C' that is ε-isotopic to C
  - $*$ I.e., C, C' are isotopic and there is a homeomorphism $h : C \to C'$ such that $\|h(p) - p'\| \leq \varepsilon$ (for all p)
  - $*$ Previous work: [Stander-Hart'97] in SIGGRAPH, etc
  - $*$ Vegter-Plantinga'05 gave the first correct solution

- Vegter-Plantinga Model (Refinement of Marching Cube Model)
  - $*$ Use subdivision of space into a (non-uniform) grid
  - $*$ Primitive (1): (sign) evaluation of $f(X,Y)$ at grid points
  - $*$ Primitive (2): interval evaluation of $f(X,Y)$ and first derivatives at any rectangular box

- Let $\square\mathbb{R}$ be the set of (closed) intervals, $(\square\mathbb{R})^2 = \square\mathbb{R}^2$ be set of boxes
  - $*$ For any set $S \subseteq \mathbb{R}$, let $f(S) = \{f(p) : p \in S\}$
  - $*$ The interval analogue of f is $\square f : \square\mathbb{R}^2 \to \square\mathbb{R}$
  - $*$ Property (1): $f(B) \subseteq \square f(B)$
  - $*$ Property (2): If $B_i \to p \in \mathbb{R}^d$ as $i \to \infty$, then $\square f(B_i) \to f(p)$.

Meshing in 1-D

- Let $f : \mathbb{R} \to \mathbb{R}$ be any continuous function
  - $*$ First, assume f has simple zeros only, with known root separation bound Δ

- Two Criteria:
  - $*$ $C_0(I)$: $0 \notin \square f(I)$
  - $*$ $C_1(I)$: $0 \notin \square f'(I)$ where $f'(X) = \frac{\partial f}{\partial X}(X)$

- LEMMA: Assume $C_1(I)$. Then $I = [a, b]$ has at most one zero of f. Moreover I contains a zero iff $f(a)f(b) \leq 0$.

Meshing in 1-D

- Let $f : \mathbb{R} \longrightarrow \mathbb{R}$ be any continuous function
  - First, assume f has simple zeros only, with known root separation bound Δ
- Two Criteria:
  - $C_0(I)$: $0 \notin \square f(I)$
  - $C_1(I)$: $0 \notin \square f'(I)$ where $f'(X) = \frac{\partial f}{\partial X}(X)$
- LEMMA: Assume $C_1(I)$. Then $I = [a, b]$ has at most one zero of f. Moreover I contains a zero iff $f(a)f(b) \leq 0$.

Meshing in 1-D

- Let $f : \mathbb{R} \longrightarrow \mathbb{R}$ be any continuous function
  - First, assume f has simple zeros only, with known root separation bound Δ
- Two Criteria:
  - $C_0(I)$: $0 \notin \square f(I)$
  - $C_1(I)$: $0 \notin \square f'(I)$ where $f'(X) = \frac{\partial f}{\partial X}(X)$
- LEMMA: Assume $C_1(I)$. Then $I = [a, b]$ has at most one zero of f. Moreover I contains a zero iff $f(a)f(b) \leq 0$.

Meshing in 1-D

- Let $f : \mathbb{R} \longrightarrow \mathbb{R}$ be any continuous function
  - First, assume f has simple zeros only, with known root separation bound Δ

- Two Criteria:
  - $C_0(I)$: $0 \notin \square f(I)$
  - $C_1(I)$: $0 \notin \square f'(I)$ where $f'(X) = \frac{\partial f}{\partial X}(X)$

- LEMMA: Assume $C_1(I)$. Then $I = [a, b]$ has at most one zero of f. Moreover I contains a zero iff $f(a)f(b) \leq 0$.

Simple Root Isolation by Evaluation

- Algorithm to isolate roots of f in I_0

SimpleRootIsol
Input: f and I_0
Initialize Q to I_0.
while Q is non-empty
1. $I = [a, b] \leftarrow Q.remove()$.
2. if (C_0) then DISCARD I.
3. else if (C_1)
   if $f(a)f(b) \leq 0$, OUTPUT(I)
   else discard I
4. else
   4.1 Let $m \leftarrow (a+b)/2$, $I_0 \leftarrow [a, m]$ and $I_1 \leftarrow [m, b]$.
   4.2 if $f(m) = 0$ then OUTPUT$([m, m])$ and adjust endpoints of I_0, I_1 by $\Delta/2$.
   4.3 Put I_0, I_1 into Q.

- Correctness
  - ∗ It is partially correct, i.e., it is correct if it halts
  - ∗ Termination: it it fails to halt, there is an infinite path
  - ∗ Let $I_0 \supseteq I_1 \supseteq \cdots \supseteq I_i \supseteq \cdots$ be intervals on this path
  - ∗ Then $I_i \longrightarrow x^*$, and so $f(I_i) \longrightarrow f(x^*)$
  - ∗ Moreover, $C_0(I_i)$ and $C_1(I_i)$ must fail for all i
  - ∗ In particular, $f(x^*) = 0$ and $f'(x^*) = 0$. Contradiction

Simple Root Isolation by Evaluation

- Algorithm to isolate roots of f in I_0

SimpleRootIsol
Input: f and I_0
Initialize Q to I_0.
while Q is non-empty
1. $I = [a, b] \leftarrow Q.remove()$.
2. if (C_0) then DISCARD I.
3. else if (C_1)
    if $f(a)f(b) \leq 0$, OUTPUT(I)
    else discard I
4. else
    4.1 Let $m \leftarrow (a+b)/2$, $I_0 \leftarrow [a, m]$ and $I_1 \leftarrow [m, b]$.
    4.2 if $f(m) = 0$ then OUTPUT($[m, m]$) and adjust endpoints of I_0, I_1 by $\Delta/2$.
    4.3 Put I_0, I_1 into Q.

- Correctness
  - It is partially correct, i.e., it is correct if it halts
  - Termination: it it fails to halt, there is an infinite path
  - Let $I_0 \supseteq I_1 \supseteq \cdots \supseteq I_i \supseteq \cdots$ be intervals on this path
  - Then $I_i \longrightarrow x^*$, and so $f(I_i) \longrightarrow f(x^*)$
  - Moreover, $C_0(I_i)$ and $C_1(I_i)$ must fail for all i
  - In particular, $f(x^*) = 0$ and $f'(x^*) = 0$. Contradiction

Simple Root Isolation by Evaluation

- Algorithm to isolate roots of f in I_0

SimpleRootIsol
Input: f and I_0
Initialize Q to I_0.
while Q is non-empty
1. $I = [a, b] \leftarrow Q.remove()$.
2. if (C_0) then DISCARD I.
3. else if (C_1)
   if $f(a)f(b) \le 0$, OUTPUT(I)
   else discard I
4. else
   4.1 Let $m \leftarrow (a+b)/2$, $I_0 \leftarrow [a, m]$ and $I_1 \leftarrow [m, b]$.
   4.2 if $f(m) = 0$ then OUTPUT$([m, m])$ and adjust endpoints of I_0, I_1 by $\Delta/2$.
   4.3 Put I_0, I_1 into Q.

- Correctness
  - $*$ It is partially correct, i.e., it is correct if it halts
  - $*$ Termination: it it fails to halt, there is an infinite path
  - $*$ Let $I_0 \supseteq I_1 \supseteq \cdots \supseteq I_i \supseteq \cdots$ be intervals on this path
  - $*$ Then $I_i \longrightarrow x^*$, and so $f(I_i) \longrightarrow f(x^*)$
  - $*$ Moreover, $C_0(I_i)$ and $C_1(I_i)$ must fail for all i
  - $*$ In particular, $f(x^*) = 0$ and $f'(x^*) = 0$. Contradiction

Complexity Analysis

- Assume that $\square f$ gives the ideal interval

- THEOREM: Assume a square-free polynomial of degree d and L bit coefficients, and initial interval $I_0 = [-2^{-L}, 2^{-L}]$, the recursion tree has size $O(d(L + \log d))$

  ∗ Idea of proof: we need Mahler-Davenport type bounds for distances between zeros of f and critical points of f

  ∗ Subdivide I_0 by real zeros and critical points

- Non-Ideal Case

  ∗ We can get a bound based on Lipschitz constants for the curve

  ∗ Complexity is unclear as parameters of d and L

  ∗ Dependence on the choice of box function

  ∗ Experimental results – not competitive with Descartes for common examples

Complexity Analysis

- Assume that $\square f$ gives the ideal interval

- THEOREM: Assume a square-free polynomial of degree d and L bit coefficients, and initial interval $I_0 = [-2^{-L}, 2^{-L}]$, the recursion tree has size $O(d(L + \log d))$

  ∗ Idea of proof: we need Mahler-Davenport type bounds for distances between zeros of f and critical points of f

  ∗ Subdivide I_0 by real zeros and critical points

- Non-Ideal Case
  - ∗ We can get a bound based on Lipschitz constants for the curve
  - ∗ Complexity is unclear as parameters of d and L
  - ∗ Dependence on the choice of box function
  - ∗ Experimental results – not competitive with Descartes for common examples

Complexity Analysis

- Assume that $\Box f$ gives the ideal interval

- THEOREM: Assume a square-free polynomial of degree d and L bit coefficients, and initial interval $I_0 = [-2^{-L}, 2^{-L}]$, the recursion tree has size $O(d(L + \log d))$

  * Idea of proof: we need Mahler-Davenport type bounds for distances between zeros of f and critical points of f

  * Subdivide I_0 by real zeros and critical points

- Non-Ideal Case

  * We can get a bound based on Lipschitz constants for the curve
  * Complexity is unclear as parameters of d and L
  * Dependence on the choice of box function
  * Experimental results – not competitive with Descartes for common examples

Complexity Analysis

- Assume that $\Box f$ gives the ideal interval

- THEOREM: Assume a square-free polynomial of degree d and L bit coefficients, and initial interval $I_0 = [-2^{-L}, 2^{-L}]$, the recursion tree has size $O(d(L + \log d))$

  * Idea of proof: we need Mahler-Davenport type bounds for distances between zeros of f and critical points of f
  * Subdivide I_0 by real zeros and critical points

- Non-Ideal Case
  * We can get a bound based on Lipschitz constants for the curve
  * Complexity is unclear as parameters of d and L
  * Dependence on the choice of box function
  * Experimental results – not competitive with Descartes for common examples

Evaluation and Root Separation Bounds

- Multiple Roots (singular case)

  * Let $K = K_I$ be a Lipschitz constant for I: for $x, y \in I$, $|f(x) - f(y)| \leq K|x - y|$.

  * Root separation bound: let $\Delta(f, g) = \min\{|\alpha - \beta| : \alpha \in Zero(f), \beta \in Zero(g), \alpha \neq \beta\}$

  * Evaluation bound: let $\mathrm{EV}(f, g) = \min\{|f(\alpha)| : \alpha \in Zero(f) \setminus Zero(g)\}$

  Write $\Delta(f) = \Delta(f, f)$ and $\mathrm{EV}(f) = \mathrm{EV}(f, f')$

Evaluation and Root Separation Bounds

- Multiple Roots (singular case)

  $*$ Let $K = K_I$ be a Lipschitz constant for I: for $x, y \in I$, $|f(x) - f(y)| \leq K|x - y|$.

  $*$ Root separation bound: let $\Delta(f,g) = \min\{|\alpha - \beta| : \alpha \in Zero(f), \beta \in Zero(g), \alpha \neq \beta\}$

  $*$ Evaluation bound: let $\mathrm{EV}(f,g) = \min\{|f(\alpha)| : \alpha \in Zero(f) \setminus Zero(g)\}$

  Write $\Delta(f) = \Delta(f,f)$ and $\mathrm{EV}(f) = \mathrm{EV}(f,f')$

Evaluation Based Isolation in the Singular Case

- NonAdaptic MultipleRoot Isolation
  - Subdivide as long as interval is larger than $\min\{\Delta(f), \Delta(f'), \Delta(f, f'), \mathrm{EV}(f)/3K, \mathrm{EV}(f', f)/2K'\}$

ROOTISOL Input: $F : \mathbb{R} \longrightarrow \mathbb{R}$, and interval I_0
Output: A list F of non-overlapping subintervals of I_0, representing isolating intervals for all the zeros of F in I_0. Each isolating interval is classified as singular or simple.

```
Initialize Q to I_0.
while Q ≠ ∅
    I = [a, b] ← Q.remove().
    if |I| > min{Δ(F), Δ(F'), Δ(F, F'), EV(F,F')/3K, EV(F',F)/3K'}
        if the midpoint of I, m = (a + b), is root
            if the midpoint of I is also a root of F', put [m, m] into L (singular).
            else put [m, m] into L (nonsingular)
        Split I in two equal halves, and put them in Q.
    else
        if (F'(a)F'(b) ≤ 0)
1.          if |f(a)| ≤ EV(F, F')/3, put I into L (singular).
2.          else Discard I.
        else —- Thus, F'(a)F'(b) > 0
            if (f(a)f(b) ≤ 0)
3.              if (|F'(a)| ≤ EV(F', F)/3), put I into L (singular)
4.              else put I into L (nonsingular).
5.          else Discard I.
```

Evaluation Based Isolation in the Singular Case

- NonAdaptic MultipleRoot Isolation

  $*$ Subdivide as long as interval is larger than $\min\{\Delta(f), \Delta(f'), \Delta(f, f'), \mathrm{EV}(f)/3K, \mathrm{EV}(f', f)/2K'\}$

ROOTISOL Input: $F : \mathbb{R} \to \mathbb{R}$, and interval I_0
Output: A list F of non-overlapping subintervals of I_0, representing isolating intervals for all the zeros of F in I_0. Each isolating interval is classified as singular or simple.
    Initialize Q to I_0.
    while $Q \neq \emptyset$
        $I = [a, b] \leftarrow Q.remove()$.
        if $|I| > \min\{\Delta(F), \Delta(F'), \Delta(F, F'), \frac{\mathrm{EV}(F,F')}{3K}, \frac{\mathrm{EV}(F',F)}{3K'}\}$
            if the midpoint of I, $m = (a + b)$, is root
                if the midpoint of I is also a root of F', put $[m, m]$ into L (singular).
                else put $[m, m]$ into L (nonsingular)
            Split I in two equal halves, and put them in Q.
        else
            if $(F'(a)F'(b) \leq 0)$
1.   if $|f(a)| \leq \mathrm{EV}(F, F')/3$, put I into L (singular).
2.   else Discard I.
            else —- Thus, $F'(a)F'(b) > 0$
                if $(f(a)f(b) \leq 0)$
3.   if $(|F'(a)| \leq \mathrm{EV}(F', F)/3)$, put I into L (singular)
4.   else put I into L (nonsingular).
5.   else Discard I.

- Adaptive version
  - ∗ Too long to fit the slide — see the end

- Adaptive version
  - ∗ Too long to fit the slide — see the end

VI. Conclusion

Summary and Open Problems

- Our results are based on:
  - Amortization arguments in algebraic complexity analysis
  - Effective use of Davenport-Mahler type bounds
- Decreasing complexity in the EST Routine:
  - Sturm $>$ Descartes $>$ Evaluation
  - Cheaper EST allows more adaptive complexity
  - This potential is not currently realized
- Open Problems:
  - Prove optimality of Sturm/Descartes method when $L = o(\log d)$
  - Develop practical techniques in Evaluation Based Isolation
  - Improved complexity analysis of Evaluation Based Isolation
  - Extend analysis to higher dimensional analogues
- Finally, a perspective for Evaluation Based Isolation:
  - Sturm and Descartes methods are now quite mature areas
  - But it took 3 decades of progress...
  - Cf. the optimal Sturm algorithm we presented

Summary and Open Problems

- Our results are based on:
  - Amortization arguments in algebraic complexity analysis
  - Effective use of Davenport-Mahler type bounds
- Decreasing complexity in the EST Routine:
  - Sturm $>$ Descartes $>$ Evaluation
  - Cheaper EST allows more adaptive complexity
  - This potential is not currently realized
- Open Problems:
  - Prove optimality of Sturm/Descartes method when $L = o(\log d)$
  - Develop practical techniques in Evaluation Based Isolation
  - Improved complexity analysis of Evaluation Based Isolation
  - Extend analysis to higher dimensional analogues
- Finally, a perspective for Evaluation Based Isolation:
  - Sturm and Descartes methods are now quite mature areas
  - But it took 3 decades of progress...
  - Cf. the optimal Sturm algorithm we presented

Summary and Open Problems

- Our results are based on:
  - Amortization arguments in algebraic complexity analysis
  - Effective use of Davenport-Mahler type bounds
- Decreasing complexity in the EST Routine:
  - Sturm $>$ Descartes $>$ Evaluation
  - Cheaper EST allows more adaptive complexity
  - This potential is not currently realized
- Open Problems:
  - Prove optimality of Sturm/Descartes method when $L = o(\log d)$
  - Develop practical techniques in Evaluation Based Isolation
  - Improved complexity analysis of Evaluation Based Isolation
  - Extend analysis to higher dimensional analogues
- Finally, a perspective for Evaluation Based Isolation:
  - Sturm and Descartes methods are now quite mature areas
  - But it took 3 decades of progress...
  - Cf. the optimal Sturm algorithm we presented

Summary and Open Problems

- Our results are based on:
  * Amortization arguments in algebraic complexity analysis
  * Effective use of Davenport-Mahler type bounds
- Decreasing complexity in the EST Routine:
  * Sturm $>$ Descartes $>$ Evaluation
  * Cheaper EST allows more adaptive complexity
  * This potential is not currently realized
- Open Problems:
  * Prove optimality of Sturm/Descartes method when $L = o(\log d)$
  * Develop practical techniques in Evaluation Based Isolation
  * Improved complexity analysis of Evaluation Based Isolation
  * Extend analysis to higher dimensional analogues
- Finally, a perspective for Evaluation Based Isolation:
  * Sturm and Descartes methods are now quite mature areas
  * But it took 3 decades of progress...
  * Cf. the optimal Sturm algorithm we presented

Summary and Open Problems

- Our results are based on:
  - Amortization arguments in algebraic complexity analysis
  - Effective use of Davenport-Mahler type bounds

- Decreasing complexity in the EST Routine:
  - Sturm $>$ Descartes $>$ Evaluation
  - Cheaper EST allows more adaptive complexity
  - This potential is not currently realized

- Open Problems:
  - Prove optimality of Sturm/Descartes method when $L = o(\log d)$
  - Develop practical techniques in Evaluation Based Isolation
  - Improved complexity analysis of Evaluation Based Isolation
  - Extend analysis to higher dimensional analogues

- Finally, a perspective for Evaluation Based Isolation:
  - Sturm and Descartes methods are now quite mature areas
  - But it took 3 decades of progress...
  - Cf. the optimal Sturm algorithm we presented

END of TALK

Thanks for Listening!

- Papers download from
  http://cs.nyu.edu/exact/
- "Amortized Bound for Root Isolation via Sturm Sequences"
  * Z.Du, V.Sharma, C.Yap.
    Workshop on Symbolic-Numeric Computation (SNC 2006)
- "Almost Tight Recursion Tree Bounds for the Descartes Method"
  * A.Eigenwilig, V.Sharma, C.Yap.
    (to appear) ISSAC 2006
- "Exact Evaluation-Based Root Isolation"
  * M.Burr, C.Wu, V.Sharma, C.Yap.
    Preprint, 2006

Adaptive Version

- Algorithm

Adaptive Version

- Algorithm

AdaptiveRootIsol Input: $f : \mathbb{R} \to \mathbb{R}$, and interval $I_0 = [r, s]$
all the zeros of f in I_0. Each isolating interval is classified as singular or simple.

1. if The lower endpoint, r, of I_0 is a root of f
    - if r is also a root of f', put $[r, r]$ into L (singular).
    - else put $[r, r]$ into L (nonsingular).
    - change the lower endpoint of I_0 to $r + \Delta(f)$.

   if The upper endpoint, s, of I_0 is a root of f
    - if s is also a root of f', put $[s, s]$ into L (singular).
    - else put $[s, s]$ into L (nonsingular).
    - change the upper endpoint of I_0 to $s - \Delta(f)$.

   Initialize Q to I_0.
   while $Q \neq \emptyset$
    - $I = [a, b] \leftarrow Q.remove()$.

2. if $0 \notin \square f(I)$, discard I
    - else if $|I| > min\{\Delta(f), \Delta(f'), \Delta(f'')\}$
        - if the midpoint of I, $m = (a + b)/2$, is root of f
            - if the midpoint of I is also a root of f', put $[m, m]$ into L (singular).
            - else put $[m, m]$ into L (nonsingular)

3. put $[a, m - \Delta(f)]$ and $[m + \Delta(f), b]$ into Q.
    - else Split I in two equal halves, and put them in Q.
    - else

4. if $f'(a)f'(b) \leq 0$

4.1 if $f(a)f(b) < 0$, put I in L (nonsingular).

4.2 else

4.3 if $f(a)f'(a) > 0$, discard I.

4.4 while $\min\{|f(a)|, |f(b)|\} \geq EV(f, f')$ and $\max\{|f(a)|, |f(b)|\} \leq K|b - a|$
- if the midpoint of I, $m = (a + b)/2$, is a root of f, put $[m, m]$ into L (singular)
    - break to main while loop.

4.5 else if $f'(a)f'(b) \leq 0$, set $I = [a, m]$
- else set $I = [m, b]$.

4.6 if $\max\{|f(a)|, |f(b)|\} > K|b - a|$, discard I.

4.7 else put I into L (singular).

5. else

5.1 if $f(a)f(b) > 0$, discard I.

5.2 else