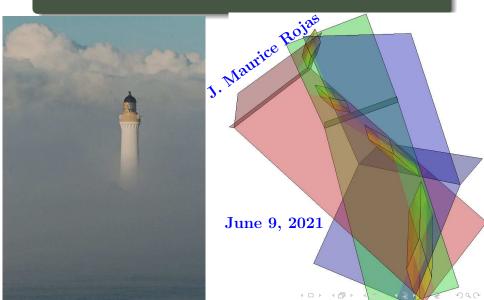
Counting Real Roots in Polynomial-Time for Systems Supported on Circuits



Motivation & Background



Counting Real Roots in Polynomial-Time

イロト イボト イヨト イヨト

Motivation & Background

Pewnomials and Number Theory



Counting Real Roots in Polynomial-Time

ヨト・イヨト

Motivation & Background

Pewnomials and Number Theory

③ Faster Real Root Counting for Circuit Systems



→ Ξ →

Counting Real Roots in Polynomial-Time

We'd like to solve...

...systems like:

 $\begin{pmatrix} 2x_1^3x_2^{124}x_3^50x_4^{82}x_5^{60} + x_1^{76}x_2^{240}x_4^{41}x_5 + x_1^{74}x_1^{179}x_3^5x_5^{57} + x_1^{25}x_2^{203}x_3^{44}x_4 + x_1^{20}x_2^{167}x_3^{64}x_1^{12}x_6^{88} - 37137x_1^{58}x_2^{194}x_3^{24}x_3^{68}x_5^{57} - \frac{9}{2}x_3^{166}x_4^{68}x_5^{343}, \\ x_1^{36}x_1^{34}x_2^{144}x_3^{50}x_4^{82}x_5^{60} + x_1^{76}x_2^{240}x_4^{41}x_5 + x_1^{74}x_1^{179}x_3^{55}x_5^{57} + x_1^{25}x_2^{203}x_3^{44}x_4 + x_1^{20}x_1^{267}x_3^{64}x_1^{22}x_5^{86} - 37137x_1^{58}x_2^{194}x_3^{24}x_3^{68}x_5^{55} - \frac{9}{2}x_3^{166}x_4^{68}x_5^{343}, \\ x_1^{36}x_1^{294}x_3^{50}x_4^{82}x_5^{60} + x_1^{76}x_2^{240}x_4^{41}x_5 + x_1^{74}x_1^{179}x_3^{55}x_5^{57} + x_1^{25}x_2^{203}x_3^{44}x_4 + x_1^{20}x_1^{267}x_3^{44}x_1^{22}x_5^{86} - 21009x_1^{58}x_2^{194}x_3^{24}x_4^{6}x_5^{55} - \frac{21}{4}x_3^{166}x_4^{68}x_5^{343}, \\ x_1^{36}x_1^{294}x_3^{50}x_4^{82}x_5^{60} + x_1^{76}x_2^{240}x_4^{41}x_5 + x_1^{74}x_1^{179}x_3^{25}x_5^{57} + x_1^{25}x_2^{203}x_3^{44}x_4 + x_1^{20}x_1^{267}x_3^{44}x_1^{22}x_5^{86} - 20109x_1^{58}x_2^{194}x_3^{24}x_4^{6}x_5^{25} - \frac{21}{4}x_3^{166}x_4^{68}x_5^{343}, \\ x_1^{36}x_1^{294}x_3^{50}x_4^{82}x_5^{60} + x_1^{76}x_2^{240}x_4^{41}x_5 + x_1^{74}x_1^{179}x_3^{25}x_5^{57} + x_1^{25}x_2^{203}x_3^{44}x_4 + x_1^{20}x_1^{267}x_3^{44}x_1^{22}x_5^{86} - 20769x_1^{58}x_2^{194}x_3^{24}x_4^{68}x_5^{55} - \frac{21}{4}x_3^{166}x_4^{68}x_5^{343}, \\ x_1^{36}x_2^{194}x_3^{50}x_4^{82}x_5^{60} + x_1^{76}x_2^{240}x_4^{41}x_5 + x_1^{74}x_1^{179}x_3^{25}x_5^{57} + x_1^{25}x_2^{203}x_3^{44}x_4 + 2x_1^{20}x_1^{267}x_3^{44}x_1^{2}x_5^{86} - 20769x_1^{58}x_2^{194}x_3^{24}x_4^{68}x_5^{55} - \frac{21}{4}x_3^{166}x_4^{68}x_5^{343}, \\ x_1^{36}x_2^{194}x_3^{60}x_4^{82}x_5^{60} + x_1^{76}x_2^{240}x_4^{41}x_5 + x_1^{74}x_2^{179}x_3^{25}x_5^{57} + x_1^{25}x_2^{203}x_3^{44}x_4 + 2x_1^{20}x_1^{267}x_3^{64}x_1^{22}x_5^{86} - 20754x_1^{58}x_2^{194}x_3^{24}x_4^{68}x_5^{5} - \frac{21}{4}x_3^{166}x_4^{68}x_5^{343} \end{pmatrix} \\ \mathbf{quickly}.$



We'd like to solve...

...systems like:

 $\begin{pmatrix} 2x_1^3 x_2^{124} x_3^5 x_4^{82} x_5^{60} + x_1^{76} x_2^{240} x_4^{41} x_5 + x_1^{74} x_1^{79} x_3^{25} x_5^{57} + x_1^{25} x_2^{203} x_4^{34} x_4 + x_1^{20} x_2^{167} x_3^{64} x_1^{42} x_5^{66} - 37137x_1^{58} x_1^{124} x_3^{24} x_4^{28} x_5^{55} - \frac{9}{2} x_1^{36} x_4^{68} x_5^{343}, \\ x_1^{36} x_1^{34} x_2^{144} x_3^{56} x_4^{82} x_5^{66} + x_1^{76} x_2^{240} x_4^{41} x_5 + x_1^{74} x_1^{179} x_3^{25} x_5^{57} + x_1^{25} x_2^{203} x_3^{44} x_4 + x_1^{20} x_1^{267} x_3^{64} x_1^{12} x_5^{66} - 24849 x_1^{58} x_1^{124} x_3^{24} x_4^{26} x_5^{55} - \frac{21}{4} x_3^{166} x_4^{68} x_5^{343}, \\ x_1^{36} x_1^{34} x_3^{14} x_3^{56} x_4^{82} x_5^{66} + x_1^{76} x_2^{240} x_4^{41} x_5 + 2x_1^{74} x_1^{179} x_3^{25} x_5^{57} + x_1^{25} x_2^{203} x_3^{44} x_4 + x_1^{20} x_1^{267} x_3^{64} x_1^{12} x_5^{66} - 21009 x_1^{58} x_1^{124} x_3^{24} x_4^{26} x_5^{55} - \frac{21}{4} x_3^{166} x_4^{68} x_5^{343}, \\ x_1^{36} x_1^{34} x_3^{14} x_3^{56} x_4^{28} x_5^{66} + x_1^{76} x_2^{240} x_4^{41} x_5 + 2x_1^{74} x_1^{179} x_3^{25} x_5^{57} + x_1^{25} x_2^{203} x_3^{44} x_4 + x_1^{20} x_1^{267} x_3^{44} x_1^{22} x_5^{66} - 21009 x_1^{58} x_1^{124} x_3^{24} x_4^{26} x_5^{55} - \frac{21}{4} x_3^{166} x_4^{68} x_5^{343}, \\ x_1^{36} x_1^{24} x_3^{50} x_4^{28} x_5^{66} + x_1^{76} x_2^{240} x_4^{41} x_5 + x_1^{74} x_1^{279} x_3^{25} x_5^{57} + 2x_1^{25} x_2^{203} x_3^{44} x_4 + x_1^{20} x_1^{267} x_3^{44} x_1^{22} x_5^{68} - 20769 x_1^{58} x_1^{124} x_4^{24} x_6^{55} - \frac{21}{4} x_3^{166} x_4^{68} x_5^{343}, \\ x_1^{36} x_1^{24} x_3^{50} x_4^{25} x_5^{66} + x_1^{76} x_2^{240} x_4^{41} x_5 + x_1^{74} x_1^{279} x_3^{25} x_5^{57} + x_1^{25} x_2^{203} x_3^{44} x_4 + 2x_1^{20} x_1^{267} x_3^{44} x_4^{2} x_5^{68} - 20769 x_1^{58} x_1^{124} x_4^{24} x_6^{55} - \frac{21}{4} x_3^{166} x_4^{68} x_5^{343}, \\ x_1^{36} x_1^{24} x_3^{50} x_4^{56} x_5^{56} + x_1^{76} x_2^{240} x_4^{41} x_5 + x_1^{74} x_1^{279} x_3^{25} x_5^{57} + x_1^{25} x_2^{203} x_3^{44} x_4 + 2x_1^{20} x_1^{267} x_3^{64} x_4^{12} x_5^{68} - 20759 x_1^{58} x_1^{294} x_4^{24} x_6^{56} x_5^{5}$

But how?



We'd like to solve...

...systems like:

 $\begin{pmatrix} 2x_1^3 x_2^{124} x_3^5 x_4^{82} x_5^{60} + x_1^{76} x_2^{240} x_4^{41} x_5 + x_1^{74} x_1^{79} x_3^{25} x_5^{57} + x_1^{25} x_2^{203} x_4^{34} x_4 + x_1^{20} x_2^{167} x_3^{64} x_4^{12} x_5^{68} - 37137x_1^{58} x_1^{124} x_3^{24} x_4^{28} x_5^{55} - \frac{9}{2} x_3^{166} x_4^{68} x_5^{343}, \\ x_1^{36} x_1^{34} x_2^{144} x_3^{56} x_4^{22} x_5^{60} + x_1^{76} x_2^{240} x_4^{41} x_5 + x_1^{74} x_1^{79} x_3^{25} x_5^{57} + x_1^{25} x_2^{203} x_3^{44} x_4 + x_1^{20} x_1^{267} x_3^{64} x_4^{12} x_5^{68} - 24849 x_1^{58} x_1^{124} x_3^{24} x_4^{26} x_5^{55} - \frac{21}{4} x_3^{166} x_4^{68} x_5^{343}, \\ x_1^{36} x_1^{34} x_3^{14} x_3^{50} x_4^{22} x_5^{60} + x_1^{76} x_2^{240} x_4^{41} x_5 + 2x_1^{74} x_1^{79} x_3^{25} x_5^{57} + x_1^{25} x_2^{203} x_3^{44} x_4 + x_1^{20} x_1^{267} x_3^{64} x_4^{12} x_5^{68} - 21009 x_1^{58} x_2^{124} x_3^{24} x_4^{26} x_5^{55} - \frac{21}{4} x_3^{166} x_4^{68} x_5^{343}, \\ x_1^{36} x_1^{34} x_3^{14} x_3^{50} x_4^{22} x_5^{60} + x_1^{76} x_2^{240} x_4^{41} x_5 + 2x_1^{74} x_1^{79} x_3^{25} x_5^{57} + x_1^{25} x_2^{203} x_3^{44} x_4 + x_1^{20} x_1^{267} x_3^{64} x_4^{12} x_5^{68} - 21009 x_1^{58} x_1^{124} x_3^{24} x_4^{26} x_5^{55} - \frac{21}{4} x_3^{166} x_4^{68} x_5^{343}, \\ x_1^{36} x_1^{24} x_3^{50} x_4^{25} x_5^{20} + x_1^{76} x_2^{240} x_4^{41} x_5 + x_1^{74} x_1^{79} x_3^{25} x_5^{57} + 2x_1^{25} x_2^{203} x_3^{44} x_4 + x_1^{20} x_1^{267} x_3^{44} x_4^{22} x_5^{68} - 20769 x_1^{18} x_1^{124} x_4^{26} x_5^{55} - \frac{21}{4} x_3^{166} x_4^{68} x_5^{343}, \\ x_1^{36} x_1^{24} x_3^{50} x_4^{25} x_5^{60} + x_1^{76} x_2^{240} x_4^{41} x_5 + x_1^{74} x_1^{79} x_3^{25} x_5^{57} + x_1^{25} x_2^{203} x_3^{44} x_4 + 2x_1^{20} x_1^{267} x_3^{44} x_4^{27} x_5^{68} - 20769 x_1^{18} x_1^{24} x_4^{26} x_5^{55} - \frac{21}{4} x_3^{166} x_4^{68} x_5^{343}, \\ x_1^{36} x_1^{24} x_3^{50} x_4^{25} x_5^{60} + x_1^{76} x_4^{24} x_5^{56} + x_1^{76} x_4^{21} x_5^{56} + x_1^{76} x_4^{21} x_5^{56} + x_1^{76} x_4^{21} x_5^{26} - 20754 x_1^{58} x_1^{29} x_3^{24} x_4^{26} x_5^{5} - \frac{21}{4} x_1^{66} x_6^{68} x_5^{343}, \\ x_1$

But how?

More importantly, why?



Polynomials Equations and Complexity Theory

• Complexity theory is the friend of cryptography,



Counting Real Roots in Polynomial-Time

Polynomials Equations and Complexity Theory

• Complexity theory is the friend of cryptography, as well as the difficult cousin of learning theory



Polynomials Equations and Complexity Theory

• Complexity theory is the friend of cryptography, as well as the difficult cousin of learning theory and numerical computation.



- Complexity theory is the friend of cryptography, as well as the difficult cousin of learning theory and numerical computation.
- Polynomials are really just circuits



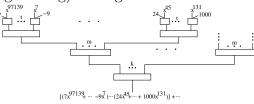
- Complexity theory is the friend of cryptography, as well as the difficult cousin of learning theory and numerical computation.
- Polynomials are really just circuits (in the sense of electrical engineering)



- Complexity theory is the friend of cryptography, as well as the difficult cousin of learning theory and numerical computation.
- Polynomials are really just circuits (in the sense of electrical engineering) in disguise...

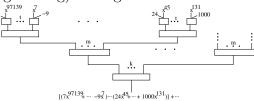


- Complexity theory is the friend of cryptography, as well as the difficult cousin of learning theory and numerical computation.
- Polynomials are really just circuits (in the sense of electrical engineering) in disguise...





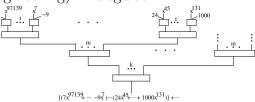
- Complexity theory is the friend of cryptography, as well as the difficult cousin of learning theory and numerical computation.
- Polynomials are really just circuits (in the sense of electrical engineering) in disguise...



Fundamental Idea: Theorems about existence of roots are close to the \mathbf{P} vs. \mathbf{NP} Problem.



- Complexity theory is the friend of cryptography, as well as the difficult cousin of learning theory and numerical computation.
- Polynomials are really just circuits (in the sense of electrical engineering) in disguise...

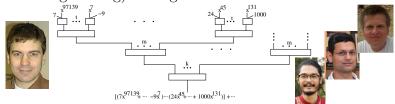


Fundamental Idea: Theorems about existence of roots are close to the **P** vs. **NP** Problem. Theorems about the deeper structure of polynomials are close to derandomization,



ロト ・ 同ト ・ ヨト ・ ヨト

- Complexity theory is the friend of cryptography, as well as the difficult cousin of learning theory and numerical computation.
- Polynomials are really just circuits (in the sense of electrical engineering) in disguise...



Fundamental Idea: Theorems about existence of roots are close to the **P** vs. **NP** Problem. Theorems about the deeper structure of polynomials are close to derandomization, i.e., the **P** vs. **BPP** Problem [Koiran, '11; Dutta, Saxena, Thierauf, '



Descartes' Rule, Biochemistry, and Learning Theory...



Recent work on chemical reaction



Counting Real Roots in Polynomial-Time

< ロト < 同ト < ヨト < ヨ

Descartes' Rule, Biochemistry, and Learning Theory...



Recent work on chemical reaction networks makes serious use of <u>exact</u> counting of real roots

for sparse systems [Bihan, Dickenstein, Giaroli, Shiu, 2019–2020].



Descartes' Rule, Biochemistry, and Learning Theory...



Recent work on chemical reaction networks makes serious use of <u>exact</u> counting of real roots

for sparse systems [Bihan, Dickenstein, Giaroli, Shiu, 2019–2020].

 $\bullet[{\rm Ren},\,{\rm Zhang},\,2020]$ applies tropical few nomials to neural networks...



Descartes' Rule, Biochemistry, and Learning Theory...



Recent work on chemical reaction networks makes serious use of <u>exact</u> counting of real roots

for sparse systems [Bihan, Dickenstein, Giaroli, Shiu, 2019–2020].

 $\bullet[\operatorname{Ren},\,\operatorname{Zhang},\,2020]$ applies tropical few nomials to neural networks...



[Bihan, Dickenstein, Forsgård, 2020] Multivariate version of Descartes' Rule for (n + 2)-nomial $n \times n$ systems...



(4) (日本)

Descartes' Rule, Biochemistry, and Learning Theory...



Recent work on chemical reaction networks makes serious use of <u>exact</u> counting of real roots

for sparse systems [Bihan, Dickenstein, Giaroli, Shiu, 2019–2020].

 $\bullet[\operatorname{Ren},\,\operatorname{Zhang},\,2020]$ applies tropical few nomials to neural networks...



[Bihan, Dickenstein, Forsgård, 2020] Multivariate version of Descartes' Rule for (n + 2)-nomial $n \times n$ systems...

But what about *exact counting*?



< ロト < 同ト < ヨト < ヨト

Detecting Real Roots is Already Hard for $\underline{1}$ Sparse (Multivariate) Polynomial!

Complexity Lower Bound. [Bihan, Rojas, Stella, 2009] Fix any $\varepsilon > 0$



Counting Real Roots in Polynomial-Time

Detecting Real Roots is Already Hard for $\underline{1}$ Sparse (Multivariate) Polynomial!

Complexity Lower Bound.

[Bihan, Rojas, Stella, 2009] Fix any $\varepsilon > 0$ and suppose there is an algorithm that, for any polynomial $f \in \mathbb{Q}[x_1, \ldots, x_n]$ with at most $n + n^{\varepsilon}$ terms,



Counting Real Roots in Polynomial-Time

Detecting Real Roots is Already Hard for $\underline{1}$ Sparse (Multivariate) Polynomial!

Complexity Lower Bound.

[Bihan, Rojas, Stella, 2009] Fix any $\varepsilon > 0$ and suppose there is an algorithm that, for any polynomial $f \in \mathbb{Q}[x_1, \ldots, x_n]$ with at most $n + n^{\varepsilon}$ terms, decides if f has a real root,



Detecting Real Roots is Already Hard for $\underline{1}$ Sparse (Multivariate) Polynomial!

Complexity Lower Bound.

[Bihan, Rojas, Stella, 2009] Fix any $\varepsilon > 0$ and suppose there is an algorithm that, for any polynomial $f \in \mathbb{Q}[x_1, \ldots, x_n]$ with at most $n + n^{\varepsilon}$ terms, decides if f has a real root, using time polynomial in the bit-size of f.



Detecting Real Roots is Already Hard for $\underline{1}$ Sparse (Multivariate) Polynomial!

Complexity Lower Bound.

[Bihan, Rojas, Stella, 2009] Fix any $\varepsilon > 0$ and suppose there is an algorithm that, for any polynomial $f \in \mathbb{Q}[x_1, \ldots, x_n]$ with at most $n + n^{\varepsilon}$ terms, decides if f has a real root, using time polynomial in the bit-size of f. Then $\mathbf{P} = \mathbf{NP}$.



Detecting Real Roots is Already Hard for $\underline{1}$ Sparse (Multivariate) Polynomial!

Complexity Lower Bound.

[Bihan, Rojas, Stella, 2009] Fix any $\varepsilon > 0$ and suppose there is an algorithm that, for any polynomial $f \in \mathbb{Q}[x_1, \ldots, x_n]$ with at most $n + n^{\varepsilon}$ terms, decides if f has a real root, using time polynomial in the bit-size of f. Then $\mathbf{P} = \mathbf{NP}$.

• We use the Turing (bit) model, and



Detecting Real Roots is Already Hard for $\underline{1}$ Sparse (Multivariate) Polynomial!

Complexity Lower Bound.

[Bihan, Rojas, Stella, 2009] Fix any $\varepsilon > 0$ and suppose there is an algorithm that, for any polynomial $f \in \mathbb{Q}[x_1, \ldots, x_n]$ with at most $n + n^{\varepsilon}$ terms, decides if f has a real root, using time polynomial in the bit-size of f. Then $\mathbf{P} = \mathbf{NP}$.

• We use the Turing (bit) model, and size(f):=Total number of bits in coefficients



Detecting Real Roots is Already Hard for $\underline{1}$ Sparse (Multivariate) Polynomial!

Complexity Lower Bound.

[Bihan, Rojas, Stella, 2009] Fix any $\varepsilon > 0$ and suppose there is an algorithm that, for any polynomial $f \in \mathbb{Q}[x_1, \ldots, x_n]$ with at most $n + n^{\varepsilon}$ terms, decides if f has a real root, using time polynomial in the bit-size of f. Then $\mathbf{P} = \mathbf{NP}$.

• We use the Turing (bit) model, and size(f):= Total number of bits in coefficients and the exponents of f.



< ロト < 同ト < ヨト < ヨト

Detecting Real Roots is Already Hard for $\underline{1}$ Sparse (Multivariate) Polynomial!

Complexity Lower Bound.

[Bihan, Rojas, Stella, 2009] Fix any $\varepsilon > 0$ and suppose there is an algorithm that, for any polynomial $f \in \mathbb{Q}[x_1, \ldots, x_n]$ with at most $n + n^{\varepsilon}$ terms, decides if f has a real root, using time polynomial in the bit-size of f. Then $\mathbf{P} = \mathbf{NP}$.

- We use the Turing (bit) model, and size(f):= Total number of bits in coefficients and the exponents of f.
- The case of n + 1 terms is a cute exercise for your students:



< ロト < 同ト < ヨト < ヨト

Complexity Lower Bound.

[Bihan, Rojas, Stella, 2009] Fix any $\varepsilon > 0$ and suppose there is an algorithm that, for any polynomial $f \in \mathbb{Q}[x_1, \ldots, x_n]$ with at most $n + n^{\varepsilon}$ terms, decides if f has a real root, using time polynomial in the bit-size of f. Then $\mathbf{P} = \mathbf{NP}$.

- We use the Turing (bit) model, and size(f):= Total number of bits in coefficients and the exponents of f.
- The case of n + 1 terms is a cute exercise for your students: An O(n)-time algorithm



Complexity Lower Bound.

[Bihan, Rojas, Stella, 2009] Fix any $\varepsilon > 0$ and suppose there is an algorithm that, for any polynomial $f \in \mathbb{Q}[x_1, \ldots, x_n]$ with at most $n + n^{\varepsilon}$ terms, decides if f has a real root, using time polynomial in the bit-size of f. Then $\mathbf{P} = \mathbf{NP}$.

- We use the Turing (bit) model, and size(f):=Total number of bits in coefficients and the exponents of f.
- The case of n + 1 terms is a cute exercise for your students: An O(n)-time algorithm (assuming affinely independent exponents)



Complexity Lower Bound.

[Bihan, Rojas, Stella, 2009] Fix any $\varepsilon > 0$ and suppose there is an algorithm that, for any polynomial $f \in \mathbb{Q}[x_1, \ldots, x_n]$ with at most $n + n^{\varepsilon}$ terms, decides if f has a real root, using time polynomial in the bit-size of f. Then $\mathbf{P} = \mathbf{NP}$.

- We use the Turing (bit) model, and size(f):= Total number of bits in coefficients and the exponents of f.
- The case of n + 1 terms is a cute exercise for your students: An O(n)-time algorithm (assuming affinely independent exponents) is possible!



Complexity Lower Bound.

[Bihan, Rojas, Stella, 2009] Fix any $\varepsilon > 0$ and suppose there is an algorithm that, for any polynomial $f \in \mathbb{Q}[x_1, \ldots, x_n]$ with at most $n + n^{\varepsilon}$ terms, decides if f has a real root, using time polynomial in the bit-size of f. Then $\mathbf{P} = \mathbf{NP}$.

- We use the Turing (bit) model, and size(f):= Total number of bits in coefficients and the exponents of f.
- The case of n + 1 terms is a cute exercise for your students: An O(n)-time algorithm (assuming affinely independent exponents) is possible!
- But what about systems?

Counting Roots for (n + 1)-nomial $n \times n$ Systems is Easy, but...

Bonus Exercise. Given $[c_{i,j}] \in \{-H, \ldots, H\}^{n \times (n+1)}$,



Counting Roots for (n + 1)-nomial $n \times n$ Systems is Easy, but...

Bonus Exercise. Given $[c_{i,j}] \in \{-H, \ldots, H\}^{n \times (n+1)}$, and affinely independent $a_1, \ldots, a_{n+1} \in \{-d, \ldots, d\}^n$,



Counting Roots for (n + 1)-nomial $n \times n$ Systems is Easy, but...

Bonus Exercise. Given $[c_{i,j}] \in \{-H, \ldots, H\}^{n \times (n+1)}$, and affinely independent $a_1, \ldots, a_{n+1} \in \{-d, \ldots, d\}^n$, count the roots of



Counting Roots for (n + 1)-nomial $n \times n$ Systems is Easy, but...

Bonus Exercise. Given $[c_{i,j}] \in \{-H, \ldots, H\}^{n \times (n+1)}$, and affinely independent $a_1, \ldots, a_{n+1} \in \{-d, \ldots, d\}^n$, count the roots of

$$c_{1,1}x^{a_1} + \dots + c_{1,n+1}x^{a_{n+1}} = 0$$

$$\vdots$$

$$c_{n,1}x^{a_1} + \dots + c_{n,n+1}x^{a_{n+1}} = 0$$



Counting Real Roots in Polynomial-Time

Counting Roots for (n + 1)-nomial $n \times n$ Systems is Easy, but...

Bonus Exercise. Given $[c_{i,j}] \in \{-H, \ldots, H\}^{n \times (n+1)}$, and affinely independent $a_1, \ldots, a_{n+1} \in \{-d, \ldots, d\}^n$, count the roots of

$$c_{1,1}x^{a_1} + \dots + c_{1,n+1}x^{a_{n+1}} = 0$$

$$\vdots$$

$$c_{n,1}x^{a_1} + \dots + c_{n,n+1}x^{a_{n+1}} = 0$$

in \mathbb{R}^n_+ ,



Counting Roots for (n + 1)-nomial $n \times n$ Systems is Easy, but...

Bonus Exercise. Given $[c_{i,j}] \in \{-H, \ldots, H\}^{n \times (n+1)}$, and affinely independent $a_1, \ldots, a_{n+1} \in \{-d, \ldots, d\}^n$, count the roots of

$$c_{1,1}x^{a_1} + \dots + c_{1,n+1}x^{a_{n+1}} = 0$$

$$\vdots$$

$$c_{n,1}x^{a_1} + \dots + c_{n,n+1}x^{a_{n+1}} = 0$$

in \mathbb{R}^n_+ , $(\mathbb{R}^*)^n$,



Counting Roots for (n + 1)-nomial $n \times n$ Systems is Easy, but...

Bonus Exercise. Given $[c_{i,j}] \in \{-H, \ldots, H\}^{n \times (n+1)}$, and affinely independent $a_1, \ldots, a_{n+1} \in \{-d, \ldots, d\}^n$, count the roots of

$$c_{1,1}x^{a_1} + \dots + c_{1,n+1}x^{a_{n+1}} = 0$$

$$\vdots$$

$$c_{n,1}x^{a_1} + \dots + c_{n,n+1}x^{a_{n+1}} = 0$$

in \mathbb{R}^n_+ , $(\mathbb{R}^*)^n$, and \mathbb{R}^n in time polynomial in $n + \log(dH)$...



Counting Roots for (n + 1)-nomial $n \times n$ Systems is Easy, but...

Bonus Exercise. Given $[c_{i,j}] \in \{-H, \ldots, H\}^{n \times (n+1)}$, and affinely independent $a_1, \ldots, a_{n+1} \in \{-d, \ldots, d\}^n$, count the roots of

$$c_{1,1}x^{a_1} + \dots + c_{1,n+1}x^{a_{n+1}} = 0$$

$$\vdots$$

$$c_{n,1}x^{a_1} + \dots + c_{n,n+1}x^{a_{n+1}} = 0$$

in \mathbb{R}^n_+ , $(\mathbb{R}^*)^n$, and \mathbb{R}^n in time polynomial in $n + \log(dH)$...

• Smith Factorization helps,



Counting Roots for (n + 1)-nomial $n \times n$ Systems is Easy, but...

Bonus Exercise. Given $[c_{i,j}] \in \{-H, \ldots, H\}^{n \times (n+1)}$, and affinely independent $a_1, \ldots, a_{n+1} \in \{-d, \ldots, d\}^n$, count the roots of

$$\begin{vmatrix} c_{1,1}x^{a_1} + \dots + c_{1,n+1}x^{a_{n+1}} = 0 \\ \vdots \\ c_{n,1}x^{a_1} + \dots + c_{n,n+1}x^{a_{n+1}} = 0 \end{vmatrix}$$

in \mathbb{R}^n_+ , $(\mathbb{R}^*)^n$, and \mathbb{R}^n in time polynomial in $n + \log(dH)$...

• Smith Factorization helps, and you'll ultimately need (polynomial-time!) Linear Programming for the case ∞ !



Counting Roots for (n + 1)-nomial $n \times n$ Systems is Easy, but...

Bonus Exercise. Given $[c_{i,j}] \in \{-H, \ldots, H\}^{n \times (n+1)}$, and affinely independent $a_1, \ldots, a_{n+1} \in \{-d, \ldots, d\}^n$, count the roots of

$$\begin{vmatrix} c_{1,1}x^{a_1} + \dots + c_{1,n+1}x^{a_{n+1}} = 0 \\ \vdots \\ c_{n,1}x^{a_1} + \dots + c_{n,n+1}x^{a_{n+1}} = 0 \end{vmatrix}$$

in \mathbb{R}^n_+ , $(\mathbb{R}^*)^n$, and \mathbb{R}^n in time polynomial in $n + \log(dH)$...

- Smith Factorization helps, and you'll ultimately need (polynomial-time!) Linear Programming for the case ∞ !
- (n+2)-nomial $n \times n$ systems are the next step up,



Counting Roots for (n + 1)-nomial $n \times n$ Systems is Easy, but...

Bonus Exercise. Given $[c_{i,j}] \in \{-H, \ldots, H\}^{n \times (n+1)}$, and affinely independent $a_1, \ldots, a_{n+1} \in \{-d, \ldots, d\}^n$, count the roots of

$$\begin{vmatrix} c_{1,1}x^{a_1} + \dots + c_{1,n+1}x^{a_{n+1}} = 0 \\ \vdots \\ c_{n,1}x^{a_1} + \dots + c_{n,n+1}x^{a_{n+1}} = 0 \end{vmatrix}$$

in \mathbb{R}^n_+ , $(\mathbb{R}^*)^n$, and \mathbb{R}^n in time polynomial in $n + \log(dH)$...

• Smith Factorization helps, and you'll ultimately need (polynomial-time!) Linear Programming for the case ∞ !

• (n+2)-nomial $n \times n$ systems are the next step up, and the last tractable case



Counting Roots for (n + 1)-nomial $n \times n$ Systems is Easy, but...

Bonus Exercise. Given $[c_{i,j}] \in \{-H, \ldots, H\}^{n \times (n+1)}$, and affinely independent $a_1, \ldots, a_{n+1} \in \{-d, \ldots, d\}^n$, count the roots of

$$\begin{vmatrix} c_{1,1}x^{a_1} + \dots + c_{1,n+1}x^{a_{n+1}} = 0 \\ \vdots \\ c_{n,1}x^{a_1} + \dots + c_{n,n+1}x^{a_{n+1}} = 0 \end{vmatrix}$$

in \mathbb{R}^n_+ , $(\mathbb{R}^*)^n$, and \mathbb{R}^n in time polynomial in $n + \log(dH)$...

• Smith Factorization helps, and you'll ultimately need (polynomial-time!) Linear Programming for the case ∞ !

• (n+2)-nomial $n \times n$ systems are the next step up, and the last tractable case (for now)...



Main Theorem (coarsely)

[Rojas, 2020] For any fixed n,



 < □ > < □ > < □ > < □ > < □ > < □ > <</td>

 Counting Real Roots in Polynomial-Time

Main Theorem (coarsely)

[Rojas, 2020] For any fixed n, one can exactly count the roots in \mathbb{R}^n_+ ,



Counting Real Roots in Polynomial-Time

Main Theorem (coarsely)

[Rojas, 2020] For any fixed n, one can exactly count the roots in \mathbb{R}^n_+ , $(\mathbb{R}^*)^n$,



Counting Real Roots in Polynomial-Time

Main Theorem (coarsely)

[Rojas, 2020] For any fixed n, one can exactly count the roots in \mathbb{R}^n_+ , $(\mathbb{R}^*)^n$, and \mathbb{R}^n



Main Theorem (coarsely)

[Rojas, 2020] For any fixed n, one can exactly count the roots in \mathbb{R}^n_+ , $(\mathbb{R}^*)^n$, and \mathbb{R}^n of any generic (n+2)-nomial $n \times n$ polynomial system



 <□ > < □ > < □ > < □ > < □ > < □ > < □ >

 Counting Real Roots in Polynomial-Time

Main Theorem (coarsely)

[Rojas, 2020] For any fixed n, one can exactly count the roots in \mathbb{R}^n_+ , $(\mathbb{R}^*)^n$, and \mathbb{R}^n of any generic (n+2)-nomial $n \times n$ polynomial system over \mathbb{Q}



Main Theorem (coarsely)

[Rojas, 2020] For any fixed n, one can exactly count the roots in \mathbb{R}^n_+ , $(\mathbb{R}^*)^n$, and \mathbb{R}^n of any generic (n+2)-nomial $n \times n$ polynomial system over \mathbb{Q} in deterministic polynomial-time.



イロト イボト イヨト イヨト

Main Theorem (coarsely)

[Rojas, 2020] For any fixed n, one can exactly count the roots in \mathbb{R}^n_+ , $(\mathbb{R}^*)^n$, and \mathbb{R}^n of any generic (n+2)-nomial $n \times n$ polynomial system over \mathbb{Q} in deterministic polynomial-time.

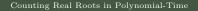
Example: In <1 second, we can see that the 7-nomial 5×5 system defined by



[Rojas, 2020] For any fixed n, one can exactly count the roots in \mathbb{R}^n_+ , $(\mathbb{R}^*)^n$, and \mathbb{R}^n of any generic (n+2)-nomial $n \times n$ polynomial system over \mathbb{Q} in deterministic polynomial-time.

Example: In <1 second, we can see that the 7-nomial 5×5 system defined by

 $\left(2x_{1}^{3}x_{2}^{194}x_{3}^{5}x_{4}^{22}x_{5}^{60} + x_{1}^{76}x_{2}^{240}x_{4}^{41}x_{5} + x_{1}^{74}x_{2}^{179}x_{3}^{25}x_{5}^{57} + x_{1}^{25}x_{2}^{203}x_{3}^{44}x_{4} + x_{1}^{20}x_{2}^{167}x_{3}^{4}x_{4}^{12}x_{5}^{68} - 37137cx_{1}^{58}x_{2}^{194}x_{3}^{4}x_{4}^{28}x_{5}^{55} - \frac{9}{2}x_{1}^{366}x_{4}^{68}x_{5}^{343}, \\ x_{1}^{36}x_{2}^{194}x_{3}^{5}x_{4}^{22}x_{5}^{60} + x_{1}^{76}x_{2}^{240}x_{4}^{41}x_{5} + x_{1}^{74}x_{2}^{179}x_{3}^{25}x_{5}^{57} + x_{1}^{25}x_{2}^{203}x_{3}^{44}x_{4} + x_{1}^{20}x_{2}^{167}x_{4}^{64}x_{4}^{28}x_{5}^{66} - 248490x_{1}^{58}x_{2}^{194}x_{4}^{24}x_{6}^{25}x_{5}^{-21}x_{4}^{14}x_{5}^{66}x_{6}^{68}x_{5}^{343}, \\ x_{1}^{36}x_{2}^{194}x_{3}^{5}x_{4}^{22}x_{5}^{60} + x_{1}^{76}x_{2}^{240}x_{4}^{41}x_{5} + x_{1}^{74}x_{1}^{179}x_{3}^{25}x_{5}^{57} + x_{1}^{25}x_{2}^{203}x_{3}^{44}x_{4} + x_{1}^{20}x_{2}^{167}x_{4}^{44}x_{5}^{26} - 21009cx_{1}^{58}x_{2}^{194}x_{4}^{34}x_{6}^{25} - \frac{21}{4}x_{3}^{166}x_{6}^{68}x_{5}^{343}, \\ x_{1}^{36}x_{1}^{194}x_{3}^{5}x_{4}^{22}x_{5}^{60} + x_{1}^{76}x_{2}^{240}x_{4}^{41}x_{5} + x_{1}^{74}x_{1}^{179}x_{3}^{25}x_{5}^{57} + 2x_{1}^{25}x_{2}^{203}x_{3}^{44}x_{4} + x_{1}^{20}x_{2}^{167}x_{3}^{44}x_{4}^{12}x_{5}^{68} - 20769cx_{1}^{58}x_{2}^{194}x_{4}^{34}x_{6}^{25} - \frac{21}{4}x_{3}^{166}x_{6}^{68}x_{5}^{343}, \\ x_{1}^{36}x_{2}^{194}x_{3}^{5}x_{4}^{22}x_{5}^{60} + x_{1}^{76}x_{2}^{240}x_{4}^{41}x_{5} + x_{1}^{74}x_{1}^{179}x_{3}^{25}x_{5}^{57} + 2x_{1}^{25}x_{2}^{203}x_{3}^{44}x_{4} + x_{1}^{20}x_{2}^{167}x_{3}^{44}x_{4}^{12}x_{5}^{68} - 20769cx_{1}^{58}x_{2}^{194}x_{4}^{34}x_{6}^{55} - \frac{21}{4}x_{3}^{66}x_{6}^{68}x_{5}^{343}, \\ x_{1}^{36}x_{2}^{49}x_{3}^{50}x_{4}^{25}x_{5}^{6} + x_{1}^{76}x_{2}^{240}x_{4}^{41}x_{5} + x_{1}^{74}x_{2}^{179}x_{3}^{25}x_{5}^{57} + 2x_{1}^{25}x_{2}^{203}x_{3}^{44}x_{4} + x_{1}^{20}x_{2}^{167}x_{3}^{4}x_{4}^{12}x_{5}^{68} - 20769cx_{1}^{58}x_{2}^{194}x_{4}^{34}x_{6}^{5}x_{5}^{5} - \frac{21}{4}x_{3}^{66}x_{6}^{68}x_{5}^{343}, \\ x_{1}^{36}x_{2}^{49}x_{3}^{40}x_{5}^{5}x_{5}^{5} + x_{1}^{75}x_{2}^{203}x_{3}^{4}x_{4} +$



- ロト - (母 ト - (ヨ ト -) ヨ

[Rojas, 2020] For any fixed n, one can exactly count the roots in \mathbb{R}^n_+ , $(\mathbb{R}^*)^n$, and \mathbb{R}^n of any generic (n+2)-nomial $n \times n$ polynomial system over \mathbb{Q} in deterministic polynomial-time.

Example: In <1 second, we can see that the 7-nomial 5×5 system defined by

 $\begin{array}{l} \left(2x_{1}^{3}x_{2}^{194}x_{3}^{5}x_{4}^{22}x_{5}^{60} + x_{1}^{76}x_{2}^{240}x_{4}^{41}x_{5} + x_{1}^{74}x_{2}^{179}x_{3}^{25}x_{5}^{57} + x_{1}^{25}x_{2}^{203}x_{4}^{34}x_{4} + x_{1}^{20}x_{2}^{167}x_{4}^{54}x_{4}^{12}x_{5}^{68} - 37137cx_{1}^{58}x_{1}^{94}x_{4}^{24}x_{4}^{25}x_{5}^{57} - \frac{9}{2}x_{1}^{366}x_{4}^{68}x_{5}^{33}, \\ x_{1}^{36}x_{1}^{194}x_{3}^{5}x_{4}^{22}x_{5}^{60} + x_{1}^{76}x_{2}^{240}x_{4}^{41}x_{5} + x_{1}^{74}x_{1}^{179}x_{3}^{25}x_{5}^{57} + x_{1}^{25}x_{2}^{203}x_{4}^{34}x_{4} + x_{1}^{20}x_{2}^{167}x_{4}^{64}x_{4}^{2}x_{5}^{66} - 24849cx_{1}^{58}x_{1}^{194}x_{4}^{34}x_{4}^{25}x_{5}^{5} - \frac{21}{4}x_{3}^{166}x_{6}^{68}x_{5}^{33}, \\ x_{1}^{36}x_{1}^{194}x_{3}^{50}x_{4}^{82}x_{5}^{60} + x_{1}^{76}x_{2}^{240}x_{4}^{41}x_{5} + 2x_{1}^{74}x_{1}^{179}x_{3}^{25}x_{5}^{57} + x_{1}^{25}x_{2}^{203}x_{4}^{34}x_{4} + x_{1}^{20}x_{2}^{167}x_{4}^{64}x_{4}^{2}x_{5}^{66} - 20109cx_{1}^{58}x_{1}^{194}x_{4}^{34}x_{6}^{25} - \frac{21}{4}x_{3}^{166}x_{6}^{68}x_{5}^{343}, \\ x_{1}^{36}x_{1}^{194}x_{3}^{50}x_{4}^{82}x_{5}^{60} + x_{1}^{76}x_{2}^{240}x_{4}^{41}x_{5} + x_{1}^{74}x_{1}^{129}x_{3}^{25}x_{5}^{57} + 2x_{1}^{25}x_{2}^{203}x_{3}^{44}x_{4} + x_{1}^{20}x_{2}^{167}x_{4}^{64}x_{4}^{2}x_{5}^{66} - 20109cx_{1}^{58}x_{1}^{194}x_{4}^{24}x_{6}^{25} - \frac{21}{4}x_{3}^{166}x_{6}^{68}x_{5}^{343}, \\ x_{1}^{36}x_{1}^{194}x_{3}^{50}x_{4}^{82}x_{5}^{60} + x_{1}^{76}x_{2}^{240}x_{4}^{41}x_{5} + x_{1}^{74}x_{1}^{129}x_{3}^{25}x_{5}^{57} + 2x_{1}^{25}x_{2}^{203}x_{3}^{44}x_{4} + x_{1}^{20}x_{2}^{167}x_{4}^{64}x_{4}^{2}x_{5}^{66} - 20769cx_{1}^{58}x_{1}^{194}x_{4}^{24}x_{6}^{36}x_{5}^{-2} - \frac{21}{4}x_{3}^{166}x_{6}^{68}x_{5}^{343}, \\ x_{1}^{36}x_{1}^{194}x_{3}^{50}x_{4}^{82}x_{5}^{60} + x_{1}^{76}x_{2}^{240}x_{4}^{41}x_{5} + x_{1}^{74}x_{2}^{179}x_{3}^{25}x_{5}^{57} + x_{1}^{25}x_{2}^{203}x_{3}^{44}x_{4} + 2x_{1}^{20}x_{1}^{167}x_{4}^{64}x_{4}^{12}x_{5}^{68} - 20769cx_{1}^{58}x_{1}^{194}x_{4}^{24}x_{6}^{5}x_{5}^{-2} - \frac{21}{4}x_{3}^{166}x_{6}^{68}x_{5}^{343}, \\ x_{1}^{36}x_{1}^{24}x_{3}x_{3}x_{5}^{60} + x_{1}^{76}x_{2}^{$

[Rojas, 2020] For any fixed n, one can exactly count the roots in \mathbb{R}^n_+ , $(\mathbb{R}^*)^n$, and \mathbb{R}^n of any generic (n+2)-nomial $n \times n$ polynomial system over \mathbb{Q} in deterministic polynomial-time.

Example: In <1 second, we can see that the 7-nomial 5×5 system defined by

 $\begin{pmatrix} (2x_1^{36}x_2^{194}x_3^{50}x_4^{22}x_5^{60} + x_1^{76}x_2^{240}x_1^{41}x_5 + x_1^{74}x_2^{179}x_3^{25}x_5^{57} + x_1^{25}x_2^{203}x_3^{44}x_4 + x_1^{20}x_2^{167}x_3^{44}x_1^{2}x_5^{86} - 37137cx_1^{58}x_2^{194}x_3^{24}x_4^{26}x_5^{25} - \frac{9}{2}x_1^{366}x_4^{68}x_3^{43}, \\ x_1^{36}x_2^{194}x_3^{50}x_4^{22}x_5^{60} + 2x_1^{76}x_2^{240}x_1^{41}x_5 + x_1^{74}x_2^{179}x_3^{25}x_5^{57} + x_1^{25}x_2^{203}x_3^{44}x_4 + x_1^{20}x_2^{167}x_3^{64}x_1^{2}x_5^{86} - 24849cx_1^{58}x_2^{194}x_3^{44}x_4^{56}x_5^{25} - \frac{21}{4}x_1^{366}x_6^{68}x_5^{343}, \\ x_1^{36}x_2^{194}x_3^{50}x_4^{22}x_5^{60} + x_1^{76}x_2^{240}x_4^{41}x_5 + x_1^{74}x_2^{179}x_3^{25}x_5^{57} + x_1^{25}x_2^{203}x_3^{44}x_4 + x_1^{20}x_2^{167}x_6^{44}x_1^{2}x_6^{86} - 21009cx_1^{58}x_2^{194}x_3^{44}x_6^{25} - \frac{21}{4}x_1^{366}x_6^{68}x_5^{343}, \\ x_1^{36}x_2^{194}x_3^{50}x_4^{22}x_5^{60} + x_1^{76}x_2^{240}x_4^{41}x_5 + x_1^{74}x_2^{179}x_3^{25}x_5^{57} + 2x_1^{25}x_2^{203}x_3^{44}x_4 + x_1^{20}x_2^{167}x_6^{44}x_1^{2}x_6^{86} - 20709cx_1^{58}x_2^{194}x_3^{44}x_6^{25} - \frac{21}{4}x_1^{366}x_6^{68}x_5^{343}, \\ x_1^{36}x_2^{194}x_3^{50}x_4^{22}x_5^{60} + x_1^{76}x_2^{240}x_4^{41}x_5 + x_1^{74}x_2^{179}x_3^{25}x_5^{57} + x_1^{25}x_2^{203}x_3^{44}x_4 + x_1^{20}x_2^{167}x_6^{44}x_1^{2}x_6^{86} - 20709cx_1^{58}x_1^{194}x_3^{44}x_6^{25} - \frac{21}{4}x_1^{366}x_6^{68}x_5^{343}, \\ x_1^{36}x_1^{194}x_3^{50}x_4^{22}x_5^{60} + x_1^{76}x_2^{240}x_4^{41}x_5 + x_1^{74}x_2^{179}x_3^{25}x_5^{57} + x_1^{25}x_2^{203}x_3^{44}x_4 + x_1^{20}x_2^{167}x_4^{64}x_1^{2}x_6^{86} - 20709cx_1^{58}x_1^{194}x_3^{44}x_6^{52} - \frac{21}{4}x_1^{366}x_6^{68}x_5^{343}, \\ x_1^{36}x_1^{294}x_3^{50}x_4^{25}x_5^{60} + x_1^{76}x_2^{240}x_4^{41}x_5 + x_1^{74}x_2^{179}x_3^{25}x_5^{57} + x_1^{25}x_2^{230}x_3^{44}x_4 + x_1^{20}x_2^{167}x_4^{64}x_4^{2}x_5^{86} - 20754cx_1^{58}x_1^{94}x_3^{44}x_6^{55} - \frac{21}{4}x_3^{166}x_6^{68}x_3^{343}, \\ x_1^{52}x_1^{52}x_2^{52}x_3^{52} + x_1^{75}x_2^{52}x_3^{57} + x_1^{52}x_2^{230}x_3^{44}x_4 + x_1^{20}x_2^{167}x_3^{64}x_4^{12}x_5^{86} - 20754cx_1^{58}x_2^{19}x_3^{44}x_$

[Rojas, 2020] For any fixed n, one can exactly count the roots in \mathbb{R}^n_+ , $(\mathbb{R}^*)^n$, and \mathbb{R}^n of any generic (n+2)-nomial $n \times n$ polynomial system over \mathbb{Q} in deterministic polynomial-time.

Example: In <1 second, we can see that the 7-nomial 5×5 system defined by

 $\begin{pmatrix} 2x_1^{36}x_2^{194}x_3^{50}x_2^{82}x_5^{60} + x_1^{76}x_2^{240}x_4^{41}x_5 + x_1^{74}x_2^{179}x_3^{25}x_5^{57} + x_1^{25}x_2^{203}x_3^{44}x_4 + x_1^{20}x_2^{167}x_3^{64}x_4^{12}x_5^{68} - 37125x_1^{194}x_3^{14}x_3^{4}x_5^{25} - \frac{9}{2}x_3^{166}x_4^{68}x_5^{343}, \\ x_1^{36}x_2^{194}x_3^{50}x_4^{82}x_5^{60} + 2x_1^{76}x_2^{240}x_4^{41}x_5 + x_1^{74}x_2^{179}x_3^{25}x_5^{57} + x_1^{25}x_2^{203}x_3^{44}x_4 + x_1^{20}x_2^{167}x_5^{10} \text{let } \mathbf{roots}_1 \\ x_1^{36}x_2^{194}x_3^{30}x_4^{25}x_5^{60} + x_1^{76}x_2^{240}x_4^{41}x_5 + 2x_1^{74}x_1^{179}x_3^{25}x_5^{57} + x_1^{25}x_2^{203}x_3^{44}x_4 + x_1^{20}x_2^{167}x_5^{10} \text{let } \mathbf{roots}_1 \\ x_1^{36}x_2^{194}x_3^{50}x_4^{82}x_5^{60} + x_1^{76}x_2^{240}x_4^{41}x_5 + x_1^{74}x_1^{179}x_3^{25}x_5^{57} + x_1^{25}x_2^{203}x_3^{44}x_4 + x_1^{20}x_2^{167}x_6^{44}x_5^{26} - 21009cx_1^{58}x_2^{194}x_3^{43}x_4^{36}x_5^{57} - \frac{21}{4}x_1^{36}x_6^{48}x_5^{343}, \\ x_1^{36}x_1^{194}x_3^{50}x_4^{82}x_5^{60} + x_1^{76}x_2^{240}x_4^{41}x_5 + x_1^{74}x_1^{179}x_2^{25}x_5^{57} + x_2^{25}x_2^{203}x_4^{44}x_4 + x_1^{20}x_2^{167}x_6^{44}x_1^{2}x_5^{68} - 20709cx_1^{58}x_2^{194}x_3^{44}x_4^{36}x_5^{25} - \frac{21}{4}x_1^{36}x_6^{48}x_5^{343}, \\ x_1^{36}x_1^{194}x_3^{50}x_4^{82}x_5^{60} + x_1^{76}x_2^{240}x_4^{41}x_5 + x_1^{14}x_1^{179}x_2^{24}x_2^{44}x_4 + x_2^{10}x_2^{167}x_6^{44}x_1^{2}x_5^{68} - 20769cx_1^{58}x_2^{194}x_3^{44}x_4^{36}x_5^{25} - \frac{21}{4}x_3^{16}x_6^{68}x_5^{343}, \\ x_1^{36}x_1^{194}x_3^{50}x_4^{82}x_5^{60} + x_1^{76}x_2^{240}x_4^{41}x_5 + x_1^{14}x_1^{179}x_2^{24}x_3^{44}x_4 + 2x_1^{20}x_2^{167}x_3^{64}x_4^{12}x_5^{68} - 20769cx_1^{58}x_2^{194}x_3^{44}x_4^{36}x_5^{25} - \frac{21}{4}x_3^{16}x_6^{68}x_5^{343}, \\ x_1^{36}x_2^{194}x_3^{50}x_4^{82}x_5^{60} + x_1^{76}x_2^{240}x_4^{41}x_5 + x_1^{14}x_2^{52}x_2^{210}x_3^{44}x_4 + 2x_1^{20}x_2^{167}x_3^{64}x_4^{12}x_5^{68} - 20754cx_1^{58}x_2^{194}x_3^{44}x_4^{36}x_5^{5} - \frac{21}{4}x_3^{16}x_6^{68}x_5^{343}, \\ has exactly 2, 6, 6, 2, 2, or 0 positive roots, respectively \\ when c is \frac{1}{20731}, \frac{1}{20730}, \frac{1}{14399}, \frac{1}{14399},$

Fewnomials and the Quest for Tight Bounds: 2000s



[Bihan, Sottile, 2007] Let $c_{i,j} \in \mathbb{R}$ for all i, j,



Counting Real Roots in Polynomial-Time

Fewnomials and the Quest for Tight Bounds: 2000s



[Bihan, Sottile, 2007] Let $c_{i,j} \in \mathbb{R}$ for all i, j, let $a_1, \ldots, a_{n+k} \in \mathbb{R}^n$ be any points not all lying in an affine hyperplane,



Fewnomials and the Quest for Tight Bounds: 2000s



[Bihan, Sottile, 2007] Let $c_{i,j} \in \mathbb{R}$ for all i, j, let $a_1, \ldots, a_{n+k} \in \mathbb{R}^n$ be any points not all lying in an affine hyperplane, and write $x^{a_i} = x_1^{a_{1,i}} \cdots x_n^{a_{n,i}}$.



Fewnomials and the Quest for Tight Bounds: 2000s



[Bihan, Sottile, 2007] Let $c_{i,j} \in \mathbb{R}$ for all i, j, let $a_1, \ldots, a_{n+k} \in \mathbb{R}^n$ be any points not all lying in an affine hyperplane, and write $x^{a_i} = x_1^{a_{1,i}} \cdots x_n^{a_{n,i}}$.

Real Fewnomial Theorem⁺. Any real (n + k)-nomial $n \times n$ polynomial system of the form...

$$F := (f_1, \dots, f_n) = \begin{cases} c_{1,1}x^{a_1} + \dots + c_{1,n+k}x^{a_{n+k}} \\ \vdots \\ c_{n,1}x^{a_1} + \dots + c_{n,n+k}x^{a_{n+k}} \end{cases}$$



Fewnomials and the Quest for Tight Bounds: 2000s



[Bihan, Sottile, 2007] Let $c_{i,j} \in \mathbb{R}$ for all i, j, let $a_1, \ldots, a_{n+k} \in \mathbb{R}^n$ be any points not all lying in an affine hyperplane, and write $x^{a_i} = x_1^{a_{1,i}} \cdots x_n^{a_{n,i}}$.

Real Fewnomial Theorem⁺. Any real (n + k)-nomial $n \times n$ polynomial system of the form...

$$F := (f_1, \dots, f_n) = \begin{cases} c_{1,1}x^{a_1} + \dots + c_{1,n+k}x^{a_{n+k}} \\ \vdots \\ c_{n,1}x^{a_1} + \dots + c_{n,n+k}x^{a_{n+k}} \end{cases}$$

has no more than $\frac{e^2+3}{4}2^{(k-1)(k-2)/2}n^{k-1}$ non-degenerate roots in \mathbb{R}^n_+



Fewnomials and the Quest for Tight Bounds: 2000s



[Bihan, Sottile, 2007] Let $c_{i,j} \in \mathbb{R}$ for all i, j, let $a_1, \ldots, a_{n+k} \in \mathbb{R}^n$ be any points not all lying in an affine hyperplane, and write $x^{a_i} = x_1^{a_{1,i}} \cdots x_n^{a_{n,i}}$.

Real Fewnomial Theorem⁺. Any real (n + k)-nomial $n \times n$ polynomial system of the form...

$$F := (f_1, \dots, f_n) = \begin{cases} c_{1,1}x^{a_1} + \dots + c_{1,n+k}x^{a_{n+k}} \\ \vdots \\ c_{n,1}x^{a_1} + \dots + c_{n,n+k}x^{a_{n+k}} \end{cases}$$

has no more than $\frac{e^2+3}{4}2^{(k-1)(k-2)/2}n^{k-1}$ non-degenerate roots in \mathbb{R}^n_+ . [Bihan, Rojas, Sottile '07]: \exists systems with $\left\lfloor \frac{n+k-1}{\min\{n,k-1\}} \right\rfloor^{\min\{n,k-1\}}$ positive roots.

Fewnomials and the Quest for Tight Bounds: 2000s



[Bihan, Sottile, 2007] Let $c_{i,j} \in \mathbb{R}$ for all i, j, let $a_1, \ldots, a_{n+k} \in \mathbb{R}^n$ be any points not all lying in an affine hyperplane, and write $x^{a_i} = x_1^{a_{1,i}} \cdots x_n^{a_{n,i}}$.

Real Fewnomial Theorem⁺. Any real (n + k)-nomial $n \times n$ polynomial system of the form...

$$F := (f_1, \dots, f_n) = \begin{cases} c_{1,1}x^{a_1} + \dots + c_{1,n+k}x^{a_{n+k}} \\ \vdots \\ c_{n,1}x^{a_1} + \dots + c_{n,n+k}x^{a_{n+k}} \end{cases}$$

has no more than $\frac{e^2+3}{4}2^{(k-1)(k-2)/2}n^{k-1}$ non-degenerate roots in \mathbb{R}^n_+ [Bertrand, Bihan, Sottile '06]: Tight bound for k = 2 of n+1.



Fewnomials and the Quest for Tight Bounds: 2010s



Theorem. [Bürgisser, Ergür, Tonelli-Cueto, 2019] For an (n+k)-nomial $n \times n$ system with independent standard real Gaussian coefficients



Fewnomials and the Quest for Tight Bounds: 2010s



Theorem. [Bürgisser, Ergür, Tonelli-Cueto, 2019] For an (n + k)-nomial $n \times n$ system with independent standard real Gaussian coefficients and fixed support not lying in an affine hyperplane,



A (10) × (10) × (10)

Fewnomials and the Quest for Tight Bounds: 2010s



Theorem. [Bürgisser, Ergür, Tonelli-Cueto, 2019] For an (n+k)-nomial $n \times n$ system with independent standard real Gaussian coefficients and fixed support not lying in an affine hyperplane, the average number of roots in $(\mathbb{R}^*)^n$ is $\leq \frac{1}{2} \binom{n+k}{k}$.



- 4 同 1 - 4 回 1 - 4 回

Bisection \longrightarrow Diophantine Approximation

Lemma. (e.g., [Ye, 1995]) $O(\log D)$ steps of bisection are enough to get you a succinct approximant to $\sqrt[D]{c}$



Counting Real Roots in Polynomial-Time

(4 回 ト 4 回 ト 4 回 ト

Bisection \longrightarrow Diophantine Approximation

Lemma. (e.g., [Ye, 1995]) $O(\log D)$ steps of bisection are enough to get you a succinct approximant to $\sqrt[D]{c}$ for any $D \in \mathbb{N}$ and $c \in \mathbb{Q}_+$.



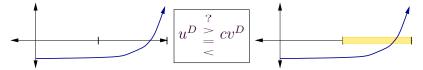
(4 回 ト 4 回 ト 4 回 ト

Bisection \longrightarrow Diophantine Approximation

Lemma. (e.g., [Ye, 1995]) $O(\log D)$ steps of bisection are enough to get you a succinct approximant to $\sqrt[D]{c}$ for any $D \in \mathbb{N}$ and $c \in \mathbb{Q}_+$. Counting bit operations, we get time $O(\operatorname{size}(x_1^D - c)^{2+\varepsilon}).$



Lemma. (e.g., [Ye, 1995]) $O(\log D)$ steps of bisection are enough to get you a succinct approximant to $\sqrt[D]{c}$ for any $D \in \mathbb{N}$ and $c \in \mathbb{Q}_+$. Counting bit operations, we get time $O(\operatorname{size}(x_1^D - c)^{2+\varepsilon}).$



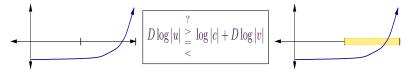


Lemma. (e.g., [Ye, 1995]) $O(\log D)$ steps of bisection are enough to get you a succinct approximant to $\sqrt[D]{c}$ for any $D \in \mathbb{N}$ and $c \in \mathbb{Q}_+$. Counting bit operations, we get time $O(\operatorname{size}(x_1^D - c)^{2+\varepsilon}).$





Lemma. (e.g., [Ye, 1995]) $O(\log D)$ steps of bisection are enough to get you a succinct approximant to $\sqrt[D]{c}$ for any $D \in \mathbb{N}$ and $c \in \mathbb{Q}_+$. Counting bit operations, we get time $O(\operatorname{size}(x_1^D - c)^{2+\varepsilon}).$





Lemma. (e.g., [Ye, 1995]) $O(\log D)$ steps of bisection are enough to get you a succinct approximant to $\sqrt[D]{c}$ for any $D \in \mathbb{N}$ and $c \in \mathbb{Q}_+$. Counting bit operations, we get time $O(\operatorname{size}(x_1^D - c)^{2+\varepsilon}).$



[Baker, 1966] If $a_i, b_i \in \mathbb{Z}$ with $A := \max_i \log |a_i|$, $B := \max_i \log |b_i|$, and $\Lambda := \sum_{i=1}^n b_i \log a_i$, then $\Lambda \neq 0 \Longrightarrow |\log |\Lambda|| = O(A)^n \log B.$



Real Univariate Trinomials?

•Deciding the sign of a trinomial $f \in \mathbb{Q}[x_1]$ at $z \in \mathbb{Q}$ in time $(\operatorname{size}(f) + \operatorname{height}(z))^{O(1)}$ is an open problem!





Counting Real Roots in Polynomial-Time

- 4 同 1 - 4 回 1 - 4 回 1

Real Univariate Trinomials?

•Deciding the sign of a trinomial $f \in \mathbb{Q}[x_1]$ at $z \in \mathbb{Q}$ in time $(\operatorname{size}(f) + \operatorname{height}(z))^{O(1)}$ is an open problem!

•So bisection is obstructed!





(4) (日本) (日本) (日本)

Real Univariate Trinomials?

- •Deciding the sign of a trinomial $f \in \mathbb{Q}[x_1]$ at $z \in \mathbb{Q}$ in time $(\operatorname{size}(f) + \operatorname{height}(z))^{O(1)}$ is an open problem!
- $\bullet So$ bisection is obstructed!



•Despite nice progress by [Jindal, Sagraloff, 2017] on *coarse* approximants,



- 4 同 1 - 4 回 1 - 4 回 1

Real Univariate Trinomials?

- •Deciding the sign of a trinomial $f \in \mathbb{Q}[x_1]$ at $z \in \mathbb{Q}$ in time $(\operatorname{size}(f) + \operatorname{height}(z))^{O(1)}$ is an open problem!
- $\bullet So$ bisection is obstructed!



- 4 同 1 4 日 1 4 日 1

•Despite nice progress by [Jindal, Sagraloff, 2017] on *coarse* approximants, real root counting in time $(t \log(dH))^{O(1)}$ is still an open problem!



Exact Counting for (n+2)-nomial $n \times n$ Systems

Main Theorem. [Rojas, 2020] For any (n + 2)-nomial $n \times n$ system F over \mathbb{Q} ,



- - E - b

Counting Real Roots in Polynomial-Time

Main Theorem. [Rojas, 2020] For any (n + 2)-nomial $n \times n$ system F over \mathbb{Q} , with exponents not lying in a common affine hyperplane,



Main Theorem. [Rojas, 2020] For any (n + 2)-nomial $n \times n$ system F over \mathbb{Q} , with exponents not lying in a common affine hyperplane, and coefficient matrix $[c_{i,j}]$ of rank^{*} n,



Main Theorem. [Rojas, 2020] For any (n + 2)-nomial $n \times n$ system F over \mathbb{Q} , with exponents not lying in a common affine hyperplane, and coefficient matrix $[c_{i,j}]$ of rank^{*} n, we can count exactly its number of roots in \mathbb{R}^{n}_{+} ,



Main Theorem. [Rojas, 2020] For any (n + 2)-nomial $n \times n$ system F over \mathbb{Q} , with exponents not lying in a common affine hyperplane, and coefficient matrix $[c_{i,j}]$ of rank^{*} n, we can count exactly its number of roots in \mathbb{R}^n_+ , $(\mathbb{R}^*)^n$,



Main Theorem. [Rojas, 2020] For any (n + 2)-nomial $n \times n$ system F over \mathbb{Q} , with exponents not lying in a common affine hyperplane, and coefficient matrix $[c_{i,j}]$ of rank^{*} n, we can count exactly its number of roots in \mathbb{R}^n_+ , $(\mathbb{R}^*)^n$, and \mathbb{R}^n ,



Main Theorem. [Rojas, 2020] For any (n + 2)-nomial $n \times n$ system F over \mathbb{Q} , with exponents not lying in a common affine hyperplane, and coefficient matrix $[c_{i,j}]$ of rank^{*} n, we can count exactly its number of roots in \mathbb{R}^n_+ , $(\mathbb{R}^*)^n$, and \mathbb{R}^n , in deterministic time $O(n \log(nd) + n^2 \log(nH)))^{2n+4}$.



Main Theorem. [Rojas, 2020] For any (n + 2)-nomial $n \times n$ system F over \mathbb{Q} , with exponents not lying in a common affine hyperplane, and coefficient matrix $[c_{i,j}]$ of rank^{*} n, we can count exactly its number of roots in \mathbb{R}^n_+ , $(\mathbb{R}^*)^n$, and \mathbb{R}^n , in deterministic time $O(n \log(nd) + n^2 \log(nH)))^{2n+4}$.

 $\bullet \mathrm{Key}$ new ingredients are a refined version of Liouville's Theorem,



Main Theorem. [Rojas, 2020] For any (n + 2)-nomial $n \times n$ system F over \mathbb{Q} , with exponents not lying in a common affine hyperplane, and coefficient matrix $[c_{i,j}]$ of rank^{*} n, we can count exactly its number of roots in \mathbb{R}^n_+ , $(\mathbb{R}^*)^n$, and \mathbb{R}^n , in deterministic time $O(n \log(nd) + n^2 \log(nH)))^{2n+4}$.

•Key new ingredients are a refined version of *Liouville's Theorem*, and a theorem of [Baker, Wustholtz, 1993]



Main Theorem. [Rojas, 2020] For any (n + 2)-nomial $n \times n$ system F over \mathbb{Q} , with exponents not lying in a common affine hyperplane, and coefficient matrix $[c_{i,j}]$ of rank^{*} n, we can count exactly its number of roots in \mathbb{R}^n_+ , $(\mathbb{R}^*)^n$, and \mathbb{R}^n , in deterministic time $O(n \log(nd) + n^2 \log(nH)))^{2n+4}$.

•Key new ingredients are a refined version of *Liouville's Theorem*, and a theorem of [Baker, Wustholtz, 1993] on linear combinations of logs in *algebraic* numbers...



(4 回 ト 4 ヨト 4 ヨト

Main Theorem. [Rojas, 2020] For any (n + 2)-nomial $n \times n$ system F over \mathbb{Q} , with exponents not lying in a common affine hyperplane, and coefficient matrix $[c_{i,j}]$ of rank^{*} n, we can count exactly its number of roots in \mathbb{R}^n_+ , $(\mathbb{R}^*)^n$, and \mathbb{R}^n , in deterministic time $O(n \log(nd) + n^2 \log(nH)))^{2n+4}$.

•Key new ingredients are a refined version of *Liouville's Theorem*, and a theorem of [Baker, Wustholtz, 1993] on linear combinations of logs in *algebraic* numbers...

•Sufficiently refined versions of the abc-Conjecture can reduce the complexity to polynomial in n...



Key Idea #1: Gaussian Elimination

By Gauss-Jordan Elimination applied to the linear combinations of monomials, our original 5×5 example can be reduced to

$$\begin{array}{rcl} x_1^{36}x_1^{194}x_3^{-116}x_4^{14}x_5^{-283} & = & 16384cx_1^{58}x_2^{194}x_3^{-142}x_4^{-32}x_5^{-318} + \frac{1}{4} \\ & x_1^{76}x_2^{240}x_3^{-166}x_4^{-27}x_5^{-342} & = & 4096cx_1^{18}x_2^{194}x_3^{-142}x_4^{-32}x_5^{-318} + 1 \\ & x_1^{74}x_2^{179}x_3^{-141}x_4^{-68}x_5^{-286} & = & 256cx_1^{58}x_2^{194}x_3^{-142}x_4^{-32}x_5^{-318} + 1 \\ & x_1^{25}x_2^{205}x_3^{-122}x_4^{-67}x_5^{-343} & = & 16cx_1^{58}x_2^{194}x_3^{-142}x_4^{-32}x_5^{-318} + 1 \\ & x_1^{20}x_2^{167}x_3^{-102}x_4^{-56}x_5^{-275} & = & cx_1^{58}x_2^{194}x_3^{-142}x_4^{-32}x_5^{-318} + 1 \end{array}$$



b 4 T b

Key Idea #1: Gaussian Elimination

By Gauss-Jordan Elimination applied to the linear combinations of monomials, our original 5×5 example can be reduced to

$$\begin{array}{rcl} x_1^{36}x_2^{194}x_3^{-116}x_4^{14}x_5^{-283} & = & 16384cx_1^{58}x_2^{194}x_3^{-142}x_5^{-32}x_5^{-318} + \frac{1}{4} \\ & x_1^{76}x_2^{240}x_3^{-166}x_4^{-27}x_5^{-342} & = & 4096cx_1^{58}x_2^{194}x_3^{-142}x_4^{-32}x_5^{-318} + 1 \\ & x_1^{74}x_2^{179}x_3^{-141}x_4^{-68}x_5^{-286} & = & 256cx_1^{58}x_2^{194}x_3^{-142}x_4^{-32}x_5^{-318} + 1 \\ & x_1^{25}x_2^{203}x_3^{-122}x_4^{-67}x_5^{-343} & = & 16cx_1^{58}x_2^{194}x_3^{-142}x_4^{-32}x_5^{-318} + 1 \\ & x_1^{20}x_2^{167}x_3^{-102}x_4^{-56}x_5^{-275} & = & cx_1^{58}x_2^{194}x_3^{-142}x_4^{-32}x_5^{-318} + 1 \end{array}$$

Next: Observe that the set of exponent vectors is a *circuit* in the sense of combinatorics:



Key Idea #1: Gaussian Elimination

By Gauss-Jordan Elimination applied to the linear combinations of monomials, our original 5×5 example can be reduced to

$$\begin{array}{rcl} x_1^{36}x_2^{194}x_3^{-116}x_4^{14}x_5^{-283} & = & 16384cx_1^{58}x_2^{194}x_3^{-142}x_4^{-32}x_5^{-318} + \frac{1}{4} \\ & x_1^{76}x_2^{240}x_3^{-166}x_4^{-27}x_5^{-342} & = & 4096cx_1^{58}x_2^{194}x_3^{-142}x_4^{-32}x_5^{-318} + 1 \\ & x_1^{74}x_2^{179}x_3^{-141}x_4^{-68}x_5^{-286} & = & 256cx_1^{58}x_2^{194}x_3^{-142}x_4^{-32}x_5^{-318} + 1 \\ & x_1^{25}x_2^{203}x_3^{-122}x_4^{-67}x_5^{-343} & = & 16cx_1^{58}x_2^{194}x_3^{-142}x_4^{-32}x_5^{-318} + 1 \\ & x_1^{20}x_2^{167}x_3^{-102}x_4^{-56}x_5^{-275} & = & cx_1^{58}x_2^{194}x_3^{-142}x_4^{-32}x_5^{-318} + 1 \end{array}$$

Next: Observe that the set of exponent vectors is a *circuit* in the sense of combinatorics: It is the union of a point and the vertex set of a simplex...



Key Idea #2: Gale Dual Form

The exponent vectors a_1, \ldots, a_7 in our example have a *unique* affine relation:



Counting Real Roots in Polynomial-Time

Key Idea #2: Gale Dual Form

The exponent vectors a_1, \ldots, a_7 in our example have a *unique* affine relation:

 $-2a_1+2a_2-2a_3+2a_4-2a_5+a_6+a_7=\mathbf{O}.$



Key Idea #2: Gale Dual Form

The exponent vectors a_1, \ldots, a_7 in our example have a *unique* affine relation:

 $-2a_1 + 2a_2 - 2a_3 + 2a_4 - 2a_5 + a_6 + a_7 = \mathbf{O}.$ So then, $\zeta \in \mathbb{R}^5_+$ a root $\Longrightarrow u := \zeta_1^{58} \zeta_2^{194} \zeta_4^{-32} \zeta_5^{-318}$ must be a root



Key Idea #2: Gale Dual Form

The exponent vectors a_1, \ldots, a_7 in our example have a *unique* affine relation:

 $\begin{array}{l} -2a_1+2a_2-2a_3+2a_4-2a_5+a_6+a_7\!=\!\mathbf{O}.\\ \text{So then, } \zeta\in\mathbb{R}^5_+\text{ a root}\Longrightarrow u\!:=\!\zeta_1^{58}\zeta_2^{194}\zeta_4^{-32}\zeta_5^{-318} \text{ must be a root of the } Gale \ Dual \ rational \ function\\ (16384cu+1)^{-2}(4096cu+1)^2(256cu+1)^{-2}(16cu+1)^2(cu+1)^{-2}u-1. \end{array}$

Equivalently, u must be a real root of the *linear* combination of logarithms



(4 回 ト 4 回 ト 4 回 ト

Key Idea #2: Gale Dual Form

The exponent vectors a_1, \ldots, a_7 in our example have a *unique* affine relation:

 $\begin{array}{l} -2a_1+2a_2-2a_3+2a_4-2a_5+a_6+a_7\!=\!\mathbf{O}.\\ \text{So then, } \zeta\in\mathbb{R}^5_+\text{ a root}\Longrightarrow u\!:=\!\zeta_1^{58}\zeta_2^{194}\zeta_4^{-32}\zeta_5^{-318} \text{ must be a root of the } Gale \ Dual \ rational \ function\\ (16384cu+1)^{-2}(4096cu+1)^2(256cu+1)^{-2}(16cu+1)^2(cu+1)^{-2}u-1. \end{array}$

Equivalently, u must be a real root of the *linear* combination of logarithms

 $-2 \log |16384cu+1| + 2 \log |4096cu+1| - 2 \log |256cu+1| + 2 \log |16cu+1| - 2 \log |cu+1| + \log |u|,$



・ロト ・ 母 ト ・ ヨ ト ・ ヨ ト

Key Idea #2: Gale Dual Form

The exponent vectors a_1, \ldots, a_7 in our example have a *unique* affine relation:

 $\begin{array}{l} -2a_1+2a_2-2a_3+2a_4-2a_5+a_6+a_7\!=\!\mathbf{O}.\\ \text{So then, } \zeta\in\mathbb{R}^5_+\text{ a root}\Longrightarrow u\!:=\!\zeta_1^{58}\zeta_2^{194}\zeta_4^{-32}\zeta_5^{-318} \text{ must be a root of the } Gale \ Dual \ rational \ function\\ (16384cu+1)^{-2}(4096cu+1)^2(256cu+1)^{-2}(16cu+1)^2(cu+1)^{-2}u-1. \end{array}$

Equivalently, u must be a real root of the *linear* combination of logarithms -2 log |16384cu + 1| + 2 log |4096cu + 1| - 2 log |256cu + 1| + 2 log |16cu + 1| - 2 log |cu + 1| + log |u|,

provided some additional sign conditions are met...

Note: The exponents are usually *much* larger!



・ロト ・ 日 ・ ・ 日 ・ ・ 日 ・

Key Idea #3: Critical Points and Diophantine Approximation

Examine the graph of the linear combination of logarithms...

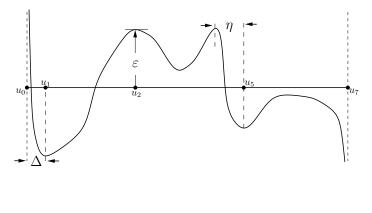


Counting Real Roots in Polynomial-Time

< A > <

Key Idea #3: Critical Points and Diophantine Approximation

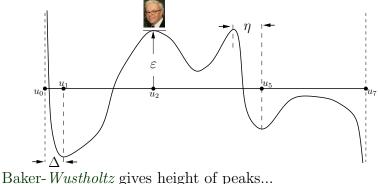
Examine the graph of the linear combination of logarithms...

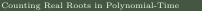




Key Idea #3: Critical Points and Diophantine Approximation

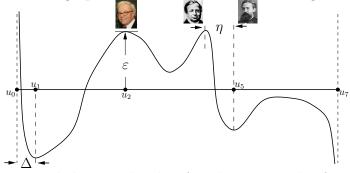
Examine the graph of the linear combination of logarithms...





Key Idea #3: Critical Points and Diophantine Approximation

Examine the graph of the linear combination of logarithms...

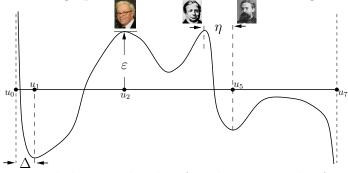


Baker-*Wustholtz* gives height of peaks... Bounds of Liouville and Markov control Δ and η ...



Key Idea #3: Critical Points and Diophantine Approximation

Examine the graph of the linear combination of logarithms...

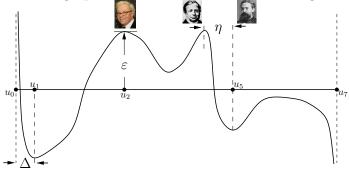


Baker-*Wustholtz* gives height of peaks... Bounds of Liouville and Markov control Δ and η ... Then use Rolle's Theorem,



Key Idea #3: Critical Points and Diophantine Approximation

Examine the graph of the linear combination of logarithms...

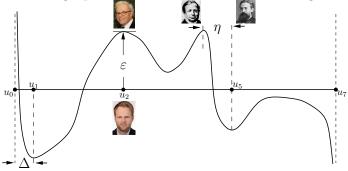


Baker-*Wustholtz* gives height of peaks... Bounds of Liouville and Markov control Δ and η ... Then use Rolle's Theorem, AGM Iteration for accurate logs,



Key Idea #3: Critical Points and Diophantine Approximation

Examine the graph of the linear combination of logarithms...



Baker-*Wustholtz* gives height of peaks... Bounds of Liouville and Markov control Δ and η ... Then use Rolle's Theorem, AGM Iteration for accurate logs, and Sturm-Habicht sequences to isolate critical points!...



Counting Real Roots in Polynomial-Time

Upshot: Randomize!

Even though there are (n + 1)-nomial $n \times n$ systems,



Counting Real Roots in Polynomial-Time

イロト イボト イヨト イヨト

Even though there are (n + 1)-nomial $n \times n$ systems, and univariate tetranomials with exponentially close real roots,



(4 回 ト 4 ヨト 4 ヨト

Even though there are (n + 1)-nomial $n \times n$ systems, and univariate tetranomials with exponentially close real roots, these appear to be rare in practice [Mignotte, 1995; Paouris, Phillipson, Rojas, 2019; Rojas, Zhu, 2021].





Even though there are (n + 1)-nomial $n \times n$ systems, and univariate tetranomials with exponentially close real roots, these appear to be rare in practice [Mignotte, 1995; Paouris, Phillipson, Rojas, 2019; Rojas, Zhu, 2021].



While we can now count real roots in time $(\log(dH))^{O(n)}$, there



Even though there are (n + 1)-nomial $n \times n$ systems, and univariate tetranomials with exponentially close real roots, these appear to be rare in practice [Mignotte, 1995; Paouris, Phillipson, Rojas, 2019; Rojas, Zhu, 2021].



While we can now count real roots in time $(\log(dH))^{O(n)}$, there is growing evidence that we can attain complexity $(n \log(DH))^{O(1)}$



Even though there are (n + 1)-nomial $n \times n$ systems, and univariate tetranomials with exponentially close real roots, these appear to be rare in practice [Mignotte, 1995; Paouris, Phillipson, Rojas, 2019; Rojas, Zhu, 2021].



While we can now count real roots in time $(\log(dH))^{O(n)}$, there is growing evidence that we can attain complexity $(n \log(DH))^{O(1)}$ on average [Deng, Ergür, Paouris, Rojas, 2021]...





\heartsuit Thank you for your attention!

See www.math.tamu.edu/~rojas for preprints and further info...



Counting Real Roots in Polynomial-Time