Polynomial-Time Amoeba Neighborhood Membership and Faster Localized Solving

Eleanor Anthony, Sheridan Grant, Peter Gritzmann, and J. Maurice Rojas

Dedicated to Tien-Yien Li, in honor of his birthday.

1 Introduction

As students, we are often asked to draw (hopefully without a calculator) real zero sets of low degree polynomials in few variables. As scientists and engineers, we are often asked to count or approximate (hopefully with some computational assistance) real and complex solutions of arbitrary systems of polynomial equations in many variables. If one allows sufficiently coarse approximations, then the latter problem is as easy as the former. Our main results clarify this transition from hardness to easiness. In particular, we significantly speed up certain queries involving

E. Anthony
Mathematics Department, University of Mississippi, Hume Hall 305, P.O. Box 1848, MS 38677-1848, USA
e-mail: ecanthon@go.olemiss.edu
Partially supported by NSF REU grant DMS-1156589.

S. Grant
Mathematics Department, 640 North College Avenue, Claremont, CA 91711, USA
e-mail: sheridan.grant@pomona.edu
Partially supported by NSF REU grant DMS-1156589.

P. Gritzmann
Fakultät für Mathematik, Technische Universität München, 80290 München, Germany
e-mail: gritzmann@tum.de
Work supported in part by the German Research Foundation (DFG).

J.M. Rojas (✉)
Mathematics Department, Texas A&M University, TAMU 3368, College Station, TX 77843-3368, USA
e-mail: rojas@math.tamu.edu
Partially supported by NSF MCS grant DMS-0915245.

J. Bennett et al. (eds.), *Topological and Statistical Methods for Complex Data*, Mathematics and Visualization, DOI 10.1007/978-3-662-44900-4_15

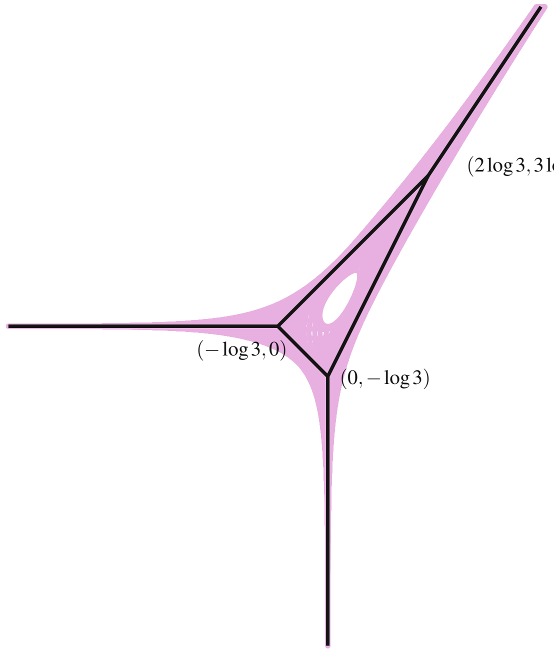
distances between points and complex algebraic hypersurfaces (see Theorems 1.4–1.6 below). We then apply our metric results to finding specially constructed start systems—dramatically speeding up traditional homotopy continuation methods—to approximate, or rule out, roots of selected norm (see Sect. 3).

Polynomial equations are ubiquitous in numerous applications, such as algebraic statistics [29], chemical reaction kinetics [42], discretization of partial differential equations [28], satellite orbit design [47], circuit complexity [36], and cryptography [10]. The need to solve larger and larger equations, in applications as well as for theoretical purposes, has helped shape algebraic geometry and numerical analysis for centuries. More recent work in algebraic complexity tells us that many basic questions involving polynomial equations are **NP**-hard (see, e.g., [13, 52]). This is by no means an excuse to consider polynomial equation solving hopeless: Computational scientists solve problems of near-exponential complexity every day.

Thanks to recent work on *Smale's 17th Problem* [8, 14], we have learned that randomization and approximation can be the key to avoiding the bottlenecks present in deterministic algorithms for solving hard questions involving complex roots of polynomial systems. Smale's 17th Problem concerns the *average-case* complexity of approximating *a single* complex root of a random polynomial system and is well-discussed in [54–58, 60, 61]. Our ultimate goal is to extend this philosophy to the harder problem of *localized solving*: estimating how far the *nearest* root of a given system of polynomials (or intersection of several zero sets) is from a given point. Here, we start by first approximating the shape of a single zero set, and then in Sect. 3 we outline a tropical-geometric approach to localized solving. Toward this end, let us first recall the natural idea (see, e.g., [65]) of drawing zero sets on log-paper. In what follows, we let $\mathbb{C}^*$ denote the non-zero complex numbers and write $\mathbb{C}\left[x_1^{\pm 1}, \ldots, x_n^{\pm 1}\right]$ for the ring of Laurent polynomials with complex coefficients, i.e., polynomials with negative exponents allowed. Also, for any two vectors $u := (u_1, \ldots, u_N)$ and $v := (v_1, \ldots, v_N)$ in $\mathbb{R}^N$, we use $u \cdot v$ to denote the standard dot product $u_1v_1 + \cdots + u_Nv_N$.

Definition 1.1 We set $x := (x_1, \ldots, x_n)$ and $\mathrm{Log}|x| := (\log|x_1|, \ldots, \log|x_n|)$, and, for any $f \in \mathbb{C}\left[x_1^{\pm 1}, \ldots, x_n^{\pm 1}\right]$, we define $\mathrm{Amoeba}(f)$ to be the set $\{\mathrm{Log}|x| \;:\; f(x) = 0\ ,\ x \in (\mathbb{C}^*)^n\}$. We call f an *n-variate t-nomial* when we can write $f(x) = \sum_{i=1}^{t} c_i x^{a_i}$ with $c_i \neq 0$, $a_i := (a_{1,i}, \ldots, a_{n,i})$, the a_i are pair-wise distinct, and $x^{a_i} := x_1^{a_{1,i}} x_2^{a_{2,i}} \cdots x_n^{a_{n,i}}$ for all i. When f is not the zero polynomial, we define the *Archimedean tropical variety of f*, denoted $\mathrm{ArchTrop}(f)$, to be the set of all $w \in \mathbb{R}^n$ for which $\max_i |c_i e^{a_i \cdot w}|$ is attained for at least *two* distinct indices i. Finally, we define $\mathrm{ArchTrop}(0)$ to be $\mathbb{R}^n$. $\diamond$

In Sect. 3 we will see how amoebae and tropical varieties are useful for speeding up polynomial system solving.



Example 1.2 Taking $f(x)=1+x_1^3+x_2^2-3x_1x_2$, an illustration of Amoeba(f) and ArchTrop(f), truncated to $[-7,7]^2$, appears above. Amoeba(f) is lightly shaded, while ArchTrop(f) is the piecewise-linear curve. ⋄

One may be surprised that Amoeba(f) and ArchTrop(f) are so highly structured: Amoeba(f) has tentacles reminiscent of a living amoeba, and ArchTrop(f) is a *polyhedral complex*, i.e., a union of polyhedra intersecting only along common faces (see Definition 2.7 below). One may also be surprised that Amoeba(f) and ArchTrop(f) are so closely related: Every point of one set is close to some point of the other, and both sets have topologically similar complements (4 open connected components, exactly one of which is bounded). Example 2.2 below shows that we need not always have ArchTrop$(f)\subseteq$Amoeba(f).

To quantify how close Amoeba(f) and ArchTrop(f) are in general, one can recall the *Hausdorff distance*, denoted $\Delta(U,V)$, between two subsets $U,V\subseteq\mathbb{R}^n$: It is defined to be the maximum of $\sup_{u\in U}\inf_{v\in V}|u-v|$ and $\sup_{v\in V}\inf_{u\in U}|u-v|$. We then have the following recent result of Avendaño, Kogan, Nisse, and Rojas.

Theorem 1.3 ([4]) *Suppose f is any n-variate t-nomial. Then* Amoeba(f) *and* ArchTrop(f) *are (a) identical for $t\leq 2$ and (b) at Hausdorff distance no greater than* $(2t-3)\log(t-1)$ *for $t\geq 3$. In particular, for $t\geq 2$, we also have*

$$\sup_{u\in \text{Amoeba}(f)}\ \inf_{v\in \text{ArchTrop}(f)}|u-v|\leq \log(t-1).$$

Finally, for any $t > n \geq 1$, there is an n-variate t-nomial f with

$$\Delta(\mathrm{Amoeba}(f), \mathrm{ArchTrop}(f)) \geq \log(t-1). \qquad \blacksquare$$

Note that the preceding upper bounds are completely independent of the coefficients, degree, and number of variables of f. Our upcoming examples show that Amoeba(f) and ArchTrop(f) are sometimes much closer than the bound above.

Our first two main results help set the stage for applying Archimedean tropical varieties to speed up polynomial root approximation. Recall that $\mathbb{Q}[\sqrt{-1}]$ denotes those complex numbers whose real and imaginary parts are both rational. Our complexity results will all be stated relative to the classical Turing (bit) model, with the underlying notion of input size clarified below in Definition 1.7.

Theorem 1.4 *Suppose $w \in \mathbb{R}^n$ and $f \in \mathbb{C}\left[x_1^{\pm 1}, \ldots, x_n^{\pm 1}\right]$ is a t-nomial with $t \geq 2$. Then*

$$-\log(t-1) \leq \inf_{u \in \mathrm{Amoeba}(f)} |u - w| - \inf_{v \in \mathrm{ArchTrop}(f)} |v - w| \leq (2t-3)\log(t-1).$$

In particular, if we also assume that n is fixed and $(f, w) \in \mathbb{Q}[\sqrt{-1}]\left[x_1^{\pm 1}, \ldots, x_n^{\pm 1}\right] \times \mathbb{Q}^n$ with f a t-nomial, then we can compute polynomially many bits of $\inf_{v \in \mathrm{ArchTrop}(f)} |v - w|$ in polynomial-time, and there is a polynomial-time algorithm that declares either (a) $\inf_{u \in \mathrm{Amoeba}(f)} |u - w| \leq (2t-2)\log(t-1)$ or (b) $w \notin \mathrm{Amoeba}(f)$ and $\inf_{u \in \mathrm{Amoeba}(f)} |u - w| \geq \inf_{v \in \mathrm{ArchTrop}(f)} |v - w| - \log(t-1) > 0$.

Theorem 1.4 is proved in Sect. 5. The importance of Theorem 1.4 is that deciding whether an input rational point w lies in an input Amoeba(f), *even restricting to the special case $n = 1$*, is already **NP**-hard [4].

ArchTrop(f) naturally partitions $\mathbb{R}^n$ into finitely many (relatively open) polyhedral cells of dimension 0 through n. We call the resulting polyhedral complex $\Sigma(\mathrm{ArchTrop}(f))$ (see Definition 2.7 below). In particular, finding the cell of $\Sigma(\mathrm{ArchTrop}(f))$ containing a given $w \in \mathbb{R}^n$ gives us more information than simply deciding whether w lies in ArchTrop(f).

Theorem 1.5 *Suppose n is fixed. Then there is a polynomial-time algorithm that, for any input $(f, w) \in \mathbb{Q}\left[\sqrt{-1}\right]\left[x_1^{\pm 1}, \ldots, x_n^{\pm 1}\right] \times \mathbb{Q}^n$ with f a t-nomial, outputs the closure of the unique cell σ_w of $\Sigma(\mathrm{ArchTrop}(f))$ containing w, described as an explicit intersection of $O(t^2)$ half-spaces.*

Theorem 1.5 is proved in Sect. 4. As a consequence, we can also find explicit regions, containing a given query point w, where f can not vanish. Let d denote the degree of f. While our present algorithm evincing Theorem 1.5 has complexity exponential in n, its complexity is polynomial in $\log d$ (see Definition 1.7 below). The best previous techniques from computational algebra, including recent advances on Smale's 17th Problem [8, 14], yield complexity no better than polynomial in $\frac{(d+n)!}{d!n!} \geq \max\left\{\left(\frac{d+n}{d}\right)^d, \left(\frac{d+n}{n}\right)^n\right\}$.

Our framework also enables new positive and negative results on the complexity of approximating the intersection of several Archimedean tropical varieties.

Theorem 1.6 *Suppose n is fixed. Then there is a polynomial-time algorithm that, for any input k and $(f_1, \ldots, f_k, w) \in \left(\mathbb{Q}[\sqrt{-1}]\left[x_1^{\pm 1}, \ldots, x_n^{\pm 1}\right]\right)^k \times \mathbb{Q}^n$, outputs the closure of the unique cell σ_w of $\Sigma\left(\bigcup_{i=1}^k \mathrm{ArchTrop}(f_i)\right)$ containing w, described as an explicit intersection of half-spaces. (In particular, whether w lies in $\bigcap_{i=1}^k \mathrm{ArchTrop}(f_i)$ is decided as well.) However, if n is allowed to vary, then deciding whether σ_w has a vertex in $\bigcap_{i=1}^{n} \mathrm{ArchTrop}(f_i)$ is* **NP***-hard.*

Theorem 1.6 is proved in Sect. 6. We will see in Sect. 3 how the first assertion of Theorem 1.6 is useful for finding special start-points for *Newton Iteration* and *Homotopy Continuation* that sometimes enable the approximation of just the roots with norm vector near $(e^{w_1}, \ldots, e^{w_n})$. The final assertion of Theorem 1.6 can be considered as a refined tropical analogue to a classical algebraic complexity result: Deciding whether an arbitrary input system of polynomials equations (with integer coefficients) has a complex root is **NP**-hard. (There are standard reductions from known **NP**-complete problems, such as integer programming or Boolean satisfiability, to complex root detection [21, 52].)

On the practical side, we point out that the algorithms underlying Theorems 1.4–1.6 are quite easily implementable. (A preliminary `Matlab` implementation of our algorithms is available upon request.) Initial experiments indicate that a large-scale implementation could be a worthwhile companion to existing polynomial system solving software.

Before moving on to the necessary technical background, let us first clarify our underlying input size and point out some historical context.

Definition 1.7 We define the *input size* of an integer c to be $\mathrm{size}(c) := \log(2 + |c|)$ and, for $p, q \in \mathbb{Z}$ relative prime with $|q| \geq 2$, $\mathrm{size}(p/q) := \mathrm{size}(p) + \mathrm{size}(q)$. Given a polynomial $f \in \mathbb{Q}[x_1, \ldots, x_n]$, written $f(x) = \sum_{i=1}^t c_i x^{a_i}$, we then define $\mathrm{size}(f)$ to be $\sum_{i=1}^t \left(\mathrm{size}(c_i) + \sum_{j=1}^n \mathrm{size}(a_{i,j})\right)$, where $a_i = (a_{i,1}, \ldots, a_{i,n})$ for all i. Similarly, we define the *input size of a point* $(v_1, \ldots, v_n) \in \mathbb{Q}^n$ as $\sum_{i=1}^n \mathrm{size}(v_i)$. Considering real and imaginary parts, and summing the respect sizes, we then extend the definition of input size further still to polynomials in $\mathbb{Q}\left[\sqrt{-1}\right][x_1, \ldots, x_n]$. Finally, for any system of polynomials $F := (f_1, \ldots, f_k)$, we set $\mathrm{size}(F) := \sum_{i=1}^k \mathrm{size}(f_i)$. $\diamond$

Note in particular that the size of an input in Theorem 1.6 is $\mathrm{size}(w) + \sum_{i=1}^k \mathrm{size}(f_i)$.

Remark 1.8 The reader may wonder why we have not considered the *phases* of the root coordinates and focussed just on norms. The phase analogue of an amoeba is the *co-amoeba*, which has only recently been studied [30, 46, 48]. While it is known that the phases of the coordinates of the roots of polynomial systems satisfy certain equidistribution laws (see, e.g., [35, Thm. 1 (pp. 82–83), Thm. 2 (pp. 87–88), and Cor. 3′ (p. 88)] and [2]), there does not yet appear to be a phase analogue of $\mathrm{ArchTrop}(f)$. Nevertheless, we will see in Sect. 3 that our techniques sometimes allow us to approximate not just norms of root coordinates but roots in full. ⋄

Historical Notes *Using convex and/or piecewise-linear geometry to understand solutions of algebraic equations can be traced back to work of Newton (on power series expansions for algebraic functions) around 1676 [44].*

More recently, tropical geometry [6, 17, 32, 38, 39] has emerged as a rich framework for reducing deep questions in algebraic geometry to more tractable questions in polyhedral and piecewise-linear geometry. For instance, Gelfand, Kapranov, and Zelevinsky first observed the combinatorial structure of amoebae around 1994 [22]. ⋄

2 Background

2.1 Convex, Piecewise-Linear, and Tropical Geometric Notions

Let us first recall the origin of the phrase "tropical geometry", according to [51]: the *tropical semifield* $\mathbb{R}_{\mathrm{trop}}$ is the set $\mathbb{R} \cup \{-\infty\}$, endowed with the operations $x \odot y := x + y$ and $x \oplus y := \max\{x, y\}$. The adjective "tropical" was coined by French computer scientists, in honor of Brazilian computer scientist Imre Simon, who did pioneering work with algebraic structures involving $\mathbb{R}_{\mathrm{trop}}$. Just as algebraic geometry relates geometric properties of zero sets of polynomials to the structure of ideals in commutative rings, tropical geometry relates the geometric properties of certain polyhedral complexes (see Definition 2.7 below) to the structure of ideals in $\mathbb{R}_{\mathrm{trop}}$.

Here we work with a particular kind of tropical variety that, thanks to Theorem 1.3, approximates $\mathrm{Amoeba}(f)$ quite well. The binomial case is quite instructive.

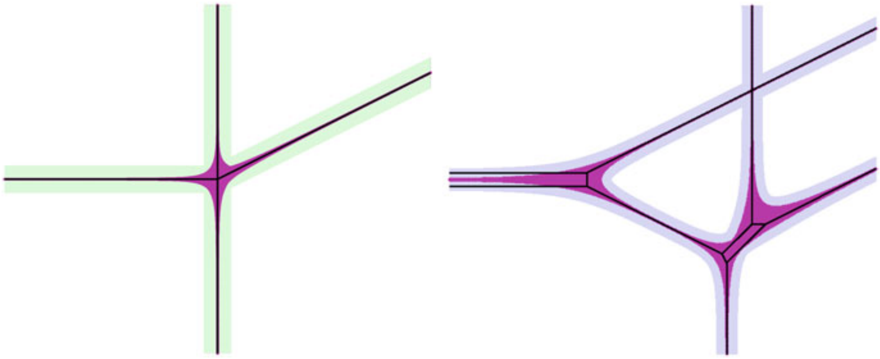
Proposition 2.1 *For any $a \in \mathbb{Z}^n$ and non-zero complex c_1 and c_2, we have*

$$\mathrm{Amoeba}(c_1 + c_2x^a) = \mathrm{ArchTrop}(c_1 + c_2x^a) = \{w \in \mathbb{R}^n \mid a \cdot w = \log|c_1/c_2|\}.$$

Proof If $c_1 + c_2x^a = 0$ then $|c_2x^a| = |c_1|$. We then obtain $a \cdot w = \log|c_1/c_2|$ upon taking logs and setting $w = \mathrm{Log}|x|$. Conversely, for any w satisfying $a \cdot w = \log|c_1/c_2|$, note that $x = e^{w+\theta\sqrt{-1}}$, with $a \cdot \theta$ the imaginary part of $-c_1/c_2$, satisfies $c_1 + c_2x^a = 0$. This proves that $\mathrm{Amoeba}(c_1 + c_2x^a)$ is exactly the stated affine hyperplane. Similarly, since the definition of $\mathrm{ArchTrop}(c_1 + c_2x^a)$ implies that we seek w with $|c_2 e^{a\cdot w}| = |c_1|$, we see that $\mathrm{ArchTrop}(c_1 + c_2 x^a)$ defines the same hyperplane. ■

While $\mathrm{ArchTrop}(f)$ and $\mathrm{Amoeba}(f)$ are always metrically close, $\mathrm{ArchTrop}(f)$ need not even have the same homotopy type as $\mathrm{Amoeba}(f)$ in general.

Example 2.2



Letting $f := 1 + x_2^2 + x_2^4 + x_1x_2^2 + x_1x_2^4 + x_1^2x_2 + x_1^2x_2^2 + x_1^3$ and $g := 0.1 + 0.2x_2^2 + 0.1x_2^4 + 10x_1x_2^2 + 0.001x_1x_2^4 + 0.01x_1^2x_2 + 0.1x_1^2x_2^2 + 0.000005x_1^3$ we obtain the amoebae and tropical varieties (and more lightly shaded neighborhoods), restricted to $[-11, 11] \times [-9, 9]$, respectively drawn on the left and right above. The outermost shape in the left-hand (resp. right-hand) illustration is a neighborhood of $\mathrm{ArchTrop}(f)$ (resp. $\mathrm{Amoeba}(g)$).

It turns out that every point of $\mathrm{Amoeba}(f)$ (resp. $\mathrm{ArchTrop}(g)$) lies well within a distance of 0.65 (resp. 0.49) of some point of $\mathrm{ArchTrop}(f)$ (resp. $\mathrm{Amoeba}(g)$), safely within the distance $\log 7 < 1.946$ (resp. $13 \log 7 < 25.3$) guaranteed by the second (resp. first) bound of Theorem 1.3. Note also that $\mathrm{ArchTrop}(g)$ has two holes while $\mathrm{Amoeba}(g)$ has only a single hole.[1] ◇

Given any f one can naturally construct a convergent sequence of polynomials whose amoebae tend to $\mathrm{ArchTrop}(f)$. This fact can be found in earlier papers of Viro and Mikhalkin, e.g., [41, 65]. However, employing Theorem 1.3 here, we can give a 5-line proof.

Theorem 2.3 *For any n-variate t-nomial f written $\sum_{i=1}^{t} c_i x^{a_i}$, and $s > 0$, define $f^{*s}(x) := \sum_{i=1}^{t} c_i^s x^{a_i}$. Then $\Delta\left(\frac{1}{s}\mathrm{Amoeba}(f^{*s}), \mathrm{ArchTrop}(f)\right) \to 0$ as $s \to +\infty$.*

Proof By Theorem 1.3, $\Delta(\mathrm{Amoeba}(f^{*s}), \mathrm{ArchTrop}(f^{*s})) \le (2t-3)\log(t-1)$ for all $s > 0$. Since $|c_i e^{a_i \cdot w}| \ge |c_j e^{a_j \cdot w}| \iff |c_i e^{a_i \cdot w}|^s \ge |c_j e^{a_j \cdot w}|^s$, we immediately obtain that $\mathrm{ArchTrop}(f^{*s}) = s\mathrm{ArchTrop}(f)$. So then

[1] A *hole* of a subset $S \subseteq \mathbb{R}^n$ is simply a bounded connected component of the complement $\mathbb{R}^n \setminus S$.

$\Delta(\text{Amoeba}(f^{*s}), \text{ArchTrop}(f^{*s})) = s\Delta\left(\frac{1}{s}\text{Amoeba}(f^{*s}), \text{ArchTrop}(f)\right)$ and thus $\Delta\left(\frac{1}{s}\text{Amoeba}(f^{*s}), \text{ArchTrop}(f)\right) \leq \frac{(2t-3)\log(t-1)}{s}$ for all $s > 0$. ■

To more easily link ArchTrop(f) with polyhedral geometry we will need two variations of the classical Newton polygon. First, let Conv(S) denote the convex hull of[2] $S \subseteq \mathbb{R}^n$, $\mathbf{O} := (0, \ldots, 0)$, and $[N] := \{1, \ldots, N\}$. Recall that a *polytope* is the convex hull of a finite point set, a *(closed) half-space* is any set of the form $\{w \in \mathbb{R}^n \mid a \cdot w \leq b\}$ (for some $b \in \mathbb{R}$ and $a \in \mathbb{R}^n \setminus \{\mathbf{O}\}$), and a *(closed) polyhedron* is any intersection of finitely many (closed) half-spaces. Polytopes are exactly polyhedra that are bounded [27, 66]. The two resulting representations of polytopes—$\mathscr{V}$*-presentation* (the convex hull of a finite point set) and $\mathscr{H}$*-presentation* (an intersection of finitely many half-spaces)—are equivalent, but can be exponentially different from an algorithmic point of view. See, e.g., [24, 25].

Definition 2.4 Given any n-variate t-nomial f written $\sum_{i=1}^{t} c_i x^{a_i}$, we define its *(ordinary) Newton polytope* to be $\text{Newt}(f) := \text{Conv}\left(\{a_i\}_{i\in[t]}\right)$, and the *Archimedean Newton polytope of f* to be $\text{ArchNewt}(f) := \text{Conv}\left(\{(a_i, -\log|c_i|)\}_{i\in[t]}\right)$. Also, for any polyhedron $P \subset \mathbb{R}^N$ and $v \in \mathbb{R}^N$, a *face* of P is any set of the form $P_v := \{x \in P \mid v \cdot x \text{ is maximized}\}$. We call v an *outer normal* of P_v. The *dimension* of P, written $\dim P$, is simply the dimension of the smallest affine linear subspace containing P. Faces of P of dimension 0, 1, and $\dim P - 1$ are respectively called *vertices*, *edges*, and *facets*. (P and $\emptyset$ are called *improper faces* of P, and we set $\dim \emptyset := -1$.) Finally, we call any face of P *lower* if and only if it has an outer normal $(w_1, \ldots, w_N)$ with $w_N < 0$, and we let the *lower hull* of ArchNewt(f) be the union of the lower faces of ArchNewt(f). ◇

The outer normals of a k-dimensional face of an n-dimensional polyhedron P form the relative interior of an $(n-k)$-dimensional polyhedron called an *outer normal cone*. Note that ArchNewt(f) usually has dimension 1 greater than that of Newt(f). ArchNewt(f) enables us to relate ArchTrop(f) to linear optimization.

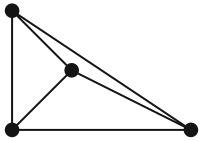
Proposition 2.5 *For any n-variate t-nomial f,* ArchTrop(f) *can also be defined as the set of all $w \in \mathbb{R}^n$ with* $\max\limits_{x \in \text{ArchNewt}(f)} \{x \cdot (w, -1)\}$ *attained on a positive-dimensional face of* ArchNewt(f).

Proof The quantity $|c_i e^{a_i \cdot w}|$ attaining its maximum for at least two indices i is equivalent to the linear form with coefficients $(w, -1)$ attaining its maximimum for at least two different points in $\{(a_i, -\log|c_i|)\}_{i\in[t]}$. Since a face of a polytope is positive-dimensional if and only if it has at least two vertices, we are done. ■

Example 2.6 The Newton polytope of our first example, $f = 1 + x_1^3 + x_2^2 - 3x_1x_2$, is simply the convex hull of the exponent vectors of the monomial terms: $\text{Conv}(\{(0,0), (3,0), (0,2), (1,1)\})$. For the Archimedean Newton polytope, we take the coefficients into account via an extra coordinate:

[2]That is, smallest convex set containing…

ArchNewt(f) = Conv$(\{(0,0,0),(3,0,0),(0,2,0),\ (1,1,-\log 3)\})$. In particular, Newt$(f)$ is a triangle and ArchNewt(f) is a triangular pyramid with base Newt$(f) \times \{0\}$ and apex lying beneath Newt$(f) \times \{0\}$. Note also that the image of the orthogonal projection of the lower hull of ArchNewt(f) onto $\mathbb{R}^2 \times \{0\}$ naturally induces a triangulation of Newt(f), as illustrated below. $\diamond$



Our last example motivates us to consider more general subdivisions and duality. (An outstanding reference is [15].) Recall that a *k-simplex* is the convex hull of $k+1$ points in $\mathbb{R}^N$ (with $N \geq k+1$) *not* lying in any $(k-1)$-dimensional affine linear subspace of $\mathbb{R}^N$. A *simplex* is then simply a k-simplex for some k.

Definition 2.7 A *polyhedral complex* is a collection of polyhedra $\Sigma = \{\sigma_i\}_i$ such that for all i we have (a) every face of σ_i is in Σ and (b) for all j we have that $\sigma_i \cap \sigma_j$ is a face of both σ_i and σ_j. (We allow improper faces like $\emptyset$, σ_i, and σ_j.) The σ_i are the *cells* of the complex, and the *underlying space of* Σ is $|\Sigma| := \bigcup_i \sigma_i$. In particular, we define Σ(ArchTrop(f)) to be the complex whose cells are exactly the (possibly improper) faces of the closures of the connected components of $\mathbb{R}^n \setminus$ ArchTrop(f).

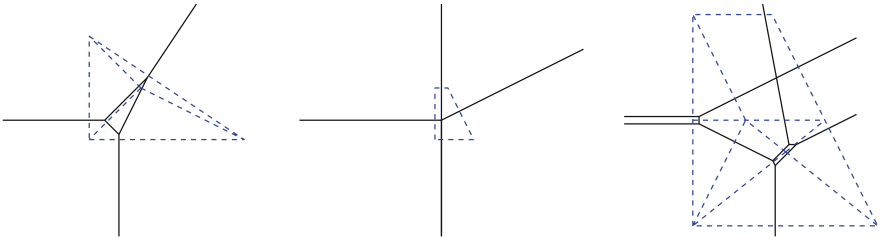
A *polyhedral subdivision* of a polyhedron P is then simply a polyhedral complex $\Sigma = \{\sigma_i\}_i$ with $|\Sigma| = P$. We call Σ a *triangulation* if and only if every σ_i is a simplex. Given any finite subset $A \subset \mathbb{R}^n$, a *polyhedral subdivision induced by* A is then just a polyhedral subdivision of Conv(A) where the vertices of all the σ_i lie in A. Finally, the *polyhedral subdivision of* Newt(f) *induced by* ArchNewt(f), denoted Σ_f, is simply the polyhedral subdivision whose cells are $\{\pi(Q) \mid Q$ is a lower face of ArchNewt$(f)\}$, where $\pi : \mathbb{R}^{n+1} \longrightarrow \mathbb{R}^n$ denotes the orthogonal projection forgetting the last coordinate. $\diamond$

Recall that a *(polyhedral) cone* is just the set of all nonnegative linear combinations of a finite set of points. Such cones are easily seen to always be polyhedra [27, 66].

Example 2.8 The illustration from Example 2.6 above shows a triangulation of the point set $\{(0,0),(3,0),(0,2),(1,1)\}$ which happens to be Σ_f for $f = 1 + x_1^3 + x_2^2 - 3x_1x_2$. More to the point, it is easily checked that the outer normals to a face of dimension k of ArchNewt(f) form a cone of dimension $3-k$. In this way, thanks to the natural partial ordering of cells in any polyhedral complex by inclusion, we get an order-reversing bijection between the cells of Σ_f and pieces of ArchTrop(f). $\diamond$

That ArchTrop(f) is always a polyhedral complex follows directly from Proposition 2.5 above. Proposition 2.5 also implies an order-reversing bijection between the cells Σ_f and the cells of $\Sigma(\text{ArchTrop}(f))$—an incarnation of *polyhedral duality* [66].

Example 2.9 Below we illustrate the aforementioned order-reversing bijection of cells through our first three tropical varieties, and corresponding subdivisions Σ_f of Newt(f):



Note that the vertices of $\Sigma(\text{ArchTrop}(f))$ correspond bijectively to the two-dimensional cells of Σ_f, and the one-dimensional cells of $\Sigma(\text{ArchTrop}(f))$ correspond bijectively to the edges of Σ_f. (In particular, the rays of $\Sigma(\text{ArchTrop}(f))$ are perpendicular to the edges of Newt(f).) Note also that the vertices of Σ_f correspond bijectively to connected components of the *complement* $\mathbb{R}^2\setminus\text{ArchTrop}(f)$. ⋄

2.2 The Complexity of Linear Programming

Let us first point out that [3, 21, 49, 59] are excellent references for further background on the classical Turing model and **NP**-completeness. The results on the complexity of linear optimization we'll use are covered at a more leisurely pace in standard monographs such as [26, 53]. See also [23].

Definition 2.10 Given any matrix $M \in \mathbb{Q}^{k\times N}$ with $i^{\underline{\text{th}}}$ row m_i, and $b := (b_1, \ldots, b_k)^\top \in \mathbb{Q}^k$, the notation $Mx \leq b$ means that $m_1 \cdot x \leq b_1, \ldots, m_k \cdot x \leq b_k$ all hold. Given any $c = (c_1, \ldots, c_N) \in \mathbb{Q}^N$ we then define the *(natural form) linear optimization problem* $\mathscr{L}(M, b, c)$ to be the following: Maximize $c \cdot x$ subject to $Mx \leq b$ and $x \in \mathbb{R}^N$. We also define $\text{size}(\mathscr{L}(M, b, c)) := \text{size}(M) + \text{size}(b) + \text{size}(c)$ (see Definition 1.7). The set of all $x \in \mathbb{R}^N$ satisfying $Mx \leq b$ is the *feasible region* of $\mathscr{L}(M, b, c)$, and when it is empty we call $\mathscr{L}(M, b, c)$ *infeasible*. Finally, if $\mathscr{L}(M, b, c)$ is feasible but does not admit a well-defined maximum, then we call $\mathscr{L}(M, b, c)$ *unbounded*. ⋄

Theorem 2.11 *Given any linear optimization problem $\mathscr{L}(M, b, c)$ as defined above, we can decide infeasibility, unboundedness, or (if $\mathscr{L}(M, b, c)$ is feasible, with bounded maximum) find an optimal solution x^*, all within time polynomial in* size($\mathscr{L}(M,b,c)$). *In particular, if* $\mathscr{L}(M,b,c)$ *is feasible, with bounded maximum, then we can find an optimal solution* x^* *of size polynomial in* size($\mathscr{L}(M,b,c)$). ■

Theorem 2.11 goes back to work of Khachiyan in the late 1970s on the *Ellipsoid Method* [34], building upon earlier work of Shor, Yudin, and Nemirovskii.

For simplicity, we will not focus on the best current complexity bounds, since our immediate goal is to efficiently prove polynomiality for our algorithms. We will need one last complexity result from linear optimization: Recall that a constraint $m_i \cdot x \leq b_i$ of $Mx \leq b$ is called *redundant* if and only if the corresponding row of M, and corresponding entry of b, can be deleted from the pair (M,b) without affecting the feasible region $\{x \in \mathbb{R}^N \mid Mx \leq b\}$.

Lemma 2.12 *Given any system of linear inequalities* $Mx \leq b$ *we can, in time polynomial in* size(M) + size(b)*, find a submatrix* M' *of* M*, and a subvector* b' *of* b*, such that* $\{x \in \mathbb{R}^N \mid M'x \leq b'\} = \{x \in \mathbb{R}^N \mid M'x \leq b'\}$ *and* $M'x \leq b'$ *has no redundant constraints.* ■

The new set of inequalities $M'x \leq b'$ is called an *irredundant representation* of $Mx \leq b$, and can easily be found by solving $\leq k$ linear optimization problems of size no larger than size($\mathscr{L}(M,b,\mathbf{O})$) (see, e.g., [53]).

The linear optimization problems we ultimately solve will have *irrational "right-hand sides"*: Our b will usually have entries that are (rational) linear combination of logarithms of integers. As is well-known in Diophantine Approximation [5], it is far from trivial to efficiently decide the sign of such irrational numbers. This problem is equivalent to deciding inequalities of the form $\alpha_1^{\beta_1} \cdots \alpha_N^{\beta_N} > 1$, where the α_i and β_i are integers. Note, in particular, that while the number of arithmetic operations necessary to decide such an inequality is easily seen to be $O((\sum_{i=1}^N \log|\beta_i|)^2)$ (via the classical binary method of exponentiation), taking bit-operations into account naively results in a problem that appears to have complexity *exponential* in $\log|\beta_1| + \cdots + \log|\beta_N|$. But we can in fact go much faster…

2.3 Irrational Linear Optimization and Approximating Logarithms

Recall the following result on comparing monomials in rational numbers.

Theorem 2.13 ([11, Sec. 2.4]) *Suppose* $\alpha_1, \ldots, \alpha_N \in \mathbb{Q}$ *are positive and* $\beta_1, \ldots, \beta_N \in \mathbb{Z}$*. Also let* A *be the maximum of the numerators and denominators of the* α_i *(when written in lowest terms) and* $B := \max_i\{|\beta_i|\}$*. Then, within*

$$O\left(N 30^N \log(B)(\log\log B)^2 \log\log\log(B)(\log(A)(\log\log A)^2 \log\log\log A)^N\right)$$

bit operations, we can determine the sign of $\alpha_1^{\beta_1} \cdots \alpha_N^{\beta_N} - 1$. ■

While the underlying algorithm is a simple application of Arithmetic-Geometric Mean Iteration (see, e.g., [9]), its complexity bound hinges on a deep estimate of Nesterenko [43], which in turn refines seminal work of Matveev [40] and Alan Baker [5] on linear forms in logarithms.

Definition 2.14 We call a polyhedron P *ℓ-rational* if and only if it is of the form $\{x \in \mathbb{R}^n \mid Mx \leq b\}$ with $M \in \mathbb{Q}^{k\times n}$ and $b = (b_1, \ldots, b_k)^\top$ satisfying

$$b_i = \beta_{1,i} \log|\alpha_1| + \cdots + \beta_{k,i} \log|\alpha_k|,$$

with $\beta_{i,j}, \alpha_j \in \mathbb{Q}$ for all i and j. Finally, we set

$$\mathrm{size}(P) := \mathrm{size}(M) + \mathrm{size}([\beta_{i,j}]) + \sum_{i=1}^{k} \mathrm{size}(\alpha_i). \diamond$$

Via the Simplex Method (or even a brute force search through all n-tuples of facets of P) we can obtain the following consequence of Theorems 2.11 and 2.13.

Corollary 2.15 *Following the notation of Definition 2.14, suppose n is fixed. Then we can decide whether P is empty, compute an irredundant representation for P, and enumerate all maximal sets of facets determining vertices of P, in time polynomial in* $\mathrm{size}(P)$. ■

The key trick behind the proof of Corollary 2.15 is that the intermediate linear optimization problems needed to find an irredundant representation for P use linear combinations (of rows of the original representation) with coefficients of moderate size (see, e.g., [53]).

3 Tropical Start-Points for Numerical Iteration and an Example

We begin by outlining a method for picking start-points for Newton Iteration (see, e.g., [12, Ch. 8] for a modern perspective) and Homotopy Continuation [7, 31, 37, 62, 64]. While we do not discuss these methods for solving polynomial equations in further detail, let us at least point out that Homotopy Continuation (combined with Smale's α-Theory for certifying roots [7,12]) is currently the fastest, most easily parallelizable, and reliable method for numerically solving polynomial systems in complete generality. Other important methods include Resultants [18] and Gröbner Bases [20]. While these alternative methods are of great utility in certain algebraic and theoretical applications [1, 19], Homotopy Continuation is currently the method of choice for practical numerical computation with extremely large polynomial systems.

Algorithm 3.1 (Coarse Approximation to Roots with Log-Norm Vector Near a Given Query Point)

INPUT. *Polynomials* $f_1, \ldots, f_n \in \mathbb{Q}\left[\sqrt{-1}\right]\left[x_1^{\pm 1}, \ldots, x_n^{\pm 1}\right]$, *with* $f_i(x) = \sum_{j=1}^{t_i} c_{i,j} x^{a_j(i)}$ *an n-variate* t_i*-nomial for all* i*, and a query point* $w \in \mathbb{Q}^n$.

OUTPUT. *An ordered n-tuple of sets of indices* $(J_i)_{i=1}^n$ *such that, for all* i, $g_i := \sum_{j \in J_i} c_{i,j} x^{a_j(i)}$ *is a sub-summand of* f_i*, and the roots of* $G := (g_1, \ldots, g_n)$ *are approximations of the roots of* $F := (f_1, \ldots, f_n)$ *with log-norm vector nearest* w.

DESCRIPTION.

1. *Let* σ_w *be the closure of the unique cell of* $\Sigma(\bigcup_{i=1}^n \mathrm{ArchTrop}(f_i))$ *(see Definition 2.7) containing* w.
2. *If* σ_w *has no vertices in* $\bigcap_{i=1}^n \mathrm{ArchTrop}(f_i)$ *then output an irredundant collection of facet inequalities for* σ_w*, output* "`There are no roots of F in` σ_w `.`"*, and* STOP.
3. *Otherwise, fix any vertex* v *of* $\sigma_w \cap \bigcap_{i=1}^n \mathrm{ArchTrop}(f_i)$ *and, for each* $i \in [n]$*, let* E_i *be any edge of* $\mathrm{ArchNewt}(f_i)$ *generating a facet of* $\mathrm{ArchTrop}(f_i)$ *containing* v.
4. *For all* $i \in [n]$*, let* $J_i := \{j \mid (a_j(i), -\log|c_{i,j}|) \in E_i\}$.
5. *Output* $(J_i)_{i=1}^n$. ■

Thanks to our main results and our preceding observations on linear optimization, we can easily obtain that our preceding algorithm has complexity polynomial in size(F) for fixed n. In particular, Step 1 is (resp. Steps 2 and 3 are) accomplished via the algorithm underlying Theorem 1.5 (resp. Corollary 2.15).

The key subtlety then is to prove that, for most inputs, our algorithm actually gives useful approximations to the roots with log-norm vector nearest the input query point w, or truthfully states that there are no root log-norm vectors in σ_w. We leave the precise metric estimates defining "most inputs" for future work. However, we point out that a key ingredient is the $\mathscr{A}$-discriminant [22], and a recent polyhedral approximation of its amoeba [50] refining the *tropical discriminant* [16]. So we will now clarify the meaning of the output of our algorithm.

The output system G is useful because, with high probability (in the sense of *random liftings*, as in [18, Lemma 6.2]), all the g_i are *binomials*, and binomial systems are particularly easy to solve: They are equivalent to linear equations in the logarithms of the original variables. In particular, any $n \times n$ binomial system always has a *unique* vector of norms for its roots.

Recall the standard notation $\mathrm{Jac}(F) := \left[\frac{\partial f_i}{\partial x_j}\right]_{n \times n}$. The connection to Newton Iteration is then easy to state: Use any root of G as a start-point $z(0)$ for the iteration $z(n+1) := z(n) - \mathrm{Jac}(F)^{-1}|_{z(n)} F(z(n))$. The connection to Homotopy Continuation is also simple: Use the pair (G, ζ) (for any root ζ of G) to start a path converging (under the usual numerical conditioning assumptions on whatever predictor-corrector method one is using) to a root of F with log-norm vector near w. Note also that while it is safer to do the extra work of Homotopy Continuation, there will be cases where the tropical start-points from Algorithm 3.1 are sufficiently good for mere Newton Iteration to converge quickly to a true root.

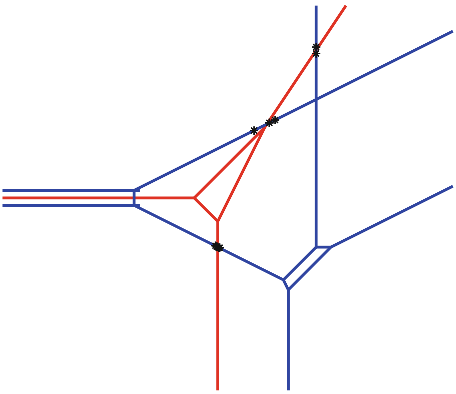
Remark 3.2 Note that, when applying Algorithm 3.1 for later Homotopy Continuation, we have the freedom to follow as few start-points, or as few paths, as we want. When our start-points (resp. paths) indeed converge to nearby roots, we obtain a tremendous savings over having to follow *all* start-points (resp. paths). ⋄

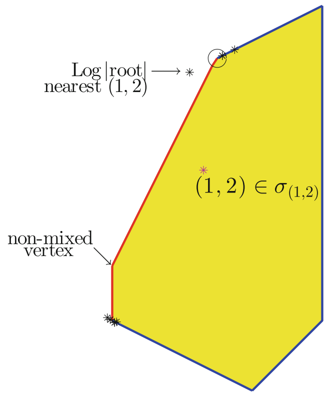
Definition 3.3 Given any n-dimensional polyhedra $P_1, \ldots, P_n \subset \mathbb{R}^n$, we call a vertex v of $\bigcap_{i=1}^n P_i$ *mixed* if and only if v lies on a facet of P_i for all i. ⋄

Note that, by construction, any vertex chosen in Step 3 of Algorithm 3.1 is mixed.

Example 3.4 Let us make a 2×2 polynomial system out of our first and third examples:

$$\begin{aligned} f_1 &:= 1 + x_1^3 + x_2^2 - 3x_1x_2 \\ f_2 &:= 0.1 + 0.2x_2^2 + 0.1x_2^4 + 10x_1x_2^2 + 0.001x_1x_2^4 \\ &\quad +0.01x_1^2x_2 + 0.1x_1^2x_2^2 + 0.000005x_1^3 \end{aligned}$$





The system $F := (f_1, f_2)$ has exactly 12 roots in $(\mathbb{C}^*)^2$, the coordinate-wise log-norms of which form the small clusters near certain intersections of ArchTrop(f_1) and ArchTrop(f_2) shown on the left illustration above. In particular, $\sigma_{(2,1)}$ is the heptagonal cell[3] magnified on the right of the figure above, and has exactly 2 vertices that are mixed. (The other 5 vertices of $\sigma_{(1,2)}$ are vertices of ArchTrop(f_i) lying in the interior of a two-dimensional cell of Σ(ArchTrop(f_{3-i})) for $i \in \{1, 2\}$.)

Applying Algorithm 3.1 we then have two possible outputs, depending on which mixed vertex of $\sigma_{(1,2)}$ we pick. The output corresponding to the circled vertex is the pair of index sets $(\{2, 3\}, \{3, 4\})$. More concretely, Algorithm 3.1 alleges that

[3]The cell looks hexagonal because it has a pair of vertices too close to distinguish visually.

the system $G := (g_1, g_2) := (x_1^3 + x_2^2, 0.1x_2^4 + 10x_1x_2^2)$ has roots with log-norm vector near a log-norm vector of a root of F that is in turn close to w. Indeed, the sole log-norm vector coming from the roots of G is $\left(\log 10, \frac{3}{2}\log 10\right)$ and the roots themselves are $\{(\pm 10, \sqrt{\mp 1000})\}$ (with both values of the square root allowed). All 4 roots in fact converge (under Newton iteration, with no need for Homotopy Continuation) to true roots of F. In particular, the root $(-10, \sqrt{1000})$ (resp. $(-10, -\sqrt{1000})$) converges to the root of F with closest (resp. third closest) log-norm vector to w. The other two roots of G converge to a conjugate pair of roots of F with log-norm vector $(2.4139, 3.5103)$ (to four decimal places) lying in the small circle in the illustration. ⋄

Remark 3.5 While we have relied upon Diophantine approximation and subtle aspects of the Simplex Method to prove our bit-complexity bounds in Theorems 1.4–1.6, one can certainly be more flexible when using Algorithm 3.1 in practical floating-point computations. For instance, heuristically, it appears that one can get away with less accuracy than stipulated by Theorem 2.13 when comparing linear combinations of logarithms. Similarly, one should feel free to use the fastest (but still reliably accurate) algorithms for linear optimization when applying our methods to large-scale polynomial systems. (See, e.g., [63].) ⋄

4 Proof of Theorem 1.5

Using $t-1$ comparisons, we can isolate all indices i such that $\max_i |c_i e^{a_i \cdot w}|$ is attained. Thanks to Theorem 2.13 this can be done in polynomial-time. We then obtain, say, J equations of the form $a_i \cdot w = -\log|c_i|$ and K inequalities of the form $a_i \cdot w > -\log|c_i|$ or $a_i \cdot w < -\log|c_i|$.

Thanks to Lemma 2.12, combined with Corollary 2.15, we can determine the exact cell of $\mathrm{ArchTrop}(f)$ containing w if $J \geq 2$. Otherwise, we obtain the unique cell of $\mathbb{R}^n \setminus \mathrm{ArchTrop}(f)$ with relative interior containing w. Note also that an $(n-1)$-dimensional face of either kind of cell must be the dual of an edge of $\mathrm{ArchNewt}(f)$. Since every edge has exactly 2 vertices, there are at most $t(t-1)/2$ such $(n-1)$-dimensional faces, and thus σ_w is the intersection of at most $t(t-1)/2$ half-spaces. So we are done. ■

Remark 4.1 Theorem 1.5 also generalizes an earlier complexity bound from [4] for deciding membership in $\mathrm{ArchTrop}(f)$. ⋄

5 Proof of Theorem 1.4

Note *The hurried reader will more quickly grasp the following proof after briefly reviewing Theorems 1.3, 1.5, and 2.13.*

Since ArchTrop(f) and Amoeba(f) are closed and non-empty, $\inf_{v\in\text{ArchTrop}(f)} |v-w| = |w-v'|$ for some point $v'\in\text{ArchTrop}(f)$ and $\inf_{u\in\text{Amoeba}(f)} |u-w| = |w-u'|$ for some point $u'\in\text{Amoeba}(f)$.

Now, by the second upper bound of Theorem 1.3, there is a point $v''\in$ ArchTrop(f) within distance $\log(t-1)$ of u'. Clearly, $|w-v'|\leq|w-v''|$. Also, by the Triangle Inequality, $|w-v''|\leq|w-u'|+|u'-v''|$. So then,

$$\inf_{v\in\text{ArchTrop}(f)} |v-w|\leq \inf_{u\in\text{Amoeba}(f)} |u-w|+\log(t-1),$$

and thus $\inf_{u\in\text{Amoeba}(f)} |u-w|-\inf_{v\in\text{ArchTrop}(f)} |v-w|\geq-\log(t-1)$.

Similarly, by the first upper bound of Theorem 1.3, there is a point $u''\in$ Amoeba(f) within distance $(2t-3)\log(t-1)$ of v'. Clearly, $|w-u'|\leq|w-u''|$. Also, by the Triangle Inequality, $|w-u''|\leq|w-v'|+|v'-u''|$. So then, $\inf_{u\in\text{Amoeba}(f)} |u-w|\leq\inf_{v\in\text{ArchTrop}(f)} |v-w|+(2t-3)\log(t-1)$, and thus

$$\inf_{u\in\text{Amoeba}(f)} |u-w|-\inf_{v\in\text{ArchTrop}(f)} |v-w|\leq(2t-3)\log(t-1).$$

So our first assertion is proved.

Now if f has coefficients with real and imaginary parts that are rational, and n is fixed, Theorem 1.5 (which we've already proved) tells us that we can decide whether w lies in ArchTrop(f) using a number of bit operations polynomial in size(w) + size(f). So we may assume $w\not\in\text{ArchTrop}(f)$ and $\dim\sigma_w=n$.

Theorem 1.5 also gives us an explicit description of σ_w as the intersection of a number of half-spaces polynomial in t. Moreover, σ_w is ℓ-rational (recall Definition 2.14), with size polynomial in size(f). So we can compute the distance D from w to ArchTrop(f) by finding which facet of σ_w has minimal distance to w. The distance from w to any such facet can be approximated to the necessary number of bits in polynomial-time via Theorem 2.13 and the classical formula for distance between a point and an affine hyperplane: $\inf_{u\in\{x\;|\;r\cdot x=s\}} |u-w| = (|r\cdot w|-\text{sign}(r\cdot w)s)/|r|$. More precisely, comparing the facet distances reduces to checking the sign of an expression of the form $\gamma_1+\gamma_2\log\left(\frac{c_i}{c_{i'}}\right)+\gamma_3\log\left(\frac{c_j}{c_{j'}}\right)$ where γ_1 (resp. γ_2, γ_3) is a rational linear combination of $\sqrt{|a_i-a_{i'}|}$ and $\sqrt{|a_j-a_{j'}|}$ (resp. rational multiple of $\sqrt{|a_i-a_{i'}|}$ or $\sqrt{|a_j-a_{j'}|}$), with coefficients of size polynomial in size(f), for some indices $i,i',j,j'\in[t]$. We can then efficiently approximate D by approximating the underlying square-roots and logarithms to sufficient precision. The latter can be accomplished by Arithmetic-Geometric Iteration, as detailed in [9], and the amount of precision needed is explicitly bounded by an earlier variant of Theorem 2.13 covering inhomogeneous linear combinations of logarithms of algebraic numbers with algebraic coefficients [5]. The resulting bounds are somewhat worse than in Theorem 2.13, but still allow us to find polynomially many leading bits of $\inf_{v\in\text{Amoeba}(f)} |v-w|$ (for $w\in\mathbb{Q}^n$) in time polynomial in size(w) + size(f).

To prove the final assertion, we merely decide whether $\inf_{v\in\text{ArchTrop}(f)} |v - w|$ strictly exceeds $\log(t - 1)$ or not. To do so, we need only compute a polynomial number of leading bits of $\inf_{v\in\text{ArchTrop}(f)} |v - w|$ (thanks to Theorem 2.13), and this takes time polynomial in $\text{size}(w) + \text{size}(f)$. Thanks to our initial observations using the Triangle Inequality, it is clear that Output (b) or Output (a) occurs according as $\inf_{v\in\text{ArchTrop}(f)} |v - w| > \log(t - 1)$ or not. So we are done. ■

6 Proving Theorem 1.6

6.1 Fast Cell Computation: Proof of the First Assertion

First, we apply Theorem 1.5 to (f_i, w) for each $i \in [k]$ to find which $\text{ArchTrop}(f_i)$ contain w.

If w lies in no $\text{ArchTrop}(f_i)$, then we simply use Corollary 2.15 (as in our proof of Theorem 1.5) to find an explicit description of the closure of the cell of $\mathbb{R}^n \setminus \bigcup_{i=1}^k \text{ArchTrop}(f_i)$ containing w. Otherwise, we find the cells of $\text{ArchTrop}(f_i)$ (for those i with $\text{ArchTrop}(f_i)$ containing w) that contain w. Then, applying Corollary 2.15 once again, we explicitly find the unique cell of $\bigcap\limits_{\text{ArchTrop}(f_i)\ni w} \text{ArchTrop}(f_i)$ containing w.

Assume that f_i has exactly t_i monomial terms for all i. In either of the preceding cases, the total number of half-spaces involved is no more than $\sum_{i=1}^k t_i(t_i - 1)/2$. So the overall complexity of our redundancy computations is polynomial in the input size and we are done. ■

6.2 Hardness of Detecting Mixed Vertices: Proving the Second Assertion

It will clarify matters if we consider a related **NP**-hard problem for rational polytopes first.

Ultimately, our proof boils down to a reduction from the following problem, equivalent to the famous **NP**-complete PARTITION problem (see below): Decide if a vertex of the hypercube $[-1, 1]^n$ lies on a prescribed hyperplane defined by an equation of the form $a \cdot x = 0$ with $a \in \mathbb{Z}^n$. Because the coordinates of a are integral, we can replace the preceding equation by the inequality $0 \le a \cdot x \le 1/2$. With a bit more work, we can reduce PARTITION to the detection of a mixed vertex for a particular intersection of polyhedra. We now go over the details.

6.2.1 Preparation over $\mathbb{Q}$

In the notation of Definition 3.3, let us first consider the following decision problem. We assume all polyhedra are given explicitly as finite collections of rational linear inequalities, with size defined as in Sect. 2.2.

MIXED-VERTEX:

Given $n \in \mathbb{N}$ and polyhedra $P_1, \ldots, P_n$ in $\mathbb{R}^n$, does $P := \bigcap_{i=1}^n P_i$ have a mixed vertex? ■

While MIXED-VERTEX can be solved in polynomial time when n is fixed (by a brute-force check over all mixed n-tuples of facets), we will show that, for n varying, the problem is **NP**-complete, even when restricting to the case where all polytopes are full-dimensional and $P_1, \ldots, P_{n-1}$ are axes-parallel bricks.

Let e_i denote the $i^{\underline{\text{th}}}$ standard basis vector in $\mathbb{R}^n$ and let $M^\top$ denote the transpose of a matrix M. Also, given $\alpha \in \mathbb{R}^n$ and $b \in \mathbb{R}$, we will use the following notation for hyperplanes and certain half-spaces in $\mathbb{R}^n$ determined by α and b: $H_{(\alpha,b)} := \{x \in \mathbb{R}^n \mid \alpha \cdot x = b\}$, $H^{\leq}_{(\alpha,b)} := \{x \in \mathbb{R}^n \mid \alpha \cdot x \leq b\}$. For $i \in [n]$, let $s_i \in \mathbb{N}$, $M_i := [m_{i,1}, \ldots, m_{i,s_i}]^\top \in \mathbb{Z}^{s_i \times n}$, $b_i := (b_{i,1}, \ldots, b_{i,s_i})^\top \in \mathbb{Z}^{s_i}$, and $P_i := \{x \in \mathbb{R}^n \mid M_i x \leq b_i\}$. Since linear optimization can be done in polynomial-time (in the cases we consider) we may assume that the presentations $(n, s_i; M_i, b_i)$ are *irredundant*, i.e., P_i has exactly s_i facets if P_i is full-dimensional, and the sets $P_i \cap H_{(m_{i,j}, b_{i,j})}$, for $j \in [s_i]$, are precisely the facets of P_i for all $i \in [n]$.

Now set $P := \bigcap_{i=1}^n P_i$. Note that $\text{size}(P)$ is thus linear in $\sum_{i=1}^n \text{size}(P_i)$.

Lemma 6.1 MIXED-VERTEX $\in$ **NP**.

Proof Since the binary sizes of the coordinates of the vertices of P are bounded by a polynomial in the input size, we can use vectors $v \in \mathbb{Q}^n$ of polynomial size as certificates. We can check in polynomial-time whether such a vector v is a vertex of P. If this is not the case, v cannot be a mixed vertex of P. Otherwise, v is a mixed vertex of P if and only if for each $i \in [n]$ there exists a facet F_i of P_i with $v \in F_i$. Since the facets of the polyhedra P_i admit polynomial-time decriptions as $\mathscr{H}$-polyhedra, this can be checked by a total of $s_1 + \cdots + s_n$ polyhedral membership tests. These membership tests are easily doable in polynomial-time since any of the underlying inequalities can be checked in polynomial-time and the number of faces of any P_i no worse than linear in the size of P_i.

So we can check in polynomial-time whether a given certificate v is a mixed vertex of P. Hence MIXED-VERTEX is in **NP**. ■

Since (in fixed dimension) we can actually list all vertices of P in polynomial-time, it is clear that MIXED-VERTEX can be solved in polynomial-time when n is fixed. When n is allowed to vary we obtain hardness:

Theorem 6.2 MIXED-VERTEX *is* **NP**-*hard, even in the special case where $P_1, \ldots, P_{n-1}$ are centrally symmetric axes-parallel bricks with vertex coordinates in $\{\pm 1, \pm 2\}$, and P_n has at most $2n + 2$ facets (with $2n$ of them parallel to coordinate hyperplanes).*

The proof of Theorem 6.2 will be based on a reduction from the following decision problem:

PARTITION

Given $d, \alpha_1, \ldots, \alpha_d \in \mathbb{N}$, *is there an* $I \subseteq [d]$ *such that* $\sum_{i\in I} \alpha_i = \sum_{i\in[d]\setminus I} \alpha_i$*?* ■

PARTITION was in the original list of **NP**-complete problems from [33].

Let an instance $(d; \alpha_1, \ldots, \alpha_d)$ of PARTITION be given, and set $\alpha := (\alpha_1, \ldots, \alpha_d)$. Then we are looking for a point $x \in \{-1, 1\}^d$ with $\alpha \cdot x = 0$.

We will now construct an equivalent instance of MIXED-VERTEX. With $n := d + 1$, $\bar{x} := (x_1, \ldots, x_{n-1})$ and $\mathbf{1}_n := (1, \ldots, 1) \in \mathbb{R}^n$ let

$$P_i := \left\{(\bar{x}, x_n) | -1 \leq x_i \leq 1, \, -2 \leq x_j \leq 2 \text{ for all } j \in [n] \setminus \{i\}\right\}.$$

Also, for $i \in [n-1]$, let

$$P_n := \{(\bar{x}, x_n) \mid -2 \cdot \mathbf{1}_{n-1} \leq \bar{x} \leq 2 \cdot \mathbf{1}_{n-1}, \, -1 \leq x_n \leq 1, \, 0 \leq 2\alpha \cdot \bar{x} \leq 1\}$$

and set $P := \bigcap_{i=1}^{n} P_i$, $\hat{\alpha} := (\alpha, 0)$.

The next lemma shows that $P_n \cap \{-1, 1\}^n$ still captures the solutions of the given instance of partition.

Lemma 6.3 $(d; \alpha_1, \ldots, \alpha_d)$ *is a "no"-instance of* PARTITION *if and only if* $P_n \cap \{-1, 1\}^n$ *is empty.*

Proof Suppose, first, that $(d; \alpha_1, \ldots, \alpha_d)$ is a "no"-instance of PARTITION. If P_n is empty there is nothing left to prove. So, let $y \in P_n$ and $w \in \{-1, 1\}^{n-1} \times \mathbb{R}$. Since $\alpha \in \mathbb{N}^d$ we have $|\hat{\alpha} \cdot w| \geq 1$. Hence, via the Cauchy-Schwarz inequality, we have $1 \leq |\hat{\alpha} \cdot w| = |\hat{\alpha} \cdot y + \hat{\alpha} \cdot (w - y)| \leq |\hat{\alpha} \cdot y| + |\hat{\alpha} \cdot (w - y)| \leq \frac{1}{2} + |\hat{\alpha}| \cdot |w - y| = \frac{1}{2} + |\alpha| \cdot |w - y|$ and thus $|w - y| \geq \frac{1}{2|\alpha|} > 0$. Therefore $P_n \cap \left(\{-1, 1\}^{n-1} \times \mathbb{R}\right)$ is empty.

Conversely, if $P_n \cap \{-1, 1\}^n$ is empty, then there is no $x \in \{\pm 1\}^{n-1}$ such that $0 \leq \alpha \cdot \bar{x} \leq \frac{1}{2}$. Since $\hat{\alpha} \in \mathbb{N}^{n-1}$, we have that $(d, \alpha_1, \ldots, \alpha_d)$ is a "No"-instance of PARTITION. ■

The next lemma reduces the possible mixed vertices to the vertical edges of the standard cube.

Lemma 6.4 *Following the preceding notation, let* v *be a mixed vertex of* $P := \bigcap_{i=1}^{n} P_i$. *Then* $v \in \{-1, 1\}^{n-1} \times [-1, 1]$.

Proof First note that $Q := \bigcap_{i=1}^{n-1} P_i = [-1, 1]^{n-1} \times [-2, 2]$. Therefore, for each $i \in [n-1]$, the only facets of P_i that meet Q are those in $H_{(e_i, \pm 1)}$ and $H_{(e_n, \pm 2)}$. Since $P \subset [-1, 1]^n$, and for each $i \in [n-1]$ the mixed vertex v must be contained in a facet of P_i, we have $v \in [-1, 1]^n \cap \bigcap_{i=1}^{n-1} \left(\bigcup_{\delta_i \in \{-1,1\}} H_{(e_i, \delta_i)} \right) = \{-1, 1\}^{n-1} \times [-1, 1]$, which proves the assertion. ■

The next lemma adds P_n to consideration.

Lemma 6.5 *Let v be a mixed vertex of $P := \bigcap_{i=1}^n P_i$. Then $v \in \{-1, 1\}^n$.*

Proof By Lemma 6.4, $v \in \{-1, 1\}^{n-1} \times [-1, 1]$. Since the hyperplanes $H_{(e_i,\pm 2)}$ do not meet $[-1, 1]^n$, we have $v \notin H_{(e_i,-2)} \cup H_{(e_i,2)}$ for all $i \in [n-1]$. Hence, v can only be contained in the constraint hyperplanes $H_{(\hat{\alpha},0)}, H_{(2\hat{\alpha},1)}, H_{(e_n,-1)}, H_{(e_n,1)}$. Since $\hat{\alpha} \in \mathbb{R}^{n-1} \times \{0\}$, the vector $\hat{\alpha}$ is linearly dependent on $e_1, \ldots, e_{n-1}$. Hence, $v \in H_{(e_n,-1)} \cup H_{(e_n,1)}$, i.e., $v \in \{-1, 1\}^n$. ■

We can now prove the **NP**-hardness of MIXED-VERTEX.

Proof of Theorem 6.2 First, let $(d; \alpha_1, \ldots, \alpha_d)$ be a "yes"-instance of PARTITION, let $x^* := (\xi_1^*, \ldots, \xi_{n-1}^*) \in \{-1, 1\}^{n-1}$ be a solution, and set $\xi_n^* := 1$, $v := (x^*, \xi_n^*)$, $F_i := H_{(e_i,\xi_i^*)} \cap P_i$ for all $i \in [n]$, and $\hat{F}_n := H_{(\hat{\alpha},0)} \cap P_n$. Then $v \in \hat{F}_n \subset P_n$, hence $v \in P$ and, in fact, v is a vertex of P. Furthermore, F_i is a facet of P_i for all $i \in [n]$, $v \in \bigcap_{i=1}^n F_i$, and thus v is a mixed vertex of P.

Conversely, let $(d; \alpha_1, \ldots, \alpha_d)$ be a "no"-instance of PARTITION, and suppose that $v \in \mathbb{R}^n$ is a mixed vertex of P. By Lemma 6.5, $v \in \{-1, 1\}^n$. Furthermore, v lies in a facet of P_n. Hence, in particular, $v \in P_n$, i.e., $P_n \cap \{-1, 1\}^n$ is non-empty. Therefore, by Lemma 6.3, $(d; \alpha_1, \ldots, \alpha_d)$ is a "yes"-instance of PARTITION. This contradiction shows that P does not have a mixed vertex.

Clearly, the transformation works in polynomial-time. ■

6.3 Proof of the Second Assertion of Theorem 1.6

It clearly suffices to show that the following variant of MIXED-VERTEX is **NP**-hard:
LOGARITHMIC-MIXED-VERTEX:
Given $n \in \mathbb{N}$ and ℓ-rational polyhedra $P_1, \ldots, P_n \subset \mathbb{R}^n$, does $P := \bigcap_{i=1}^n P_i$ have a mixed vertex? ■

Via an argument completely parallel to the last section, the **NP**-hardness of LOGARITHMIC-MIXED-VERTEX follows immediately from the **NP**-hardness of the following variant of PARTITION:
LOGARITHMIC-PARTITION
Given $d \in \mathbb{N}$, $\alpha_1, \ldots, \alpha_d \in \mathbb{N} \setminus \{0\}$, is there an $I \subseteq [d]$ such that $\sum_{i \in I} \log \alpha_i = \sum_{i \in [d] \setminus I} \log \alpha_i$? ■

We measure size in LOGARITHMIC-PARTITION just as in the original PARTITION Problem: $\sum_{i=1}^d \log \alpha_d$. Note that LOGARITHMIC-PARTITION is equivalent to the obvious variant of PARTITION where we ask for a partition making the two resulting *products* be identical. The latter problem is known to be **NP**-hard as well, thanks to [45], and is in fact also strongly **NP**-hard. ■

References

1. D'Andrea, C., Krick, T., Sombra, M.: Heights of varieties in multiprojective spaces and arithmetic Nullstellensatze. Ann. Sci. l'ENS fascicule **4**, 549–627 (2013)
2. D'Andrea, C., Galligo, A., Sombra, M.: Quantitative equidistribution for the solution of a system of sparse polynomial equations. Am. J. Math (to appear)
3. Arora, S., Barak, B.: Computational Complexity. A Modern Approach. Cambridge University Press, Cambridge (2009)
4. Avendaño, M., Kogan, R., Nisse, M., Rojas, J.M.: Metric Estimates and membership complexity for archimedean amoebae and tropical hypersurfaces. Submitted for publication, also available as Math ArXiV preprint `1307.3681`
5. Baker, A.: The theory of linear forms in logarithms. In: Transcendence Theory: Advances and Applications: Proceedings of a Conference Held at the University of Cambridge, Cambridge, January–February 1976. Academic, London (1977)
6. Baker, M., Rumely, R.: Potential Theory and Dynamics on the Berkovich Projective Line. Mathematical Surveys and Monographs, vol. 159. American Mathematical Society, Providence (2010)
7. Bates, D.J., Hauenstein, J.D., Sommese, A.J., Wampler, C.W.: Numerically Solving Polynomial Systems with Bertini. Software, Environments and Tools Series. Society for Industrial and Applied Mathematics (2013)
8. Beltrán, C., Pardo, L.M.: Smale's 17th problem: Average polynomial time to compute affine and projective solutions. J. Am. Math. Soc. **22**, 363–385 (2009)
9. Bernstein, D.J.: Computing Logarithm Intervals with the Arithmetic-Geometric Mean Iterations. Available from http://cr.yp.to/papers.html
10. Bettale, L., Faugére, J.-C., Perret, L.: Cryptanalysis of HFE, multi-HFE and variants for odd and even characteristic. Designs Codes Cryptogr. **69**(1), 1–52 (2013)
11. Bihan, F., Rojas, J.M., Stella, C.: Faster real feasibility via circuit discriminants. In: Proceedings of ISSAC 2009 (July 28–31, Seoul, Korea), pp. 39–46. ACM (2009)
12. Blum, L., Cucker, F., Shub, M., Smale, S.: Complexity and Real Computation. Springer (1998)
13. Bürgisser, P., Scheiblechner, P.: On the complexity of counting components of algebraic varieties. J. Symb. Comput. **44**(9), 1114–1136 (2009)
14. Bürgisser, P., Cucker, F.: Solving polynomial equations in smoothed polynomial time and a near solution to Smale's 17th problem. In: Proceedings STOC (Symposium on the Theory of Computation) 2010, pp. 503–512. ACM (2010)
15. De Loera, J.A., Rambau, J., Santos, F.: Triangulations, Structures for Algorithms and Applications, Algorithms and Computation in Mathematics, vol. 25. Springer, Berlin (2010)
16. Dickenstein, A., Feichtner, E.M., Sturmfels, B.: Tropical discriminants. J. Am. Math. Soc. **20**(4), 1111–1133 (2007)
17. Einsiedler, M., Kapranov, M., Lind, D.: Non-Archimedean amoebas and tropical varieties. J. die reine Angew. Math. (Crelles J.) **2006**(601), 139–157 (2006)
18. Emiris, I.Z., Canny, J.: Efficient incremental algorithms for the sparse resultant and mixed volume. J. Symb. Comput. **20**(2), 117–149 (1995)
19. Faugère, J.-C., Gaudry, P., Huot, L., Renault, G.: Using symmetries in the index calculus for elliptic curves discrete logarithm. J. Cryptol. 1–40 (2013)
20. Faugère, J.-C., Hering, M., Phan, J.: The membrane inclusions curvature equations. Adv. Appl. Math. **31**(4), 643–658 (2003).
21. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness, A Series of Books in the Mathematical Sciences. W. H. Freeman and Co., San Francisco (1979)
22. Gel'fand, I.M., Kapranov, M.M., Zelevinsky, A.V.: Discriminants, Resultants and Multidimensional Determinants. Birkhäuser, Boston (1994)

23. Gritzmann, P.: Grundlagen der Mathematischen Optimierung: Diskrete Strukturen, Komplexitätstheorie, Konvexitätstheorie, Lineare Optimierung, Simplex-Algorithmus, Dualität. Springer (2013)
24. Gritzmann, P., Klee, V.: On the complexity of some basic problems in computational convexity: I. Containment problems. Discrete Math. **136**, 129–174 (1994)
25. Gritzmann, P., Klee, V.: On the complexity of some basic problems in computational convexity: II. Volume and mixed volumes. In: Bisztriczky, T., McMullen, P., Schneider, R., Ivic Weiss A. (eds) Polytopes: Abstract, Convex and Computational, pp. 373–466. Kluwer, Boston (1994)
26. Grötschel, M., Lovász, L., Schrijver, A.: Geometric Algorithms and Combinatorial Optimization. Springer, New York (1993)
27. Grünbaum, B.: Convex Polytopes. Wiley-Interscience, London (1967); Ziegler, G. (ed) Convex Polytopes, 2nd edn. Graduate Texts in Mathematics, vol. 221. Springer (2003)
28. Hao, W., Hauenstein, J.D., Hu, B., Liu, Y., Sommese, A., Zhang, Y.-T.: Continuation along bifurcation branches for a tumor model with a necrotic core. J. Sci. Comput. **53**, 395–413 (2012)
29. Hauenstein, J., Rodriguez, J., Sturmfels, B.: Maximum likelihood for matrices with rank constraints. J. Algebraic Stat. **5**(1), 18–38 (2014). Also available as Math ArXiV preprint `1210.0198`
30. Herbst, M., Hori, K., Page, D.: Phases of $N = 2$ theories in $1 + 1$ dimensions with boundary. High Energy Phys. ArXiV preprint `0803.2045v1`
31. Huan, L.J., Li, T.-Y.: Parallel homotopy algorithm for symmetric large sparse eigenproblems. J. Comput. Appl. Math. **60**(1–2), 77–100 (1995)
32. Itenberg, I., Mikhalkin, G., Shustin, E.: Tropical Algebraic Geometry, 2nd edn. Oberwolfach Seminars, vol. 35. Birkhäuser Verlag, Basel (2009)
33. Karp, R.M.: Reducibility among combinatorial problems. In Miller, R.E., Thatcher, J.W. (eds) Complexity of Computer Computations, pp. 85–103. Plenum, New York (1972).
34. Khachiyan, L.: A polynomial algorithm in linear programming. Soviet Math. Doklady **20**, 191–194 (1979)
35. Khovanskii, A.G.: Fewnomials. AMS, Providence (1991)
36. Koiran, P., Portier, N., Rojas, J.M.: Counting Tropically Degenerate Valuations and p-adic Approaches to the Hardness of the Permanent. Submitted for publication, also available as Math ArXiV preprint `1309.0486`
37. Lee, T.-L., Li, T.-Y.: Mixed volume computation in solving polynomial systems. In: Randomization, Relaxation, and Complexity in Polynomial Equation Solving, Contemporary Mathematics, vol. 556, pp. 97–112. AMS (2011)
38. Litvinov, G.L., Sergeev, S.N. (eds) Tropical and Idempotent Mathematics, International Workshop TROPICAL-07 (Tropical and Idempotent Mathematics, August 25—30, 2007. Independent University, Contemporary Mathematics, vol. 495. AMS (2009)
39. Maclagan, D., Sturmfels, B.: Introduction to Tropical Geometry, AMS Graduate Studies in Mathematics, vol. 161, AMS (2015, to appear).
40. Matveev, E.M.: An explicit lower bound for a homogeneous rational linear form in logarithms of algebraic numbers. II. Izv. Ross. Akad. Nauk Ser. Mat. **64**(6), 125–180 (2000); translation in Izv. Math. **64**(6), 1217–1269 (2000)
41. Mikhalkin, G.: Decomposition into pairs-of-pants for complex algebraic hypersurfaces. Topology **43**(5), 1035–1065 (2004)
42. Müller, S., Feliu, E., Regensburger, G., Conradi, C., Shiu, A., Dickenstein, A.: Sign conditions for injectivity of generalized polynomial maps with applications to chemical reaction networks and real algebraic geometry. Math ArXiV preprint `1311.5493`
43. Nesterenko, Y.: Linear Forms in Logarithms of Rational Numbers. Diophantine Approximation (Cetraro, 2000). Lecture Notes in Math., vol. 1819, pp. 53–106. Springer, Berlin (2003)
44. Newton, I.: Letter to Oldenburg dated 1676 Oct 24, the correspondence of Isaac Newton, II, pp. 126–127. Cambridge University Press, Cambridge (1960)
45. Ng, CTD., Barketau, M.S., Cheng, T.C.E., Kovalyov, M.Y.: Product Partition and related problems of scheduling and systems reliability: Computational complexity and approximation. Eur. J. Oper. Res. **207**, 601–604 (2010)

46. Nilsson, L., Passare, M.: Discriminant coamoebas in dimension two. J. Commut. Algebra **2**(4), 447–471 (2010)
47. Ning, Y., Avendaño, M.E., Mortari, D.: Sequential design of satellite formations with invariant distances. AIAA J. Spacecraft Rockets **48**(6), 1025–1032 (2011)
48. Nisse, M., Sottile, F.: Phase limit set of a variety. J. Algebra Number Theory **7**(2), 339–352 (2013)
49. Papadimitriou, C.H.: Computational Complexity. Addison-Wesley (1995)
50. Phillipson, K.R., Rojas, J.M.: $\mathscr{A}$-discriminants, and their cuttings, for complex exponents (2014, in preparation)
51. Pin, J.-E.: Tropical semirings. Idempotency (Bristol, 1994), Publ. Newton Inst., 11, pp. 50–69. Cambridge Univ. Press, Cambridge (1998)
52. Plaisted, D.A.: New NP-hard and NP-complete polynomial and integer divisibility problems. Theor. Comput. Sci. **31**(1–2), 125–138 (1984)
53. Schrijver, A.: Theory of Linear and Integer Programming. Wiley (1986)
54. Shub, M., Smale, S.: The complexity of Bezout's theorem I: Geometric aspects. J. Am. Math. Soc. **6**, 459–501 (1992)
55. Shub, M., Smale, S.: The Complexity of Bezout's theorem II: Volumes and Probabilities. In: Eyssette, F., Galligo, A. (eds) Computational Algebraic Geometry, pp. 267–285. Birkhauser (1992)
56. Shub, M., Smale, S.: The Complexity of Bezout's theorem III: Condition number and packing. J. Complexity **9**, 4–14 (1993)
57. Shub, M., Smale, S.: The complexity of Bezout's theorem IV: Probability of success; extensions. SIAM J. Numer. Anal. **33**(1), 128–148 (1996)
58. Shub, M., Smale, S.: The complexity of Bezout's theorem V: Polynomial time. Theor. Comput. Sci. **133**(1), 141–164 (1994).
59. Sipser, M.: Introduction to the Theory of Computation, 3rd edn. Cengage Learning (2012)
60. Smale, Steve, Mathematical problems for the next century. Math. Intelligencer **20**(2), 7–15 (1998)
61. Smale, S.: Mathematical Problems for the Next Century. Mathematics: Frontiers and Perspectives, pp. 271–294. Amer. Math. Soc., Providence (2000)
62. Sommese, A.J., Wampler, C.W.: The Numerical Solution to Systems of Polynomials Arising in Engineering and Science. World Scientific, Singapore (2005)
63. Spielman, D., Teng, S.H.: Smoothed analysis of termination of linear programming algorithms. Math. Program. Ser. B **97** (2003)
64. Verschelde, J.: Polynomial homotopy continuation with PHCpack. ACM Commun. Comput. Algebra **44**(4), 217–220 (2010)
65. Viro, O.Y.: Dequantization of real algebraic geometry on a logarithmic paper. In: Proceedings of the 3rd European Congress of Mathematicians. Progress in Math, vol. 201, pp. 135–146. Birkhäuser (2001)
66. Ziegler, G.M.: Lectures on Polytopes, Graduate Texts in Mathematics. Springer (1995)