Intrinsic Near Quadratic Complexity Bounds for Real Multivariate Root Counting*

J. Maurice Rojas**

Department of Mathematics, Y6508
City University of Hong Kong
83 Tat Chee Avenue
Kowloon, HONG KONG
FAX: (Hong Kong) [852] 2788-8561
`mamrojas@math.cityu.edu.hk`
`http://www.cityu.edu.hk/ma/staff/rojas`

We give a new algorithm, with three versions, for computing the number of real roots of a system of n polynomial equations in n unknowns. The first version is of Monte Carlo type and, neglecting logarithmic factors, runs in time **quadratic** in the average number of complex roots of a closely related system. The other two versions run nearly as fast and progressively remove a measure zero locus of failures present in the first version. Via a slight simplification of our algorithm, we can also count complex roots, with or without multiplicity, within the same complexity bounds. We also derive an even faster algorithm for the special case $n=2$, which may be of independent interest.

Remark 1 *For technical reasons, we will only consider roots with all coordinates nonzero. This restriction will be lifted in forthcoming work of the author, and most of the theory necessary for this extension has already appeared in [Roj97c,Roj98a]. We will also assume that all our input polynomial systems have only finitely many complex roots with all coordinates nonzero. This mild restriction can also be lifted, but the details will be covered in a later paper. In any case, a feature of our algorithm is that we* **do** *allow infinitely many roots at infinity.*[1]

Our Main Theorem, along with an independent approach in [MP98], represent the first complexity bounds for real root counting which are **intrinsic** in the sense that they depend mainly on the geometric properties of the underlying

---

* Extended abstract to appear in the Proceedings of the Sixth Annual European Symposium on Algorithms, Lecture Notes on Computer Science, vol. 1461, pp. 127–138, Springer-Verlag (1998).

** This research was completed at City University of Hong Kong and partially funded by a National Science Foundation Mathematical Sciences Postdoctoral Fellowship.

[1] Most prior resultant-based algorithms for equation solving assume finiteness of the number of solutions (typically in projective space) of a homogeneous version of the input system.

zero set. Indeed, it was an open question whether any older algorithms for real root counting satisfied these more refined complexity bounds.[2]

Before stating our results, we give a quick illustration of the benefits of intrinsic bounds. Consider the following bivariate polynomial system:

$$f_1(x,y) = a_1 + a_2x^8y^2 + a_3x^9y^2 + a_4x^{10}y^2 + a_5x^{10}y^3 + a_6x^9y^3$$
$$f_2(x,y) = b_1 + b_2x + b_3xy^3 + b_4x^2y^2 + b_5x^5y^3 + b_6x^5y^4 + b_7x^4y^4$$

Suppose we choose complex constants for the coefficients $\{a_i\}$ and $\{b_i\}$. Recall that the classical Bézout's Theorem [Mum82] gives the loose upper bound of $13\cdot 9 = 117$ (the product of the total degrees of the polynomials) on the number of isolated complex roots, for **any** choice of the coefficients. We say "loose" because the **maximal** number of isolated complex roots, for any choice of coefficients, is actually 35. We will return to this example in section 3, but our point for the moment is that exploiting the monomial term structure, as opposed to working solely in terms of d (the maximum of the total degrees of the input systems), can result in much better bounds.

To state our complexity results, let F be an $n \times n$ polynomial system with support contained in an n-tuple of point sets $E := (E_1, \ldots, E_n)$ [Roj97a]. (So the n-tuple E determines precisely which monomial terms are allowed to appear in F.) Let us briefly recall Bernshtein's Theorem [Ber75,GKZ94,Ewa96,Roj98a], which asserts the following: (1) The number, $\mathcal{N}_{\mathbb{C}}$, of isolated complex roots of F with all coordinates nonzero is bounded above by the **mixed volume**, $\mathcal{M}(E)$, of the convex hulls of $E_1, \ldots, E_n$, (2) $\mathcal{M}(E) \leq d^n$ when the support of F **is** E, and (3) for fixed E, $\mathcal{N}_{\mathbb{C}} = \mathcal{M}(E)$ for all choices of coefficients outside of a codimension 1 locus. For most purposes, the last statement can be interpreted as "$\mathcal{N}_{\mathbb{C}} = \mathcal{M}(E)$ with probability 1."

Definition 1 *For any fixed $k \leq n$ and any n-tuple of point sets in $\mathbb{R}^k$, $(E_1, \ldots, E_n)$, define $\mathcal{M}^{\text{ave}}_{\mathbf{E}}$ to be the* **average** *of the k-dimensional mixed volumes $\{\mathcal{M}(\mathcal{E})\}$, as $\mathcal{E}$ ranges over all* **ordered** *k-tuples $(\mathcal{E}_1, \ldots, \mathcal{E}_k)$ with $\mathcal{E}_i \in \{E_1, \ldots, E_n\}$ for all $i \in \{1, \ldots, k\}$. Also, for any $\boldsymbol{a} \in \mathbb{Z}^n \setminus \{\mathbf{O}\}$, let $p_{\boldsymbol{a}}$ denote the orthogonal projection mapping $\mathbb{R}^n$ onto the hyperplane perpendicular to the vector $\boldsymbol{a}$.*

Main Theorem *Suppose F is an $n \times n$ polynomial system with real coefficients, a total of m monomial terms, and support contained in E. Then there is a Monte Carlo algorithm (cf. section 2, Main Algorithm) for counting the exact number of real roots of F, with the following versions and run times (counting arithmetic steps over $\mathbb{R}$):*

Version	*Run Time*$=\mathcal{O}(\cdots)$	codim
1	$\mathcal{M}(E)S_1 \log^2 S_1$	≥ 1
2	$\mathcal{M}(E)S_m^2 \log^2 S_m$	≥ 2
Safe	$\mathcal{M}(E)S_m^3 \log^2 S_m$	∞

[2] We should also point out that the existence of algorithms for real root counting with complexity bounds solely in terms of the number of **real** roots is an open problem of the utmost interest.

where[3] $S_t := \sqrt{n}e^n(\mathcal{M}^{\mathrm{ave}}_E + t^2\sqrt{n}\max_{\boldsymbol{a}\in[-t^2,t^2]^n}\{\mathcal{M}^{\mathrm{ave}}_{p_{\boldsymbol{a}}(E)}\})$. *The codimension entries refer to the codimension of a locus of coefficient values (depending only on E) where the algorithm may not work. In particular, the safe version will count the real roots of* **any** *input F and, with a controllably small probability, will at worst underestimate the number of real roots.*

Furthermore, when $n=2$, we can make the following improvements:

Version	*Run Time*$=\mathcal{O}(\cdots)$	codim
1	$m\log m + \mathcal{M}(E)S'_1\log^2 S'_1$	≥ 1
Safe	$m\log m + \mathcal{M}(E)S'_m\log^2 S'_m$	∞

where $S'_t := t\max_{\boldsymbol{a}\in[-t,t]^2}\{\mathrm{Length}(p_{\boldsymbol{a}}(E_1)) + \mathrm{Length}(p_{\boldsymbol{a}}(E_2))\}$.

Via a slight simplification of the above algorithms, the same complexity bounds hold for **complex** *root counting as well, with or without multiplicity.*

When $n>2$ and E is such that an algebraic formula of a special type exists (cf. section 1), we can remove a factor of S_m from the complexities of versions 2 and "Safe," and replace the parameter S_t by the smaller quantity $\mathcal{M}(E)+t^2\sqrt{n}\max_{\boldsymbol{a}\in[-t^2,t^2]^n}\{\sum_{i=1}^n \mathcal{M}(p_{\boldsymbol{a}}(E_1),\ldots,p_{\boldsymbol{a}}(E_{i-1}),p_{\boldsymbol{a}}(E_{i+1}),\ldots,p_{\boldsymbol{a}}(E_n))\}$. Also, in practice, one can drastically reduce S_t and S'_t once E is fixed.

We also point out that when $n>2$ our algorithm relies on a preprocessing step. However, this preprocessing need only be done once per monomial term structure. In applications where the monomial term structure is fixed and the coefficients vary many times, the preprocessing step has negligible complexity (cf. section 2).

Our algorithm for real root counting will use the **toric** (a.k.a. **sparse**) resultant[4] [Stu93,Stu94,GKZ94,EC95,Stu97], Sturm sequences [Roy95], some fast matrix arithmetic algorithms [BP94], and a bit of convex geometry [Ewa96]. After the description and complexity analysis of our algorithm in section 2, we give a simple example of our algorithm in section 3. Finally, in section 4, we briefly compare our complexity bounds with some older algorithms, as well as the new approach of [MP98].

1 An Executive Summary of Resultants

We will mainly follow the notation of [Roj97a]. In particular, let $\bar{F}:=(f_1,\ldots,f_{n+1})$ be an $(\mathbf{n}+\mathbf{1})\times\mathbf{n}$ **indeterminate** polynomial system with support $\bar{E}$. Then there is a corresponding **toric** (a.k.a. **sparse**) **resultant**, $\mathrm{Res}_{\bar{E}}(\cdot)$, which is a polynomial in the coefficients $\{c_{i,e} \mid i\in\{1,\ldots,n+1\}, e\in E_i\}$. Let $\mathcal{C}_{\bar{E}}$ be the **vector** consisting of all these indeterminate coefficients, and define[5]

[3] The first and second "average mixed volume" terms are respectively n-dimensional and $(n-1)$-dimensional. Also, $p_{\boldsymbol{a}}(E) := (p_{\boldsymbol{a}}(E_1),\ldots,p_{\boldsymbol{a}}(E_n))$.

[4] Other commonly used prefixes for this modern extension of the classical resultant include: $\mathcal{A}$-, $(\mathcal{A}_1,\ldots,\mathcal{A}_k)$-, mixed, sparse mixed, and Newton.

[5] We use $|\cdot|$ for set cardinality. In a completely analogous way, we will also let $\mathcal{C}_E$ be the vector of coefficients of F.

$|\bar{E}|:=\sum_{i=1}^{n+1}|E_i|$. Then, up to sign, $\mathrm{Res}_{\bar{E}}(\cdot)$ can be defined as the unique polynomial of lowest degree (with relatively prime integer coefficients) satisfying the following property: $\bar{F}|_{\mathcal{C}_{\bar{E}}=\mathcal{C}}$ has a root in $(\mathbb{C}^*)^n$ for some $\mathcal{C}\in\mathbb{C}^{|\bar{E}|} \Longrightarrow \mathrm{Res}_{\bar{E}}(\mathcal{C})=0$. (We let $\mathbb{C}^*$ denote the set of **nonzero** complex numbers.) For convenience, we will usually write $\mathrm{Res}_{\bar{E}}(\bar{F})$ instead of $\mathrm{Res}_{\bar{E}}(\mathcal{C})$ whenever $\mathcal{C}_{\bar{E}}=\mathcal{C}$.

Example 1 *Suppose we pick* $E_1=\cdots=E_{n+1}=\{\mathbf{O},\hat{e}_1,\ldots,\hat{e}_n\}$, *where* $\mathbf{O}\in\mathbb{R}^n$ *is the origin, and* $\hat{e}_i\in\mathbb{R}^n$ *is the* $i^{\underline{\mathrm{th}}}$ *standard basis vector. (So each* E_i *is the vertex set of a standard* n*-simplex.) Then we are dealing with a polynomial condition for* $n+1$ *affine forms in* n *variables to have a common root. In particular, it is easily verified that* $\mathrm{Res}_{\bar{E}}$ *is just the* **determinant** *of the* $(n+1)\times(n+1)$ *coefficient matrix of this* $\bar{F}$.

The toric resultant is the theoretical foundation for some of the fastest current methods for solving polynomial equations [MC92,Emi94]. Also, whereas the classical resultant really measured the existence of roots in projective space, the toric resultant actually measures the existence of roots in a suitable **toric variety** [GKZ94,Roj98a] — hence the author's preference for this particular name.

From basic elimination theory, one can deduce the following facts about the polynomial $\pi_i:=\mathrm{Res}_{\bar{E}}(F,u_i+x_i)$, where $\bar{E}:=(E,\{\mathbf{O},\hat{e}_i\})$ and F has constant coefficients: (1) π_i is a polynomial in the single variable u_i, and (2) $\pi_i(-\alpha_i)=0$ for any root $(\alpha_1,\ldots,\alpha_n)$ of F. This would give a general method to reduce root counting to the univariate case, were it not for two problems: $(\mathbf{P_1})$ The polynomial π_i can be identically zero, and $(\mathbf{P_2})$ two roots of F might have the same $i^{\underline{\mathrm{th}}}$ coordinate. However, there is a way out via better binomials.

Theorem 1 *[Roj97a] Let* F *be an* $n\times n$ *polynomial system with support* $E:=(E_1,\ldots,E_n)$ *such that* $\mathcal{M}(E)>0$. *Also let* P_E *be the sum of the convex hulls of the* E_i, *and pick* $\boldsymbol{a}\in\mathbb{Z}^n\setminus\{\mathbf{O}\}$ *such that the segment* $[\mathbf{O},\boldsymbol{a}]$ *is* **not** *parallel to any facet of* P_E. *Define* $\pi_{\boldsymbol{a}}(u_+,u_-):=\mathrm{Res}_{E_{\boldsymbol{a}}}(F,u_++u_-x^{\boldsymbol{a}})$ *where* $\{u_\pm\}$ *is a pair of parameters and* $E_{\boldsymbol{a}}:=(E,\{\mathbf{O},\boldsymbol{a}\})$. *Then* $\pi_{\boldsymbol{a}}$ *is either identically zero or a homogeneous polynomial of degree* $\mathcal{M}(E)$. *Furthermore, the following hold:*

1. *Let* $\varepsilon_\pm$ *be the lowest exponent of* $u_\pm$ *occuring in any monomial of* $\pi_{\boldsymbol{a}}$. *Then* F *has* **exactly** $\mathcal{N}:=\mathcal{M}(E)-\varepsilon_+-\varepsilon_-$ *roots in* $(\mathbb{C}^*)^n$, *counting multiplicities, provided* $\mathcal{N}<\infty$.
2. *Let* $\pi^1_{\boldsymbol{a}}$ *be the square-free part of* $\pi_{\boldsymbol{a}}$ *and let* $\varepsilon^1_\pm$ *be the lowest exponent of* $u_\pm$ *occuring in any monomial of* $\pi^1_{\boldsymbol{a}}$. *Then* F *has at least* $\mathcal{N}^1:=\deg(\pi^1_{\boldsymbol{a}})-\varepsilon^1_+-\varepsilon^1_-$ **distinct** *roots in* $(\mathbb{C}^*)^n$. *Furthermore, this count is exact provided the projection of roots* $\zeta\mapsto\zeta^{\boldsymbol{a}}$ *is one to one.*
3. *The set of all* $\mathcal{C}_E$ *such that* $\pi_{\boldsymbol{a}}$ *is identically zero (i.e.,* $\mathcal{N}=\infty$*) forms a locus of codimension* ≥ 2. *More generally, for* **arbitrary** $\boldsymbol{a}\in\mathbb{Z}^n\setminus\{\mathbf{O}\}$, *the set of* $\mathcal{C}_E$ *for which (1) and (2) fail to hold is of codimension* ≥ 1.
4. *If some monomial multiple of* $\pi_{\boldsymbol{a}}$ *is square-free, then no root of* F *or* $\pi_{\boldsymbol{a}}(u_+,-1)$ *has multiplicity* >1.
5. $\pi_{\boldsymbol{a}}(\zeta^{\boldsymbol{a}},-1)=0$ *for any root* $\zeta\in(\mathbb{C}^*)^n$ *of* F.
6. *When* $n=2$ *the first locus in (3) is always empty.* ■

Remark 2 *The convex geometric hypothesis on $\boldsymbol{a}$ is necessary in order to avoid problems* $(\mathbf{P_1})$ *and* $(\mathbf{P_2})$*. (In particular, if we only use $\boldsymbol{a}\in\{\hat{e}_i\}$, then the best we can hope for is a codimension 1 locus in assertion 3.) In particular, combining* **binomials** *with toric resultants is better (and faster) than the usual trick of combining generic linear forms with resultants.*

In the above theorem, the **mixed volume** is a function assigning a nonnegative integer to every input n-tuple of polytopes with integral vertices [Ewa96]. In particular, the hypothesis $\mathcal{M}(E)>0$ simply rules out those supports giving a system F having, under a suitable change of coordinates, a subsystem with too many equations in too few variables. This hypothesis can actually be checked in time polynomial in n, by using a slight variant of Edmond's matroid intersection algorithm [GK94].

We will also need the following technical result to build the "safe" version of our algorithm.

Theorem 2 *[Roj98b] Following the notation of Theorem 1, suppose $\pi_{\boldsymbol{a}}$ is identically zero. Let us then form the* **double toric perturbation of** $\pi_{\boldsymbol{a}}$*,* $\tilde{\pi}_{\boldsymbol{a}}$*, as follows: Define $\tilde{\pi}_{\boldsymbol{a}}$ to be the gcd of the coefficients of the lowest power of s in the following two polynomials*

$$\mathrm{Res}_{E_{\boldsymbol{a}}}(F-sF^*, u_+ + u_-x^{\boldsymbol{a}}) \text{ and } \mathrm{Res}_{E_{\boldsymbol{a}}}(F-sF^{**}, u_+ + u_-x^{\boldsymbol{a}}),$$

where F^* and F^{**} are distinct generic polynomial systems with support contained in E. Then $\tilde{\pi}_{\boldsymbol{a}}$ is a homogeneous polynomial of degree $\leq\mathcal{M}(E)$, **not** *identically equal to zero, satisfying all but one assertion of Theorem 1: in place of assertion 5, we instead have the following:*

5.′ $\tilde{\pi}_{\boldsymbol{a}}(\zeta^{\boldsymbol{a}},-1)=0$ *for any* **isolated** *root ζ of F.* ■

It immediately follows from Theorem 1 that the set of all eligible (F^*,F^{**}) for Theorem 2 lies outside of a locus of codimension ≥ 2 in coefficient space. So picking random number coefficients is actually a reliable way of generating such (F^*,F^{**}). Unfortunately, an efficient **deterministic** way of picking F^* and F^{**} is still unknown.

We will also use some basic results from the geometry of numbers, but for the sake of brevity, we will not enter into a detailed description. So to state our final technical lemma, we will instead close this section by recalling a salient fact about the current state of the art of toric resultant computation: For now, the most reliable way of computing $\mathrm{Res}_{\bar{E}}$ is to first find a matrix $R_{\bar{E}}$, with each entry either 0 or a coefficient of $\bar{F}$, such that (1) $\det(R_{\bar{E}})$ is not identically zero, and (2) $\mathrm{Res}_{\bar{E}}(\bar{F})$ divides $\det(R_{\bar{E}})$. (The remainder of this computation is covered in [Emi94,EC95], among other references.) In particular, we say that E is **determinantal** iff, for all $\boldsymbol{a}\in\mathbb{Z}^n$, we can always construct the matrix $R_{E_{\boldsymbol{a}}}$ so that the excess factor from (2) is a constant. This is the condition sufficient for the improved value of S_t in our complexity bounds, as mentioned just after the statement of our Main Theorem. Our final lemma also explains these values of S_t.

Lemma 1. *Suppose* $\boldsymbol{a}\in\mathbb{Z}^n\setminus\{\mathbf{O}\}$ *and* F *is an indeterminate* $n\times n$ *polynomial system with support* E*. Then the total degree of the toric resultant* $\mathrm{Res}_{E_{\boldsymbol{a}}}(\mathcal{C}_{\bar{E}})$ *is* $\mathcal{M}(E_1+\{\mathbf{O},\boldsymbol{a}\},\ldots,E_n+\{\mathbf{O},\boldsymbol{a}\})$*, and the degree with respect to the coefficents of* F *is exactly* $\mathcal{M}(E)$*. Furthemore,* $R_{E_{\boldsymbol{a}}}$ *is* **quasi-Toeplitz** *[BP94] and its size* S *is bounded above by*

$$\sqrt{n}e^n(\mathcal{M}^{\mathrm{ave}}_E+\mathrm{Length}(\boldsymbol{a})\mathcal{M}^{\mathrm{ave}}_{p_{\boldsymbol{a}}(E)})$$

in general, and this upper bound improves to

$$\mathcal{M}(E)+\mathrm{Length}(\boldsymbol{a})\sum_{i=1}^{n}\mathcal{M}(p_{\boldsymbol{a}}(E_1),\ldots,\widehat{p_{\boldsymbol{a}}(E_i)},\ldots,p_{\boldsymbol{a}}(E_n))\}$$

if E *is determinantal.* ■

In the above, we use the natural addition for point sets in $\mathbb{R}^n$: $A+B:=\{a+b \mid a\in A, b\in B\}$ for any $A,B\subseteq\mathbb{R}^n$.

2 The Power of Sparsity in Real Root Counting

The following is an overview of our algorithm. The three versions stated earlier will follow from varying the execution of Steps 3–5 (and how many times this is done), while our preprocessing step is essentially the execution of Steps 1–2.

Main Algorithm

Input: *An* $n\times n$ *polynomial system* F *with support* E*, such that* $\mathcal{M}(E)>0$.[6] *(We also assume* F *has only finitely many roots in* $(\mathbb{C}^*)^n$*, but allow infinitely many roots at infinity.)*

Output: *The number of real roots of* F *with all coordinates nonzero.*

Description:

1. *Pick a point* $\boldsymbol{a}\in\mathbb{Z}^n$ *such that the coordinates of* $\boldsymbol{a}$ *have no nontrivial common divisor and the line segment* $[\mathbf{O},\boldsymbol{a}]$ *is* **not** *parallel to any facet of the polytope* $\sum_{i=1}^n\mathrm{Conv}(E_i)$.
2. *Define* $E_{\boldsymbol{a}}$ *to be the* $(n+1)$*-tuple* $(E,\{\mathbf{O},\boldsymbol{a}\})$*, and compute* $R_{E_{\boldsymbol{a}}}$.
3. *Following the notation of Theorem 1, evaluate the polynomial* $\pi_{\boldsymbol{a}}(\alpha)$ *at* $\mathcal{M}(E)+1$ *distinct choices of* α.
4. *Using the values from Step 3, solve for the coefficients of* $\pi_{\boldsymbol{a}}$.
5. *Find the number* $\mathcal{N}_{\mathbb{R}}$ *of nonzero real roots of the polynomial* $\pi_{\boldsymbol{a}}(u_+,1)$ *via Sturm sequences.*
6. *Output* $\mathcal{N}_{\mathbb{R}}$ *as the number of real roots of* F.

Remark 3 *We actually obtain more than just the number of roots: If we simply stop at Step 4, Theorem 1 tells us that* $\pi(x,-1)$ *is a polynomial whose roots are precisely the values of* $\zeta^{\boldsymbol{a}}$ *as* ζ *ranges over all roots of* F *in* $(\mathbb{C}^*)^n$.

[6] As we remarked in the last section, this condition merely rules out certain overdetermined systems, and can easily be checked in polynomial time.

The approach taken by our algorithm (univariate reduction and counting sign alternations in a sequence of polynomial evaluations) is essentially classical. However, the use of the toric resultant, and a precise understanding of the potential degeneracies of this method, are new. The correctness of our algorithm will be derived below, so let us begin by describing our preprocessing step.

First note that $\boldsymbol{a}$ satisfies the hypotheses of our Main Algorithm if and only if $\boldsymbol{a}$ is in the complement of a finite union of hyperplanes in $\mathbb{Z}^n$. Furthermore, the number of such hyperplanes is $\mathcal{O}(m^{2n})$, when $n > 2$ [GS93], or $\mathcal{O}(m)$ when $n = 2$. It is then easily shown that if one picks a uniformly random $\boldsymbol{a}$ in $[-\mathcal{O}(m^2/\delta), \mathcal{O}(m^2/\delta)]^n$, and removes common factors from the coordinates of $\boldsymbol{a}$, then the probability that $\boldsymbol{a}$ avoids the hyperplanes is at least $1 - \delta$.

To compute $R_{E_{\boldsymbol{a}}}$, it immediately follows from [Emi96] that this will take $\mathcal{O}(Sn^{9.5}m^{6.5}\log^2 \frac{d}{\varepsilon})$ bit operations for an error probability of ε, where S is as stated in Lemma 1 and d is an upper bound on the absolute values of the coordinates of the E_i. (Note that we are also implicitly using the fact that the $(n+1)^{\underline{\text{st}}}$ support consists of just two points.) Thus, if we chose $\boldsymbol{a}$ randomly as detailed above, we immediately obtain the above complexity bound for a Monte Carlo success probability of $1 - \delta - \varepsilon$ for our preprocessing step.

Alternatively, if we want to preprocess deterministically, let us make the following more refined observation on where $\boldsymbol{a}$ must lie: $\boldsymbol{a}$ satisfies the hypotheses of our algorithm if and only if $\boldsymbol{a}$ lies in the interior of some n-dimensional cone in the complement of the hyperplane arrangement defined by the facets of P_E. To check the latter condition, it clearly suffices to know the faces of the convex hulls of all the E_i to generate such a cone σ. To then generate $R_{E_{\boldsymbol{a}}}$ without randomization, we can make use of the Cayley trick [GKZ94] to find a mixed subdivision of $E_{\boldsymbol{a}}$, and then enumerate shifted lattice points in our mixed subdivision. (That this suffices to build a resultant matrix follows from the development of [Stu93,Emi94].) From the well-known complexity bounds for computing convex hulls and simplicial subdivisions [PS85], it is not hard to thus derive a deterministic complexity bound of $\mathcal{O}(S + (m+2)^{\lfloor n - \frac{1}{2} \rfloor})$ real arithmetic steps for our preprocessing. Furthermore, we can always pick a constant number of good $\boldsymbol{a}$ in $[-\mathcal{O}(m^2), \mathcal{O}(m^2)]^n$ within the same asymptotic complexity bound.

Before describing the three versions of our algorithm, we will briefly detail the complexity of Steps 4–5 of our Main Algorithm. This will then completely reduce our complexity analysis to how we compute $\pi_{\boldsymbol{a}}$ (or $\tilde{\pi}_{\boldsymbol{a}}$), and whether we do this more than once.

First recall that the coefficients of a polynomial of degree d can be recovered via interpolation in time $\mathcal{O}(d \log^2 d)$ [BP94]. Also, via the same reference, we know that the signs at $\pm\infty$ of the Sturm sequence of such a polynomial can be computed within time $\mathcal{O}(d \log^2 d)$ as well. Furthermore, the gcd of two polynomials of degree d can also be computed within $\mathcal{O}(d \log^2 d)$ arithmetic steps [BP94]. Thus, since $d \leq \mathcal{M}(E)$ in our case at hand (by Lemma 1), we need only show that Step 3 can be done quickly enough.

Recall that the determinant of a $k \times k$ quasi-Toeplitz matrix can be evaluated within time $\mathcal{O}(k \log^2 k)$ [BP94]. Also recall that the condition that two distinct

roots of F lie on a hypersurface of the form $\{\zeta \mid \zeta^{\boldsymbol{a}}=\text{constant}\}$ is a codimension ≥ 1 condition in the space of coefficients. (This follows easily from considering resultants and the univariate discriminant.) We now describe our three versions.
Version 1: Here, since we are only interested in a codimension 1 locus, we can be quite lazy. First note that by Theorem 1 (assertion 3), we can just pick $\boldsymbol{a}=\hat{e}_i$ for any i. (So we can avoid the construction of σ if we choose to be deterministic in our preprocessing.) Next, note that $R_{E_{\boldsymbol{a}}}$ is a nonzero constant multiple of $\pi_{\boldsymbol{a}}$, outside of a codimension 1 locus. Thus, recalling our observations on the map $\zeta \mapsto \zeta^{\boldsymbol{a}}$, it suffices to work with $\mathcal{M}(E)+1$ evaluations of $\det(R_{E_{\boldsymbol{a}}})$.

To conclude our complexity analysis, simply note that size of $R_{E_{\boldsymbol{a}}}$ is bounded above as specified by Lemma 1. Since $R_{E_{\boldsymbol{a}}}$ is quasi-Toeplitz, our first complexity bounds follow. ■

Remark 4 *Our first algorithm* **can** *detect failure, but only if no two distinct complex roots of F have the same value of $\zeta^{\boldsymbol{a}}$. (This is part of the codimension ≥ 1 locus of failure.) We make this failure even less likely in our next version.*

Version 2: First note that in order to maintain a failure locus of codimension ≥ 2, we can no longer rely solely on determinants of $R_{E_{\boldsymbol{a}}}$. So we must must add an extra indeterminate and perturb, in order to compute our necessary $\mathcal{O}(\mathcal{M}(E))$ values of $\pi_{\boldsymbol{a}}(\alpha)$. This perturbation technique is described in [EC95], and results in an increase of our last complexity bound by an additional factor of S_t. (Note also that this particular step is necessary **only** if $E_{\boldsymbol{a}}$ is not determinantal.)

The only other worry is the projection of roots defined by $\zeta \mapsto \zeta^{\boldsymbol{a}}$. However, we can perform the following trick: Run the algorithm twice, but now using two distinct values of $\boldsymbol{a} \in [-\mathcal{O}(m^2), \mathcal{O}(m^2)]^n$ satisfying the hypotheses of our Main Algorithm. One then uses the **largest** putative root count, and it is easily shown (for fixed E and a fixed pair of $\boldsymbol{a}$'s) that the set of $\mathcal{C}_E$ for which both of these root counts is wrong forms a codimension ≥ 2 locus. Adding up the preceding complexities, we obtain our desired bound. ■
Safe: Here, we now need only worry about the case where $n>2$ and $\pi_{\boldsymbol{a}}$ vanishes identically. By Theorem 2, we can then simply work with $\tilde{\pi}_{\boldsymbol{a}}$ instead of $\pi_{\boldsymbol{a}}$.

This entails working with a slight modification of $R_{E_{\boldsymbol{a}}}$ with an **extra** indeterminate s, appearing in exactly S_m rows of our new matrix. We must then work over $\mathbb{R}[s]$ instead of $\mathbb{R}$, and it is easily checked (via the techniques of [Can89]) that this adds only one more factor of S_m to our complexity. The remaining gcd computation falls well within the asymptotic bounds of the preceding computational work. ■

Note that to calculate the number of **complex** roots, we can essentially use the same algorithm: We need only omit the Sturm sequence calculation, and simply calculate the exponent of the lowest order term of $\pi_{\boldsymbol{a}}(u_+, 1)$. Furthermore, if we only want to count distinct roots without multiplicities, we need only work with the square-free part of $\pi_{\boldsymbol{a}}(u_+, 1)$. This square-free part can be isolated in time $\mathcal{O}(\mathcal{M}(E)\log\mathcal{M}(E))$ by using a fast gcd algorithm [BP94]. So we have now proved Main Theorem, except for our speedier algorithm in the case of $n=2$.

For lack of space, we will simply give an abbreviated description of the improved algorithm (deferring a complete analysis to the full version of this paper):

First, pick a $U \in \mathbb{GL}_2(\mathbb{Z})$ such that the first column of U is not parallel to any edge of P_E, and the entries of U have absolute value $\mathcal{O}(t)$. Next, we make the monomial change of variables $x \mapsto u^U$. We then use our Main Algorithm, but with a slight change: We instead compute the Sylvester resultant, $\pi(u_2)$, of $f_1(u_1)$ and $f_2(u_1)$. (So f_1 and f_2 are considered as polynomials with coefficients in $\mathbb{R}[u_2]$.) Also, we use the corresponding Sylvester matrix in place of $R_{E_{\boldsymbol{a}}}$, and π in place of $\pi_{\boldsymbol{a}}$.

By using a slightly modified version 1, and a simplified version "safe," we can obtain the stated reduced complexity bounds. In particular, we respectively take $t=1$ and $t=m$ in these two versions. ■

3 A Simple Example of Our Algorithm

Suppose one would like to count the number of real roots of the following bivariate polynomial system:

$$f_1(x,y) = 1 - 2x^8y^2 - 3x^9y^2 + x^{10}y^2 + x^{10}y^3 + x^9y^3$$

$$f_2(x,y) = 1 + x + 2xy^3 - 5x^2y^2 + 2x^5y^3 + 2x^5y^4 + 2x^4y^4$$

Note that the above system has **no** roots with either coordinate zero. (If we set x and/or y to zero, the first equation becomes $1 = 0$.) So to count the real roots of this system, it suffices to count roots in $(\mathbb{R}^*)^2$ via our Main Algorithm.[7]

Following the notation of our Main Algorithm, the convex hulls $\mathrm{Conv}(E_1)$ and $\mathrm{Conv}(E_2)$ (which indeed satisfy the hypotheses of our algorithm) are illustrated in Figure 1. It is then easily verified that $\sum_{i=1}^{2} \mathrm{Conv}(E_i)$ is a heptagon with no

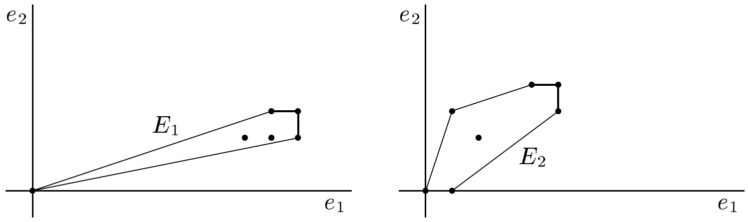

Fig. 1. The supports E_1 and E_2, along with their convex hulls. Note that each of the three "upper" edges of $\mathrm{Conv}(E_1)$ is parallel to some edge of $\mathrm{Conv}(E_2)$.

edge parallel to the line segment $[0,(1,1)]$. So we can pick $\boldsymbol{a} := (1,1)$ for Step 1.

[7] We will illustrate the general Main Algorithm, without the special optimizations for $n=2$.

At this point, we can use the author's `Matlab` software package[8] **`res2.m`** to compute $R_{E_{\boldsymbol{a}}}$. We omit a description of the resulting 63×63 matrix[9], but for the sake of completeness, we will display part of $\pi_{\boldsymbol{a}}(u_+, 1)$:

$$-38400u_+^{31} + 7808u_+^{30} + \cdots + 10u_+^3 + 2u_+^2 + 1$$

This polynomial is square-free (easily verified with `Maple`) so by Theorem 1, $\mathcal{N}_{\mathbb{R}}$ (from our Main Algorithm) will indeed be the number of real roots of our 2×2 system. `Maple` can then do the necessary Sturm sequence calculation for us via the command `realroot` in a fraction of a second to obtain that $\pi_{\boldsymbol{a}}(u_+, 1)$ has exactly **3 real roots**.

Finally, note that our Main Algorithm (via Theorem 1) also gives us the exact number of complex roots: $\underline{\mathbf{31}} = 35 - 4$. This is already a significant improvement over the classical Bézout's theorem (which gave an upper bound of 117) and even Bernshtein's theorem [Ber75] (which gives an upper bound of 35).

In closing, we point out that experimental evidence with polynomial system solving algorithms based on the **classical** resultant [MC92] suggests that we can expect a decent `C` implementation of our algorithms to run within milliseconds when $\mathcal{M}(E)$ is less than 1000 (once the preprocessing step has been done). Thus our algorithms, which are based on the more efficient toric resultant, have considerable practical potential.

4 Conclusion

We have seen a fast algorithm for counting the number of real and complex solutions of a system of n polynomial equations in n unknowns. A feature of our algorithm is that its worst-case complexity can be stated in purely geometric terms. By avoiding the use of generic linear projections, our algorithm thus avoids many classically occuring degeneracies and computational bottlenecks.

We now briefly compare our worst-case complexity bounds with those of other algorithms. It appears that before this paper, the best complexity bound for real root counting was $d^{\mathcal{O}(n)}$ [Can93,Roy95,BPR96], and work of Renegar [Ren87] suggests that the implied asymptotic constant is at worst 4. However, regardless of this asymptotic constant, there are many infinite families of sparse systems of equations for which d^n exceeds our complexity bounds by orders of magnitude. For instance, in [Roj97b], a family of systems $\{F_n\}_{n=1}^{\infty}$ is presented where F_n is $n \times n$ and has exactly r complex roots for all $n \in \mathbb{N}$. However, the corresponding value of d^n for F_n is $n^n r^n$. With a bit more geometric work, we can see that our complexity bound becomes $\mathcal{O}(\sqrt{n}e^n r)$ while the older bound is $(nr)^{\mathcal{O}(n)}$. Alternatively, to get another family of examples, we can simply

[8] An alpha version of this program is publically available from the author's webpage.

[9] The 63 (well within the given estimate for S_{13}) is but an artifact of our particular algorithm for computing $R_{E_{\boldsymbol{a}}}$. If E were determinantal (which we don't know at this point), we would be able to obtain a 47×47 matrix instead.

specialize our results to the case of n equations of total degree d to obtain a complexity bound of $\mathcal{O}(d^{2n+\frac{n+\frac{3}{2}\log n+\log\log d}{\log d}})$.

The author then discovered the paper [MP98] by Mourrain and Pan, after this paper was first submitted to these proceedings. In [MP98] the authors discuss the related problem of approximating all real and complex roots of F, as well as real root counting. For the latter problem, a Las Vegas bound of $\mathcal{O}(3^n D^2 \log D)$ is stated in [MP98, Proposition 7.2], where D is an upper bound on the number of complex roots. For the case of counting roots in $(\mathbb{R}^*)^n$, D can be taken to be $\mathcal{M}(E)$. So here, the comparison is a bit more subtle.

First, we mention that the preprocessing complexity appears to be the same in both approaches, as both methods make use of the toric resultant. Secondly, the loci of failure for the [MP98] algorithm and version 1 of our algorithm both have codimension ≥ 1.[10] (While there is overlap in these loci, neither locus contains the other.) Thirdly, no mention is made in [MP98] of how the complexity rises when the failure loci are removed.

We are thus reduced to comparing $\mathcal{O}(\mathcal{M}(E)S_1 \log^2 S_1)$ (the complexity of version 1 of our algorithm) with $\mathcal{O}(3^n \mathcal{M}(E)^2 \log \mathcal{M}(E))$. Considering the case where all the Newton polytopes are identical and equal to some polytope $P \subset \mathbb{R}^n$, we can obtain an infinite family of supports for which our bound is better, and another infinite family where the bound from [MP98] is better. (For the first family, one can assume P is any cube. For the second family, one can assume that P is a small neighborhood of an $(n-1)$-dimensional polytope of a certain type.) For lack of space, we leave the details to the full version of this paper.

In conclusion, it appears that a combination of both these techniques can be quite profitable. Also, considerable computational experimentation with these algorithms would be quite valuable for gaining further insight into such a combination. We hope to address this point in the future.

References

[Ber75] Bernshtein, D. N., *"The Number of Roots of a System of Equations,"* Functional Analysis and its Applications (translated from Russian), Vol. 9, No. 2, (1975), pp. 183–185.

[BPR96] Basu, S., Pollack, R., and Roy, M.-F., *"On the Combinatorial and Algebraic Complexity of Quantifier Elimination,"* J. ACM 43 (1996), no. 6, pp. 1002–1045.

[BP94] Bini, Dario and Pan, Victor Y. *Polynomial and Matrix Computations, Volume 1: Fundamental Algorithms,* Progress in Theoretical Computer Science, Birkhäuser, 1994.

[Can89] Canny, John F., *"Generalized Characteristic Polynomials,"* Symbolic and Algebraic Computation (Rome, 1988), pp. 293–299, Lecture Notes in Comput. Sci., 358, Springer, Berlin, 1989.

[Can93] ____________, *"Improved Algorithms for Sign Determination and Existential Quantifier Elimination,"* Comput. J. **36** (1993), no. 5, pp. 409–418.

[10] Note that by expanding our locus of failure slightly (still keeping codimension ≥ 1), we can change version 1 from Monte Carlo type to Las Vegas type (cf. remark 4).

[Emi94] Emiris, Ioannis Z., *"Sparse Elimination and Applications in Kinematics,"* Ph.D. dissertation, Computer Science Division, U. C. Berkeley (December, 1994).

[Emi96] Emiris, Ioannis Z., *"On the Complexity of Sparse Elimination,"* J. Complexity **12** (1996), no. 2, pp. 134–136.

[EC95] Emiris, Ioannis Z. and Canny, John F., *"Efficient Incremental Algorithms for the Sparse Resultant and the Mixed Volume,"* Journal of Symbolic Computation, vol. 20 (1995), pp. 117–149.

[Ewa96] Ewald, Günter, *Combinatorial Convexity and Algebraic Geometry,* Graduate Texts in Mathematics, 168, Springer-Verlag, New York, 1996.

[GKZ94] Gel'fand, I. M., Kapranov, M. M., and Zelevinsky, A. V., *Discriminants, Resultants and Multidimensional Determinants,* Birkhäuser, Boston, 1994.

[GK94] Gritzmann, Peter and Klee, Victor, *"On the Complexity of Some Basic Problems in Computational Convexity II: Volume and Mixed Volumes,"* Polytopes: Abstract, Convex, and Computational (Scarborough, ON, 1993), pp. 373–466, NATO Adv. Sci. Inst. Ser. C Math. Phys. Sci., 440, Kluwer Acad. Publ., Dordrecht, 1994.

[GS93] Gritzmann, Peter and Sturmfels, Bernd, *"Minkowski Addition of Polytopes: Computational Complexity and Applications to Gröbner Bases,"* SIAM J. Discrete Math. 6 (1993), no. 2, pp. 246–269.

[MC92] Manocha, Dinesh and Canny, John *"Real Time Inverse Kinematics for General 6R Manipulators,"* Technical Report, University of California, Berkeley, 1992.

[MP98] Mourrain, Bernard and Pan, Victor, *"Asymptotic Acceleration of Solving Multivariate Polynomial Systems of Equations,"* STOC '98, ACM Press, 1998.

[Mum82] Mumford, David, *"Algebraic Geometry I: Complex Algebraic Varieties,"* reprint of 1976 edition, Classics in Mathematics, Springer-Verlag, Berlin, 1995, x+186 pp.

[PS85] Preparata, Franco P. and Shamos, Michael Ian, *Computational Geometry: An Introduction,* Texts and Monographs in Computer Science, Springer-Verlag, New York-Berlin, 1985.

[Ren87] Renegar, Jim, *"On the Worst Case Arithmetic Complexity of Approximating Zeros of Systems of Polynomials,"* Technical Report, School of Operations Research and Industrial Engineering, Cornell University.

[Roj97a] Rojas, J. Maurice, *"Toric Laminations, Sparse Generalized Characteristic Polynomials, and a Refinement of Hilbert's Tenth Problem,"* Foundations of Computational Mathematics, selected papers of a conference, held at IMPA in Rio de Janeiro, January 1997, Springer-Verlag (1997).

[Roj97b] ________________, *"Linear Algebra for Large Nonlinear Algebra over the Reals,"* extended abstract, Sixth SIAM Conference on Applied Linear Algebra, Snowbird, Utah, October, 1997.

[Roj97c] ________________, *"Affine Elimination Theory,"* extended abstract, Proceedings of a Conference in Honor of David A. Buchsbaum, October, 1997, Northeastern University.

[Roj98a] ________________, *"Toric Intersection Theory for Affine Root Counting,"* Journal of Pure and Applied Algebra, to appear.

[Roj98b] ________________, *"Twisted Chow Forms and Toric Perturbations for Degenerate Polynomial Systems,"* submitted.

[Roy95] Roy, Marie-Françoise, *"Basic Algorithms in Real Algebraic Geometry and their Complexity: from Sturm's Theorem to the Existential Theory of Reals,"* Lectures in Real Geometry (Madrid, 1994), pp. 1–67, de Gruyter Exp. Math., 23, de Gruyter, Berlin, 1996.

[Stu93] Sturmfels, Bernd, *"Sparse Elimination Theory,"* In D. Eisenbud and L. Robbiano, editors, Proc. Computat. Algebraic Geom. and Commut. Algebra 1991, pages 377–396, Cortona, Italy, 1993, Cambridge Univ. Press.

[Stu94] ____________, *"On the Newton Polytope of the Resultant,"* Journal of Algebraic Combinatorics, 3: 207–236, 1994.

[Stu97] ____________, *"Introduction to Resultants,"* Notes from a lecture presented at an AMS short course on Applications of Computational Algebraic Geometry, San Diego, January 6–7, 1997.