On Isolating Roots in a Multiple Field Extension

Christina Katsamaki, Fabrice Rouillier

On Isolating Roots in a Multiple Field Extension

Christina Katsamaki
christina.katsamaki@inria.fr
INRIA Paris,
Sorbonne Université and Paris Université
F-75005, Paris, France

Fabrice Rouillier
Fabrice.Rouillier@inria.fr
INRIA Paris,
Sorbonne Université and Paris Université
F-75005, Paris, France

ABSTRACT

We address univariate root isolation when the polynomial's coefficients are in a multiple field extension. We consider a polynomial $F \in L[Y]$, where L is a multiple algebraic extension of $\mathbb{Q}$. We provide aggregate bounds for F and algorithmic and bit-complexity results for the problem of isolating its roots.

For the latter problem we follow a common approach based on univariate root isolation algorithms. For the particular case where F does not have multiple roots, we achieve a bit-complexity in $\widetilde{O}_B(nd^{2n+2}(d+n\tau))$, where d is the total degree and τ is the bitsize of the involved polynomials. In the general case we need to enhance our algorithm with a preprocessing step that determines the number of distinct roots of F. We follow a numerical, yet certified, approach that has bit-complexity $\widetilde{O}_B(n^2d^{3n+3}\tau + n^3d^{2n+4}\tau)$.

KEYWORDS

root isolation, field extension, bit-complexity

1 INTRODUCTION

We consider the problem of isolating the (complex) roots of a univariate polynomial over a multiple algebraic field extension –the coefficients of the polynomial are multivariate polynomial functions evaluated at algebraic numbers–. Solving in a field extension is a common problem in computational mathematics; for example it arises in the topology computation of plane curves [10, 14] or it can be seen as a sub-problem in the resolution of triangular systems [7, 26] and regular chains [3].

For $n \geq 2$, we consider $F_1 \in \mathbb{Z}[X_1], \ldots, F_n \in \mathbb{Z}[X_n]$ univariate polynomials of degree at most M and bitsize Λ and $F \in \mathbb{Z}[X_1, \ldots, X_n, Y]$ of total degree at most d and bitsize τ. We want to isolate the roots of the system

$$\left\{\begin{matrix} F_1(X_1) = 0, \ldots, F_n(X_n) = 0, \\ F(X_1, \ldots, X_n, Y) = 0\,. \end{matrix}\right\} \tag{1.1}$$

In theory, we can solve the system as follows: first, we isolate the roots of all the univariate polynomials $F_1, \ldots, F_n$. Then, for every root $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{C}^n$ of $\{F_1 = \cdots = F_n = 0\}$ we employ Pan's algorithm [21] for the approximate factorization of the univariate polynomial $F(\boldsymbol{x}, Y)$, with the worst case precision; the approximate factorization algorithm returns as many root approximations as the degree of $F(\boldsymbol{x}, Y)$ in Y is, and by utilizing the worst case precision, we can identify the root approximations that correspond to the same root. This method leads to good worst-case bit-complexity estimates (Rem. 4.3). Nevertheless, this is at the price of always requiring to perform computations using the maximum precision and it cannot lead to a practical algorithm. For example, if $n = 2$, $d = 10$, and $\tau \in O(1)$, then we have to work with $> 10^4$ bits in all of our computations. Our goal is to introduce an adaptive algorithm, depending on the multiplicities of the roots and on their pairwise distances, so we will follow a different approach.

Isolating the roots of the system in Eq. (1.1) has not been treated in the literature currently, but only in a simplified setting (e.g. [13, 22]). In [24] they consider the same problem when F does not have multiple roots; it is a generalization of a prior work for a simple algebraic extension [13, 23]. They propose three methods. The first one computes the minimal polynomial of the system and uses multivariate resultants. The second one is based on Sturm's algorithm and the third one on solving directly the polynomial using univariate root isolation algorithms, similarly to ours. The last method is the most efficient with a bit-complexity of $\widetilde{O}_B(n^3N^{2n+3})$, where N is a bound on the size of the input polynomials, without any assumptions on the input. In [10], it is the first time the general problem, in the simple extension setting, is addressed in literature [14, 16, 19, 23]. The authors provide precise amortized separation bounds for the polynomial and complexity bounds for isolating the roots. We provide further details on their method in the sequel, since we share many ideas. We could also solve the system in Eq. (1.1) by applying a general algorithm for zero-dimensional square systems of an expected complexity in $\widetilde{O}_B\left((n+1)^{n(\omega+1)+2}N^{(\omega+2)n+2}\right)$, where ω denotes the exponent in the complexity of matrix multiplication [5]. However, such a method would not exploit the special structure of the system.

Our approach and contribution

We generalize the results of [10] for any $n > 1$. By following closely their techniques, we are able to provide amortized bounds on the separation of F. On solving the system of Eq. (1.1), the idea is to approximate the coefficients of F for every $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{C}^n$ that is root of $F_1 = \ldots F_n = 0$, up to a certain precision, so that to isolate the roots of $F(\boldsymbol{x}, Y)$, it suffices to isolate the roots of its approximation. The amortized bounds that we prove in Cor. 3.3 and Cor. 3.5, quantify the required precision. To find the roots of the approximation of $F(\boldsymbol{x}, Y)$ we can now use algorithms for univariate root isolation. Particularly, we employ the algorithm of [19] that builds upon the algorithm of approximate factorization of a polynomial of Pan [21]; if a univariate polynomial is of degree d, the approximate factorization algorithm returns d root approximations. Then, the approximations must be clustered in a way so that each cluster corresponds to a root, and it contains as many root approximations as the multiplicity of the corresponding root. In [19] they run Pan's algorithm multiple times with increasing precision. For a stopping criterion, they require as input the number of distinct roots of the polynomial. Therefore, in our case, we should also compute the number of distinct roots of $F(\boldsymbol{x}, Y)$ for every root $\boldsymbol{x} \in \mathbb{C}^n$ of $F_1 = \ldots = F_n = 0$. This dominates the total bit-complexity.

In Sec. 4, we compute the number of distinct roots of $F(\boldsymbol{x}, Y)$ using a numerical approach (Lem. 4.4). We compute the principal subresultant coefficients of F and $\frac{\partial F}{\partial Y}$ with respect to Y. Then, for every root $\boldsymbol{x}$ of $F_1 = \ldots = F_n = 0$, we approximate the principal subresultant coefficients up to the necessary precision so that we can determine their sign correctly. The index of the first non-zero subresultant coefficient gives the degree of the gcd of $F(\boldsymbol{x}, Y)$ and $\frac{\partial F(\boldsymbol{x},Y)}{\partial Y}$, and thus the number of distinct roots. The total bit-complexity of solving the system of Eq. (1.1) then is described in Thm. 4.1(ii). In Rem. 4.1, we give for simplicity the bound for the case when all the polynomials have degree at most d and bitsize τ, which is $\widetilde{O}_B\left(n^2 d^{3n+3}\tau + n^3 d^{2n+4}\tau\right)$. On the contrary, when the number of distinct roots of $F(\boldsymbol{x}, Y)$ is known for every $\boldsymbol{x}$, or when F is does not have multiple roots, we can isolate the roots of the system in $\widetilde{O}_B(nd^{2n+2}(d+n\tau))$ bit operations. This is to be compared with the result of [24]; it improves it by a factor of n.

In Sec.5 we apply the aggregate bounds of F on the 'Sum of Square Roots of Integers' Problem. Comparing the length of two paths in the Euclidean Travel Salesman Problem (TSP), also relates to this problem. It has been already studied through the separation bound computation of the associated polynomial system [6, 20]. Our approach matches the latter results, and, even more, the proven bounds are aggregate.

2 NOTATION AND PREREQUISITES

Let $n \in \mathbb{N}$. We use the abbreviation $[n]$ for the set $\{1, \ldots, n\}$. We denote vectors by boldface symbols. For a vector $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{C}^n$, we denote the vector $(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) \in \mathbb{C}^{n-1}$, $i \in [n]$ by $\boldsymbol{x}_{-i}$. We call absolute L-bit approximation of a real number a, a rational number $\tilde{a}$ such that $|a - \tilde{a}| < 2^{-L}$. We denote the arithmetic, resp. bit, complexity by O, resp. O_B and we use $\widetilde{O}$, resp. $\widetilde{O}_B$, to ignore (poly-) logarithmic factors.

For a polynomial $f = \sum_{i=1}^{d} a_i X^i \in \mathbb{C}[X]$ we denote its ℓ_1-norm by $\|f\|_1$, i.e., $\|f\|_1 = \sum_{i=1}^{d} |a_i|$, its ℓ_2-norm by $\|f\|_2$, i.e., $\|f\|_2 = \sqrt{\sum_{i=1}^{d} |a_i|^2}$ and its ℓ_∞-norm by $\|f\|_\infty$, i.e., $\|f\|_\infty = \max_{i\in\{0,\ldots,d\}} |a_i|$. We denote the leading coefficient of f by $\mathrm{lc}(f)$. The k-th derivative of f is denoted by $f^{(k)}$ and $f^{[k]} := \frac{f^{(k)}}{k!}$. When f has integer coefficients, the *bitsize* of the polynomial is defined as the logarithm of its ℓ_∞-norm. All the logarithms in the present paper are of base 2. A univariate polynomial with integer coefficients is of *size* (d, τ) when its degree is at most d and it has bitsize τ. Similarly, a multivariate polynomial with integer coefficients is of *size* (d, τ) when its total degree is at most d and it has bitsize τ.

Let $I = \langle f_1, \ldots, f_k \rangle$ be a polynomial ideal in $\mathbb{C}[X_1, \ldots, X_n]$, $k \in \mathbb{N}$. We denote the complex variety defined by I by $V_{\mathbb{C}}(I)$ or $V_{\mathbb{C}}(f_1, \ldots, f_k)$. For a $\boldsymbol{x} \in V_{\mathbb{C}}(I)$, we denote its multiplicity as root of I by $\mu_I(\boldsymbol{x})$. When I is generated by one polynomial $f \in \mathbb{C}[X]$, we write for simplicity $\mu_f(x)$ to denote the multiplicity of x as root of $I = \langle f \rangle$. The ideal $\langle f_1, \ldots, f_{i-1}, f_{i+1}, \ldots, f_k \rangle$, for $i \in \{1, \ldots, k\}$ is denoted by $I \setminus f_i$.

2.1 Univariate polynomials: some bounds and root isolation

Let a univariate polynomial $f \in \mathbb{C}[X]$ and $x \in V_{\mathbb{C}}(f)$. The *local separation of f at x* is

$$\mathrm{sep}(x, f) := \min_{y \in V_{\mathbb{C}}(f), y \neq x} |y - x|.$$

The *separation* of f is

$$\mathrm{sep}(f) := \min_{x \in V_{\mathbb{C}}(f)} \mathrm{sep}(x, f).$$

The *Mahler measure* of f is defined as

$$\mathcal{M}(f) := |\mathrm{lc}(f)| \prod_{x \in V_{\mathbb{C}}(f)} \max(1, |x|)^{\mu_f(x)}.$$

The following inequality bounds the Mahler measure of f by means of its ℓ_1 and ℓ_2 norms [1, Prop.10.8 and Prop.10.9]:

$$2^{-d}\|f\|_1 \leq \mathcal{M}(f) \leq \|f\|_2. \tag{2.1}$$

In particular, if $f \in \mathbb{Z}[X]$ and it is of size (d, τ), the previous inequality becomes

$$2^{-d}\|f\|_1 \leq \mathcal{M}(f) \leq \|f\|_2 \leq 2^{\tau + \log(d+1)}. \tag{2.2}$$

Following [10, Def. 2.3 and Prop. 2.4], we introduce the definition of the *generalized discriminant* of $f \in \mathbb{C}[X]$, which is

$$\mathrm{GDisc}(f) := \mathrm{lc}(f)^{d-2} \prod_{x \in V_{\mathbb{C}}(f)} f^{[\mu_f(x)]}(x)^{\mu_f(x)}.$$

It plays an important role in the expression of several bounds in the sequel. We also define

$$\mathrm{lGDisc}(f) := \sum_{x \in V_{\mathbb{C}}(f)} \mu_f(x) |\log(|f^{[\mu_f(x)]}(x)|)| \text{ and}$$

$$\mathrm{lsep}(f) := \sum_{x \in V_{\mathbb{C}}(f)} \mu_f(x) |\log(\mathrm{sep}(x, f))|.$$

The next proposition, provides a bound for $\mathrm{lsep}(f)$ by means of $\log \mathcal{M}(f)$ and $\mathrm{lGDisc}(f)$.

Proposition 2.1 ([10, Prop. 2.7]). *For a polynomial $f \in \mathbb{C}[X]$ of degree d with $|lc(f)| \geq 1$, it holds that*

$$\mathrm{lsep}(f) \in O(d^2 + d \log \mathcal{M}(f) + \mathrm{lGDisc}(f)).$$

To isolate the roots of a univariate polynomial with coefficients in $\mathbb{C}$ we will use the algorithm of Mehlhorn et al. [19]. The algorithm requires that the number k of distinct roots is known. It first computes an approximate factorization of the polynomial using Pan's algorithm [21] with an initial precision. Assuming that the polynomial is of degree d, from the approximate factorization one obtains approximations $\tilde{z}_1, \ldots, \tilde{z}_d$ of the roots. Then, the roots are grouped in k clusters based on geometric vicinity. Every cluster is enclosed by a disc, each one corresponding to a root. If the discs are disjoint and each one contains the same number of root approximations as the multiplicity of the corresponding root, then the algorithm terminates. Otherwise, the factorization is repeated with increased precision. The next proposition summarizes their result.

Proposition 2.2 ([19, Thm. 3], [10, Prop. 2.22]). *Let $f(x) \in \mathbb{C}[x]$ of degree $d \geq 2$, for whom it holds that $1/4 \leq |lc(f)| \leq 1$. We assume that the number of distinct roots of f is known. We can compute isolating discs for all $x \in V_{\mathbb{C}}(f)$, as well as their multiplicities, in*

$$\widetilde{O}_B\big(d^3 + d^2 \log \mathcal{M}(f) + d\mathrm{lGDisc}(f)\big) .$$

As input, we need an oracle giving an absolute L-bit approximation of the coefficients of f with L bounded by

$$\widetilde{O}\big(d \log \mathcal{M}(f) + \mathrm{lsep}(f) + \mathrm{lGDisc}(f)\big) .$$

2.2 Evaluation of polynomials

If we want to evaluate a univariate polynomial $f \in \mathbb{C}[X]$ of degree d at some numbers $a_1, \ldots, a_D \in \mathbb{C}, D \in \mathbb{N}$, we can use multipoint evaluation [15]. When $D > d$, we have to repeat multipoint-evaluation $\lceil \frac{D}{d} \rceil$ times. When $D \leq d$ we have the following theorem:

Theorem 2.3 ([15, Thm.9]). *Let $f \in \mathbb{C}[X]$ be a polynomial of degree d, with absolute value of coefficients at most 2^τ, and let $a_1, \ldots, a_d \in \mathbb{C}$ be complex points with absolute values bounded by 2^Γ, where $\Gamma \geq 1$. Then, approximate multipoint evaluation up to a precision of 2^{-L} for some integer $L \geq 0$, that is, computing $\tilde{f}_i$ such that $|\tilde{f}_i - f(a_i)| \leq 2^{-L}$ for all i, can be done in*

$$\widetilde{O}_B\left(d(L + \tau + d\Gamma)\right)$$

bit-operations. The precision demand on f and the points a_i is bounded by $L + \widetilde{O}(\tau + d\Gamma)$ bits.

Now, we want to evaluate a multivariate polynomial $f \in \mathbb{C}[\boldsymbol{X}]$ at $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_D \in \mathbb{C}^n$. As discussed in [25], multipoint evaluation in the multivariate case is not an elementary extension of the univariate case, unless the evaluation points have good properties. In particular, when the evaluation points belong in a set of the form $S_1 \times \cdots \times S_n$, it is advantageous to perform multipoint evaluation at each coordinate one by one. The advantage comes from the fact that the number of different values in each coordinate is $|S_i|$, whereas the evaluation points are in total $\prod_{i=1}^n |S_i|$.

Proposition 2.4. *Let $f \in \mathbb{C}[\boldsymbol{X}]$ be a polynomial of degree d in each variable with absolute value of coefficients at most 2^τ and $S = S_1 \times \cdots \times S_n \subset \mathbb{C}^n$ be a set of complex points with absolute values bounded by 2^Γ, where $\Gamma \geq 1$, and $|S_i| \leq M$. Then, approximate multipoint evaluation up to a precision of 2^{-L} for some integer $L \geq 0$, that is, computing $\tilde{f}_i$ such that $|\tilde{f}_i - f(\boldsymbol{a}_i)| \leq 2^{-L}$ for all i, can be done in*

$$\widetilde{O}_B\left(\max(M,d)^{n-1} \left\lceil \frac{M}{d} \right\rceil (ndL + n^2 d\tau + n^3\, d^2\, \Gamma)\right)$$

bit-operations. The precision demand on f and the points a_j is bounded by $L + \widetilde{O}\left(n\tau + n^2 d\Gamma\right)$ bits.

Proof. For any $k = 1, \ldots, n-1$, we can write f as a polynomial in the variables $X_{k+1}, \ldots, X_n$ as

$$f(\boldsymbol{X}) = \sum_{\substack{e_i \in \{0,\ldots,d\},\\ i=k+1,\ldots,n}} f_{e_{k+1},\ldots,e_n}(X_1, \ldots, X_k)\ X_{k+1}^{e_{k+1}} \cdots X_n^{e_n}.$$

Then, $|f_{e_{k+1},\ldots,e_n}(X_1, \ldots, X_k)| \leq d^k \cdot 2^\tau \cdot 2^{kd\Gamma}$. In particular, for $k = n-1$ and $\mathbf{a}_{-n} \in S_1 \times \ldots S_{n-1}$, $f(\mathbf{a}_{-n}, X_n) = \sum_{i=0}^d f_i(\mathbf{a}_{-n})\, X_n^i$, with $|f_i(\mathbf{a}_{-n})| \leq 2^{\widetilde{O}(\tau + (n-1)d\Gamma)}$. Evaluating $f(\mathbf{a}_{-n}, X_n)$ at S_n with precision L can be done in $\widetilde{O}_B(d(L + \tau + (n-1)d\Gamma))$ with a precision demand on $f_i(\mathbf{a}_{-n})$ and on the points $a_n \in S_n$ in $L + \widetilde{O}(\tau + (n-1)\, d\, \Gamma)$ bits.

Recursively, for any $k \in \{1, \ldots, n-1\}$ and $\mathbf{a}^{k-1} := (a_1, \ldots, a_{k-1}) \in S_1 \times \cdots \times S_{k-1}$, we need to evaluate the polynomials $f_{e_{k+1},\ldots,e_n}(\mathbf{a}^{k-1}, X_k)$ at S_k with precision $L + (n-k)\tau + d\Gamma \sum_{i=k}^{n-1} i$. Since the polynomial has coefficients with absolute value bounded by $\tau + (k-1)d\Gamma$, the required precision on the coefficients and on the points in S_k is in $L + \widetilde{O}((n-k+1)\tau + \sum_{i=k-1}^{n-1} id\Gamma)$. For a polynomial $f_{e_{k+1},\ldots,e_n}$ this requires at most $M^{k-1} \cdot \lceil \frac{M}{d} \rceil$ multipoint evaluations of cost $\widetilde{O}_B(d(L + n\tau + n^2 d\Gamma))$ each one. For a fixed k there are at most $(d+1)^{n-k}$ polynomials $f_{e_{k+1},\ldots,e_n}$ to be evaluated, so this yields a complexity in $\widetilde{O}_B(d^{n-k} M^{k-1} \lceil \frac{M}{d} \rceil d(L + n\tau + n^2 d\Gamma))$ bit-operation. By summing for all $k = 1, \ldots, n-1$ we obtain a total bit-complexity in

$$\widetilde{O}_B\left(\max(M,d)^{n-1} \left\lceil \frac{M}{d} \right\rceil nd(L + n\tau + n^2 d\Gamma)\right) ,$$

to compute all the evaluations with an error bounded by 2^{-L} and a required precision of all the coordinates of the points in S bounded by $L + \widetilde{O}(n\tau + n^2 d\Gamma)$ bits. □

Notice that in both Thm. 2.3 and Prop. 2.4, the existence of an oracle providing the necessary approximations is assumed.

3 AMORTIZED BOUNDS FOR POLYNOMIALS IN A MULTIPLE FIELD EXTENSION

Let $F_1 \in \mathbb{Z}[X_1], \ldots, F_n \in \mathbb{Z}[X_n]$ be univariate polynomials of size (M, Λ) and $F \in \mathbb{Z}[\boldsymbol{X}, Y]$ of size (d, τ). We consider the ideals

$$\mathcal{I} = \langle F_1, \ldots, F_n \rangle \subset \mathbb{Z}[X_1, \ldots, X_n] \quad \text{and}$$
$$\mathcal{J} = \langle F_1, \ldots, F_n, F \rangle \subset \mathbb{Z}[X_1, \ldots, X_n, Y]. \tag{3.1}$$

For $\boldsymbol{x} \in V_{\mathbb{C}}(\mathcal{I})$, let $F_{\boldsymbol{x}}(Y) := F(\boldsymbol{x}, Y)$. We prove aggregate separation bounds for the roots of F in $(\mathbb{Z}[X_1, \ldots, X_n])[Y]$. We closely follow [10], where they treat the simple extension case, and we generalize their results to the n-variate field extension. We use Lem. 3.2 and Lem. 3.4, which are generalizations of Prop. 2.10 and Prop. 3.3 of [10] respectively, as building blocks for our proofs. Lem. 3.2 gives upper and lower bounds on the product of the evaluations of n-variate polynomials at all points in $V_{\mathbb{C}}(\mathcal{I})$ and in Lem. 3.4 the evaluation is of a set of $n+1$-variate polynomials at all points in $V_{\mathbb{C}}(\mathcal{J})$.

First, due to the special structure of the ideals $\mathcal{I}$ and $\mathcal{J}$, we have the following result on the multiplicities of the roots of the corresponding varieties. It will be used in the proof of Lem. 3.2.

Lemma 3.1 ([12, Prop.3], [27]). *Let $\mathcal{I}$ and $\mathcal{J}$ be the ideals of Eq. (3.1). For any $\boldsymbol{x} \in V_{\mathbb{C}}(\mathcal{I})$ and any $i \in [n]$ it holds that $\mu_{\mathcal{I}}(\boldsymbol{x}) = \mu_{F_i}(x_i) \cdot \mu_{\mathcal{I}:F_i}(\boldsymbol{x}_{-i})$. Moreover, for any $(\boldsymbol{x}, y) \in V_{\mathbb{C}}(\mathcal{J})$, it holds that $\mu_{\mathcal{J}}(\boldsymbol{x}, y) = \mu_{\mathcal{I}}(\boldsymbol{x}) \cdot \mu_{F_{\boldsymbol{x}}}(y)$.*

Lemma 3.2. *Let $\mathcal{I}$ be the ideal of Eq. (3.1) and $G_1, \ldots, G_m \in \mathbb{Z}[X_1, \ldots, X_n]$ of sizes (δ, σ).*
(i) Let $A \subseteq V_{\mathbb{C}}(\mathcal{I})$ such that for every $\boldsymbol{x} \in A$, there exists an index $i(\boldsymbol{x}) \in [m]$ such that $G_{i(\boldsymbol{x})}(\boldsymbol{x}) \neq 0$. Then,

$$\sum_{\boldsymbol{x} \in A} \mu_{\mathcal{I}}(\boldsymbol{x}) \log(|G_{i(\boldsymbol{x})}(\boldsymbol{x})|) \in \widetilde{O}\left(M^n(n+\sigma) + n\delta M^{n-1}\Lambda\right) .$$

(ii) If for every $\boldsymbol{x} \in V_{\mathbb{C}}(\mathcal{I})$ there exists an index $i \in [m]$ with $G_{i(\boldsymbol{x})}(\boldsymbol{x}) \neq 0$, then we denote by $i(\boldsymbol{x})$ the smallest such index. In this case,

$$\sum_{\boldsymbol{x}\in V_{\mathbb{C}}(\mathcal{I})} \mu_{\mathcal{I}}(\boldsymbol{x})\left|\log(|G_{i(\boldsymbol{x})}(\boldsymbol{x})|)\right| \in \widetilde{O}\left(M^n(n+\sigma)+n\delta M^{n-1}\Lambda\right).$$

Proof. (i) For any $\boldsymbol{x} = (x_1, \ldots, x_n) \in A$,

$$|G_{i(\boldsymbol{x})}(\boldsymbol{x})| \leq \binom{\delta+n}{n} 2^{\sigma} \prod_{j=1}^{n} \max\{1, |x_j|\}^{\delta},$$

since the number of monomials in $\mathbb{Z}[X_1, \ldots, X_n]$ of degree less than or equal to δ is $\binom{\delta+n}{n}$, the absolute value of every coefficient is $\leq 2^{\sigma}$ and every x_j is of degree at most δ. Therefore,

$$\prod_{\boldsymbol{x}\in A} |G_{i(\boldsymbol{x})}(\boldsymbol{x})|^{\mu_{\mathcal{I}}(\boldsymbol{x})} \leq \prod_{\boldsymbol{x}\in A} \left(\binom{\delta+n}{n} 2^{\sigma} \prod_{j=1}^{n} \max\{1, |x_j|\}^{\delta}\right)^{\mu_{\mathcal{I}}(\boldsymbol{x})}. \tag{3.2}$$

Since $\sum_{\boldsymbol{x}\in A} \mu_{\mathcal{I}}(\boldsymbol{x}) \leq M^n$ and $\binom{\delta+n}{n} \in O((\delta+n)^n)$, we have:

$$\prod_{\boldsymbol{x}\in A} \left(\binom{\delta+n}{n} 2^{\sigma}\right)^{\mu_{\mathcal{I}}(\boldsymbol{x})} \in 2^{\widetilde{O}(M^n(n+\sigma))}. \tag{3.3}$$

For $j \in [n]$ it holds that

$$\begin{aligned}
\prod_{\boldsymbol{x}\in A} \max\{1, |x_j|\}^{\delta\,\mu_{\mathcal{I}}(\boldsymbol{x})} &= \prod_{\boldsymbol{x}\in A} \left(\max\{1, |x_j|\}^{\mu_{F_j}(x_j)}\right)^{\delta\,\mu_{\mathcal{I}\setminus F_j}(\boldsymbol{x}_{-j})} \\
&= \prod_{x_j | \boldsymbol{x}\in A} \left(\max\{1, |x_j|\}^{\mu_{F_j}(x_j)}\right)^{\delta \sum_{\boldsymbol{x}_{-j}|\boldsymbol{x}\in A} \mu_{\mathcal{I}\setminus F_j}(\boldsymbol{x}_{-j})} \\
&\leq \prod_{x_j | \boldsymbol{x}\in A} \left(\max\{1, |x_j|\}^{\mu_{F_j}(x_j)}\right)^{\delta M^{n-1}} \leq \mathcal{M}(F_j)^{\delta M^{n-1}},
\end{aligned}$$

where the first equality follows from Lem. 3.1 and the first inequality from the fact that $\sum_{\boldsymbol{x}_{-j}|\boldsymbol{x}\in A} \mu_{\mathcal{I}\setminus F_j}(\boldsymbol{x}_{-j}) \leq M^{n-1}$. Note that the last inequality is true since the coefficients of F_j are in $\mathbb{Z}$ and so the absolute value of the leading coefficient of F_j is greater or equal to 1. We have that $\mathcal{M}(F_j) \in 2^{O(\Lambda+\log M)}$, following Eq. (2.2). Therefore,

$$\prod_{\boldsymbol{x}\in A} \max\{1, |x_j|\}^{\delta\mu_{\mathcal{I}}(\boldsymbol{x})} \in 2^{\widetilde{O}(\delta M^{n-1}\Lambda)}. \tag{3.4}$$

From the equations (3.2), (3.3) and (3.4), we conclude.
(ii) Let $A = \{\boldsymbol{x} \in V_{\mathbb{C}}(\mathcal{I}) \mid |G_{i(\boldsymbol{x})}(\boldsymbol{x})| \geq 1\}$. Then, we can write:

$$\begin{aligned}
\sum_{\boldsymbol{x}\in V_{\mathbb{C}}(\mathcal{I})} \mu_{\mathcal{I}}(\boldsymbol{x})\left|\log(|G_{i(\boldsymbol{x})}(\boldsymbol{x})|)\right| &= 2\sum_{\boldsymbol{x}\in A} \mu_{\mathcal{I}}(\boldsymbol{x}) \log(|G_{i(\boldsymbol{x})}(\boldsymbol{x})|) - \\
&- \sum_{\boldsymbol{x}\in V_{\mathbb{C}}(\mathcal{I})} \mu_{\mathcal{I}}(\boldsymbol{x}) \log(|G_{i(\boldsymbol{x})}(\boldsymbol{x})|).
\end{aligned} \tag{3.5}$$

Using (i) of this lemma, we obtain an upper bound for the first term of the previous sum. Thus, we only need to compute a lower bound for $\sum_{\boldsymbol{x}\in V_{\mathbb{C}}(\mathcal{I})} \mu_{\mathcal{I}}(\boldsymbol{x}) \log(|G_{i(\boldsymbol{x})}(\boldsymbol{x})|)$. Let $G(X_1, \ldots, X_n, U) = G_1(X_1, \ldots, X_n) + G_2(X_1, \ldots, X_n)U + \cdots + G_m(X_1, \ldots, X_n)U^{m-1}$. We consider $\mathrm{res}_{\boldsymbol{X}}(F_1, \ldots, F_n, G)$, the multivariate resultant where we eliminate $\boldsymbol{X}$. Using the Poisson formula [1, Thm. 4.14] we can write

$$\begin{aligned}
\mathrm{res}_{\boldsymbol{X}}(F_1, \ldots, F_n, G) &= \mathrm{res}_{\boldsymbol{X}}(\mathrm{lc}(F_1)X_1^{\deg(F_1)}, \ldots, \mathrm{lc}(F_n)X_n^{\deg(F_n)})^{O(\delta)} \cdot \\
&\cdot \prod_{\boldsymbol{x}\in V_{\mathbb{C}}(\mathcal{I})} G(\boldsymbol{x}, U)^{\mu_{\mathcal{I}}(\boldsymbol{x})} = \\
&= \left(\mathrm{res}_{\boldsymbol{X}}(X_1^{\deg(F_1)}, \ldots, X_n^{\deg(F_n)}) \prod_{j=1}^{n} \mathrm{lc}(F_j)^{\prod_{k\in[n],k\neq j}\deg(F_k)}\right)^{O(\delta)} \cdot \\
&\cdot \prod_{\boldsymbol{x}\in V_{\mathbb{C}}(\mathcal{I})} G(\boldsymbol{x}, U)^{\mu_{\mathcal{I}}(\boldsymbol{x})} = \\
&= \left(\prod_{j=1}^{n} \mathrm{lc}(F_j)^{\prod_{k\in[n],k\neq j}\deg(F_k)}\right)^{O(\delta)} \prod_{\boldsymbol{x}\in V_{\mathbb{C}}(\mathcal{I})} G(\boldsymbol{x}, U)^{\mu_{\mathcal{I}}(\boldsymbol{x})},
\end{aligned}$$

which is a polynomial in $\mathbb{Z}[U]$; it is not identically zero, since by the hypothesis, for every $\boldsymbol{x} \in V_{\mathbb{C}}(\mathcal{I})$, $G(\boldsymbol{x}, U)$ is not identically zero. The absolute value of the constant term of $\mathrm{res}_{\boldsymbol{X}}(F_1, \ldots, F_n, G)$ is

$$\left|\prod_{j=1}^{n} \mathrm{lc}(F_j)^{\prod_{k\in[n],k\neq j}\deg(F_k)}\right|^{O(\delta)} \prod_{\boldsymbol{x}\in V_{\mathbb{C}}(\mathcal{I})} |G_{i(\boldsymbol{x})}(\boldsymbol{x})|^{\mu_{\mathcal{I}}(\boldsymbol{x})} \geq 1. \tag{3.6}$$

Since $\left|\prod_{j=1}^{n} \mathrm{lc}(F_j)^{\prod_{k\in[n],k\neq j}\deg(F_k)}\right| \in 2^{O(n\Lambda M^{n-1})}$, it follows from Eq. (3.6) that

$$\prod_{\boldsymbol{x}\in V_{\mathbb{C}}(\mathcal{I})} |G_{i(\boldsymbol{x})}(\boldsymbol{x})|^{\mu_{\mathcal{I}}(\boldsymbol{x})} \in 2^{-O(n\delta\Lambda M^{n-1})}. \tag{3.7}$$

So, by applying part (i) of the lemma and Eq. (3.7) to Eq. (3.5), we conclude. □

The following corollary, provides an amortized bound on the sum of the logarithms (bitsize) of the Mahler measures of the polynomials $F_{\boldsymbol{x}}(Y)$, for all $\boldsymbol{x} \in V_{\mathbb{C}}(\mathcal{I})$ (counting multiplicities).

Corollary 3.3 (Amortized Mahler measure). *Let $\mathcal{I}$ be the ideal of Eq.* (3.1)*. Then,*

$$\sum_{\boldsymbol{x}\in V_{\mathbb{C}}(\mathcal{I})} \mu_{\mathcal{I}}(\boldsymbol{x}) \log \mathcal{M}(F_{\boldsymbol{x}}) \in \widetilde{O}\left(M^n(n+\tau+d) + nM^{n-1}d\Lambda\right).$$

Proof. We write $F(\boldsymbol{X}, Y) = f_d(\boldsymbol{X})Y^d + \cdots + f_0(\boldsymbol{X})$. For any $\boldsymbol{x} \in V_{\mathbb{C}}(\mathcal{I})$, following Eq. (2.2), it holds

$$2^{-d}\|F_{\boldsymbol{x}}\|_1 \leq \mathcal{M}(F_{\boldsymbol{x}}) \leq \|F_{\boldsymbol{x}}\|_2, \tag{3.8}$$

since the degree of any $F_{\boldsymbol{x}}(Y)$ is $\leq d$. Let

$$\begin{aligned}
|f_{M(\boldsymbol{x})}(\boldsymbol{x})| &:= \max_{j\in\{0,\ldots d\}} |f_j(\boldsymbol{x})|, \\
|f_{m(\boldsymbol{x})}(\boldsymbol{x})| &:= \min_{j\in\{0,\ldots d\}|f_j(\boldsymbol{x})\neq 0} |f_j(\boldsymbol{x})|.
\end{aligned}$$

Now, Eq. (3.8) gives

$$2^{-d}\,|f_{m(\boldsymbol{x})}(\boldsymbol{x})| \leq \mathcal{M}(F_{\boldsymbol{x}}) \leq \sqrt{d+1}\,|f_{M(\boldsymbol{x})}(\boldsymbol{x})|.$$

If we consider for all $\boldsymbol{x} \in V_{\mathbb{C}}(\mathcal{I})$ (counting multiplicities), then

$$2^{-dM^n} \prod_{\boldsymbol{x}\in V_{\mathbb{C}}(\mathcal{I})} |f_{m(\boldsymbol{x})}(\boldsymbol{x})|^{\mu_{\mathcal{I}}(\boldsymbol{x})} \leq \prod_{\boldsymbol{x}\in V_{\mathbb{C}}(\mathcal{I})} \mathcal{M}(F_{\boldsymbol{x}})^{\mu_{\mathcal{I}}(\boldsymbol{x})} \leq \leq \sqrt{d+1}^{M^n} \prod_{\boldsymbol{x}\in V_{\mathbb{C}}(\mathcal{I})} |f_{M(\boldsymbol{x})}(\boldsymbol{x})|^{\mu_{\mathcal{I}}(\boldsymbol{x})}. \quad (3.9)$$

We can bound the products on each side of the inequality in Eq. (3.9) by Lem. 3.2. This concludes the proof. □

The following lemma is an analog of Lem. 3.2, but in the case where we evaluate over $V_{\mathbb{C}}(\mathcal{J})$.

Lemma 3.4. *Let $\mathcal{I}$ and $\mathcal{J}$ be the ideals of Eq. (3.1) and $G_1, \ldots, G_m \in \mathbb{Z}[X_1, \ldots, X_n, Y]$ of sizes (δ, σ).*
(i) Let $A \subseteq V_{\mathbb{C}}(\mathcal{J})$ such that for every $(\boldsymbol{x}, y) \in A$, there exists an index $i(\boldsymbol{x}, y) \in [m]$ such that $G_{i(\boldsymbol{x},y)}(\boldsymbol{x}, y) \neq 0$. Then,

$$\sum_{(\boldsymbol{x},y)\in A} \mu_{\mathcal{J}}(\boldsymbol{x}, y) \log |G_{i(\boldsymbol{x},y)}(\boldsymbol{x}, y)| \in \widetilde{O}\left(M^n(d(n+\sigma) + \delta(n+\tau+d)) + n\delta d M^{n-1}\Lambda\right).$$

(ii) Supposing that for every $(\boldsymbol{x}, y) \in V_{\mathbb{C}}(\mathcal{J})$ there exists an index $i \in [m]$ with $G_i(\boldsymbol{x}, y) \neq 0$, we denote by $i(\boldsymbol{x}, y)$ the smallest such index. Then,

$$\sum_{(\boldsymbol{x},y)\in V_{\mathbb{C}}(\mathcal{J})} \mu(\boldsymbol{x}, y) \left|\log(|G_{i(\boldsymbol{x},y)}(\boldsymbol{x}, y)|)\right| \in \widetilde{O}\left(M^n(d(n+\sigma) + \delta(n+\tau+d)) + n\delta d M^{n-1}\Lambda\right).$$

Proof. (i) For any $(\boldsymbol{x}, y) \in A$,

$$|G_{i(\boldsymbol{x},y)}(\boldsymbol{x}, y)| \leq \binom{\delta+n+1}{n+1} 2^{\sigma} \prod_{i=1}^{n} \max\{1, |x_i|\}^{\delta} \max\{1, |y|\}^{\delta},$$

since the number of monomials in $\mathbb{Z}[X_1, \ldots, X_n, Y]$ of degree less than or equal to δ is $\binom{\delta+n+1}{n+1}$. We have that:

$$\prod_{(\boldsymbol{x},y)\in A} \left(\binom{\delta+n+1}{n+1} 2^{\sigma}\right)^{\mu_{\mathcal{J}}(\boldsymbol{x},y)} \in 2^{\widetilde{O}(M^n d(n+\sigma))}, \quad (3.10)$$

since $\sum_{(\boldsymbol{x},y)\in A} \mu_{\mathcal{J}}(\boldsymbol{x}, y) \leq M^n d$. For $j = 1, \ldots, n$:

$$\begin{aligned}
\prod_{(\boldsymbol{x},y)\in A} \max\{1, |x_j|\}^{\delta\mu_{\mathcal{J}}(\boldsymbol{x},y)} &\leq \prod_{(\boldsymbol{x},y)\in V_{\mathbb{C}}(\mathcal{J})} \max\{1, |x_j|\}^{\delta\mu_{\mathcal{J}}(\boldsymbol{x},y)} \\
&= \prod_{(\boldsymbol{x},y)\in V_{\mathbb{C}}(\mathcal{J})} \max\{1, |x_j|\}^{\delta\mu_{F_j}(x_j)\mu_{\mathcal{I}_{-j}}(\boldsymbol{x}_{-j})\mu_{F_{\boldsymbol{x}}}(y)} \\
&\leq \prod_{\boldsymbol{x}\in V_{\mathbb{C}}(\mathcal{I})} \max\{1, |x_j|\}^{\delta\mu_{F_j}(x_j)\mu_{\mathcal{I}_{-j}}(\boldsymbol{x}_{-j})d} \\
&= \prod_{x_j\in V_{\mathbb{C}}(F_j)} \max\{1, |x_j|\}^{\delta\mu_{F_j}(x_j)\sum_{\boldsymbol{x}_{-j}|\boldsymbol{x}\in V_{\mathbb{C}}(\mathcal{I})} \mu_{\mathcal{I}\setminus F_j}(\boldsymbol{x}_{-j})d} \\
&\leq \prod_{x_j\in V_{\mathbb{C}}(F_j)} \max\{1, |x_j|\}^{\delta\mu_{F_j}(x_j)M^{n-1}d} \leq \mathcal{M}(F_j)^{\delta M^{n-1}d},
\end{aligned}$$

where the first equality follows from Lem. 3.1 and the third inequality from the fact that $\sum_{\boldsymbol{x}_{-j}|\boldsymbol{x}\in V_{\mathbb{C}}(\mathcal{I})} \mu_{\mathcal{I}\setminus F_j}(\boldsymbol{x}_{-j}) \leq M^{n-1}$. Note that the last inequality is true since the coefficients of F_j are in $\mathbb{Z}$. Since $\mathcal{M}(F_j) \in 2^{O(\Lambda+\log M)}$, we have that

$$\prod_{(\boldsymbol{x},y)\in A} \max\{1, |x_j|\}^{\delta\mu_{\mathcal{J}}(\boldsymbol{x},y)} \in 2^{\widetilde{O}(\delta M^{n-1}d\Lambda)}. \quad (3.11)$$

Lastly, we have that

$$\prod_{\boldsymbol{x}\in V_{\mathbb{C}}(\mathcal{I})} |\mathrm{lc}(F_{\boldsymbol{x}}(Y)|^{\delta\mu_{\mathcal{I}}(\boldsymbol{x})} \cdot \prod_{(\boldsymbol{x},y)\in A} \max\{1, |y|\}^{\delta\mu_{\mathcal{J}}(\boldsymbol{x},y)} \leq$$

$$\leq \prod_{\boldsymbol{x}\in V_{\mathbb{C}}(\mathcal{I})} |\mathrm{lc}(F_{\boldsymbol{x}}(Y)|^{\delta\mu_{\mathcal{I}}(\boldsymbol{x})} \left(\prod_{y\in V_{\mathbb{C}}(F_{\boldsymbol{x}})} \max\{1, |y|\}^{\mu_{F_{\boldsymbol{x}}}(y)}\right)^{\delta\mu_{\mathcal{I}}(\boldsymbol{x})} \leq$$

$$\leq \prod_{\boldsymbol{x}\in V_{\mathbb{C}}(\mathcal{I})} \mathcal{M}(F_{\boldsymbol{x}})^{\delta\mu_{\mathcal{I}}(\boldsymbol{x})} \in 2^{\widetilde{O}(M^n\delta(n+\tau+d)+M^{n-1}\delta nd\Lambda)},$$

which follows from Cor.3.3. Also, from Lem. 3.2 we can bound the size of the factor $\prod_{\boldsymbol{x}\in V_{\mathbb{C}}(\mathcal{I})} |\mathrm{lc}(F_{\boldsymbol{x}}(Y)|^{\delta\mu_{\mathcal{I}}(\boldsymbol{x})}$ on the left-hand side of the previous equation, and thus, we have that

$$\prod_{(\boldsymbol{x},y)\in A} \max\{1, |y|\}^{\delta\mu_{\mathcal{J}}(\boldsymbol{x},y)} \in 2^{\widetilde{O}(M^n\delta(n+\tau+d)+M^{n-1}\delta nd\Lambda)}. \quad (3.12)$$

By putting together Eq. (3.10), Eq. (3.11) and Eq. (3.12) we can conclude.

(ii) As in the proof of Lem. 3.2, by the first part of the lemma for $A = \{(\boldsymbol{x}, y) \mid |G_{i(\boldsymbol{x},y)}(\boldsymbol{x}, y)| \geq 1\}$, we just need to find a lower bound for $\sum_{(\boldsymbol{x},y)\in V_{\mathbb{C}}(\mathcal{J})} \mu_{\mathcal{I}}(\boldsymbol{x})\mu_{F_{\boldsymbol{x}}}(y) \log(|G_{i(\boldsymbol{x},y)}(\boldsymbol{x}, y)|)$. Let $G(\boldsymbol{X}, Y, U) := G_1(\boldsymbol{X}, Y) + G_2(\boldsymbol{X}, Y)U + \cdots + G_m(\boldsymbol{X}, Y)U^{m-1}$. Let also $Q(\boldsymbol{X}, U) := \mathrm{res}_Y(G(\boldsymbol{X}, Y, U), F(\boldsymbol{X}, Y))$, be the resultant where we eliminate Y. Without loss of generality, we assume that the leading coefficient of $F(X_1, \ldots, X_n, Y)$ when considered as a polynomial in $\mathbb{Z}[X_1, \ldots, X_n][Y]$, is not canceled for any root of $\mathcal{I}$ (in the case where the leading coefficient is cancelled for some roots, F is replaced by a polynomial of smaller degree). So, the resultant is not the zero polynomial.

We consider $\mathrm{res}_{\boldsymbol{X}}(Q, F_1, \ldots, F_n)$, which is now the resultant where we eliminate $\boldsymbol{X}$. Using the Poisson formula [1, Thm. 4.14] we can write

$$\begin{aligned}
&\mathrm{res}_{\boldsymbol{X}}(Q(\boldsymbol{X}, U), F_1(X_1), \ldots, F_n(X_n)) = \\
&= \left(\prod_{j=1}^{n} \mathrm{lc}(F_j)^{\prod_{k\in[n],k\neq j}\deg(F_k)}\right)^{O(d\delta)} \prod_{\boldsymbol{x}\in V_{\mathbb{C}}(\mathcal{I})} Q(\boldsymbol{x}, U)^{\mu(\boldsymbol{x})} = \\
&= \left(\prod_{j=1}^{n} \mathrm{lc}(F_j)^{\prod_{k\in[n],k\neq j}\deg(F_k)}\right)^{O(d\delta)} \prod_{\boldsymbol{x}\in V_{\mathbb{C}}(\mathcal{I})} f_d(\boldsymbol{x})^{O(\delta)\mu_{\mathcal{I}}(\boldsymbol{x})} \cdot \\
&\qquad \cdot \prod_{y|(\boldsymbol{x},y)\in V_{\mathbb{C}}(\mathcal{J})} G(\boldsymbol{x}, y, U)^{\mu_{\mathcal{I}}(\boldsymbol{x})\mu(y)}.
\end{aligned}$$

The absolute value of the constant term of $\mathrm{res}_{\boldsymbol{X}}(Q(\boldsymbol{X}, U), F_1(X_1), \ldots, F_n(X_n)) \in \mathbb{Z}[U]$ is:

$$\left|\prod_{j=1}^{n} \mathrm{lc}(F_j)^{\prod_{k\in[n],k\neq j}\deg(F_k)}\right|^{O(d\delta)} \prod_{\boldsymbol{x}\in V_{\mathbb{C}}(\mathcal{I})} |f_d(\boldsymbol{x})|^{O(\delta)\mu_{\mathcal{I}}(\boldsymbol{x})} \cdot \prod_{(\boldsymbol{x},y)\in V_{\mathbb{C}}(\mathcal{J})} |G_{i(\boldsymbol{x},y)}(\boldsymbol{x}, y)|^{\mu_{\mathcal{I}}(\boldsymbol{x})\mu_{F_{\boldsymbol{x}}}(y)} \geq 1.$$

We have that $\left|\prod_{j=1}^{n} \mathrm{lc}(F_j)^{\prod_{k\in[n],k\neq j}\deg(F_k)}\right| \in 2^{O(n\Lambda M^{n-1})}$. From Lem. 3.2 (i) it follows that

$$\prod_{\boldsymbol{x}\in V_{\mathbb{C}}(\mathcal{I})} |f_d(\boldsymbol{x})|^{\mu_{\mathcal{I}}(\boldsymbol{x})} \in 2^{O(M^n(n+\tau)+ndM^{n-1}\Lambda)}$$

and thus

$$\prod_{(\boldsymbol{x},y)\in V_{\mathbb{C}}(\mathcal{J})} |G_{i(\boldsymbol{x},y)}(\boldsymbol{x},y)|^{\mu_{\mathcal{I}}(\boldsymbol{x})\mu_{F_{\boldsymbol{x}}}(y)} \in 2^{-O(\delta M^n(n+\tau)+nd\delta M^{n-1}\Lambda)} \tag{3.13}$$

So, by combining the first part of the lemma and Eq. (3.13), we conclude. □

Corollary 3.5 (Amortized bound on lGDisc and lsep). *Let $\mathcal{I}$ and $\mathcal{J}$ be the ideals of Eq.* (3.1). *Then,*

$$\sum_{\boldsymbol{x}\in V_{\mathbb{C}}(\mathcal{I})} \mu_{\mathcal{I}}(\boldsymbol{x})\,\mathrm{lGDisc}(F_{\boldsymbol{x}}) \in \widetilde{O}\left(dM^n(n+\tau+d)+nd^2M^{n-1}\Lambda\right)$$

and

$$\sum_{\boldsymbol{x}\in V_{\mathbb{C}}(\mathcal{I})} \mu_{\mathcal{I}}(\boldsymbol{x})\,\mathrm{lsep}(F_{\boldsymbol{x}}) \in \widetilde{O}\left(dM^n(n+\tau+d)+nd^2M^{n-1}\Lambda\right).$$

Proof. We can write

$$\sum_{\boldsymbol{x}\in V_{\mathbb{C}}(\mathcal{I})} \mu_{\mathcal{I}}(\boldsymbol{x})\mathrm{lGDisc}(F_{\boldsymbol{x}}) =$$

$$= \sum_{\boldsymbol{x}\in V_{\mathbb{C}}(\mathcal{I})} \mu_{\mathcal{I}}(\boldsymbol{x}) \sum_{y\in V_{\mathbb{C}}(F_{\boldsymbol{x}})} \mu_{F_{\boldsymbol{x}}}(y)|\log(|F_{\boldsymbol{x}}^{[\mu_{F_{\boldsymbol{x}}}(y)]}(y)|)| =$$

$$= \sum_{(\boldsymbol{x},y)\in V_{\mathbb{C}}(\mathcal{J})} \mu_{\mathcal{J}}(\boldsymbol{x},y)|\log(|F_{\boldsymbol{x}}^{[\mu_{F_{\boldsymbol{x}}}(y)]}(y)|)|$$

and then apply Lem. 3.4 for the family of polynomials $F_{\boldsymbol{x}}^{[k]}$, for $k=0,\ldots,d$. These polynomials are of size $(d,\widetilde{O}(\tau))$, therefore the first part follows. The second part is an immediate consequence of Prop. 2.1, Cor. 3.3 and the first part of this corollary. □

4 SOLVING IN A MULTIPLE FIELD EXTENSION

In this section, we study the complexity of isolating the roots of the system in Eq. (1.1). We first solve the univariate polynomials of the system and then, for every $\boldsymbol{x}\in V_{\mathbb{C}}(\mathcal{I})$, we will isolate the roots of $F_{\boldsymbol{x}}$. Following [10], we employ the univariate root isolation algorithm of Prop. 2.2. The main result of this section is summarized in the theorem that follows.

Theorem 4.1. *(i) If the number of distinct roots of $F_{\boldsymbol{x}}(Y)$ for every $\boldsymbol{x}\in V_{\mathbb{C}}(\mathcal{I})$ is known, then we compute isolating discs for all the roots and the corresponding multiplicities in*

$$\widetilde{O}_B\Big(nM^{n+1}d(n+\tau+d)+nM^nd^2(n\Lambda+d^n(n+\tau+d+\Lambda))+n^2d^{n+3}M^{n-1}\Lambda\Big).$$

(ii) If the number of distinct roots is not known, then we compute isolating discs for all the roots, together with the corresponding multiplicities in

$$\widetilde{O}_B\Big(\max(M,d^2)^{n-1}\left\lceil\frac{M}{d^2}\right\rceil((nM^n+n^2)d^5(\tau+n)+n^2\,d^6\,\Lambda(M^{n-1}+n))+$$
$$+nM^nd^{n+2}(n+\tau+d+\Lambda)+n^2d^{n+3}M^{n-1}\Lambda\Big).$$

Remark 4.1. *When $M=d$ and $\Lambda=\tau$, the bit-complexity bounds of the previous theorem become*

$$\widetilde{O}_B(nd^{2n+2}(d+n\tau)),$$

when the number of distinct roots of $F_{\boldsymbol{x}}(Y)$ is known for every $\boldsymbol{x}\in V_{\mathbb{C}}(\mathcal{I})$ (or when F is squarefree) and

$$\widetilde{O}_B\left(n^2d^{3n+3}\tau+n^3d^{2n+4}\tau\right),$$

otherwise.

Remark 4.2 (Number of distinct roots, numerically vs formally)**.** *Determining the number of distinct roots of every $F_{\boldsymbol{x}}$ in Thm. 4.1 dominates the total complexity. When $n=1$, a formal method involving univariate gcd computations can be used to find this number; the initial system is triangular and it can be efficiently decomposed into regular triangular systems, for whom the number of distinct roots over every $\boldsymbol{x}$ is constant [10, 17]. However, when $n>1$ the system is not triangular and decomposing it to a set of regular triangular systems (cf. extended gcd computation [8]) as for the case $n=1$, and thus loosing the original shape of the system, would require isolating roots of a triangular system in n variables. The latter problem is substantially more demanding than isolating the roots of n univariate polynomials. For this reason, we will follow a numerical approach to find the number of distinct roots of every $F_{\boldsymbol{x}}$.*

In the remark that follows, we compute the complexity of isolating the roots of the system of Eq. (1.1) using the algorithm of approximate factorization of Pan [21] with the maximal precision; instead of requiring the number of distinct roots of a univariate polynomial, if we approximate its roots with precision up to the separation bound, then the root approximations that are have pairwise distances smaller that the separation bound, will correspond to the same root. So, this is a method that avoids computing the number of distinct roots of the univariate polynomials $F_{\boldsymbol{x}}$. Nevertheless, it is a theoretical approach which brings about practical limitations in contrast to our adaptive method.

Remark 4.3 (Pan's algorithm with maximal precision)**.** *On isolating the roots of $F_{\boldsymbol{x}}$ for every $V_{\mathbb{C}}(\mathcal{I})$, instead of employing the algorithm of Prop. 2.2, that requires knowing the number of distinct roots, we can use Pan's algorithm of approximate factorization with precision up to the separation bound of the roots of the initial system of Eq. (1.1). Then, for every $\boldsymbol{x}\in V_{\mathbb{C}}(\mathcal{I})$, we approximate $F(\boldsymbol{x},Y)$ up to $\sum_{\boldsymbol{x}\in V_{\mathbb{C}}(\mathcal{I})}\mu_{\mathcal{I}}(\boldsymbol{x})\,\mathrm{lsep}(F_{\boldsymbol{x}})\in\widetilde{O}(M^nd(n+\tau+d)+nd^2M^{n-1}\Lambda))$ bits. From Prop. 2.4 this is done in*

$$\widetilde{O}_B\left(\max(M,d)^{n-1}\left\lceil\frac{M}{d}\right\rceil(nM^nd^2(n+\tau+d)+n^2d^3M^{n-1}\Lambda+n^2d\tau+n^3\,d^2\tau)\right)$$

since there are d polynomials to evaluate. Then, for every $F_{\boldsymbol{x}}(Y)$, Pan's algorithm runs in

$$\widetilde{O}_B\left(M^{2n}d^2(n+\tau+d)+nd^3M^{2n-1}\Lambda\right)$$

and returns isolating intervals for all the roots.

Before presenting the proof of Thm. 4.1, we need some intermediate results. The next lemma, gives upper and lower bounds on the evaluation of a polynomial over an algebraic number. For simplicity, we ignore the logarithmic factors.

Lemma 4.2. *For every $\boldsymbol{x} \in V_{\mathbb{C}}(\mathcal{I})$ and a polynomial $b(\boldsymbol{X}) \in \mathbb{Z}[\boldsymbol{X}]$ of size (δ, σ), it holds that*

$$2^{-\widetilde{O}(M^n(\sigma+n)+nM^{n-1}\delta\Lambda)} \leq |b(\boldsymbol{x})| \leq 2^{\widetilde{O}(n\delta\Lambda+\sigma)}.$$

Proof. For the upper bound, we have that

$$|b(\boldsymbol{x})| \leq \binom{\delta+n}{n} 2^{\sigma} \prod_{i\in[n]} \max(1,|x_i|)^{\delta} \leq \binom{\delta+n}{n} 2^{\sigma} 2^{\delta O(\Lambda)n},$$

since, for $i \in [n]$, $\max(1,|x_i|) \leq 2^{O(\Lambda)}$ from the Cauchy bound [1, Lem. 10.2]. For the lower bound, we follow the technique of u-resultant and consider the system:

$$F_0(\boldsymbol{X}, Y) = F_1(X_1) = \cdots = F_n(X_n) = 0, \tag{4.1}$$

where $F_0(\boldsymbol{X}, Y) = Y - b(\boldsymbol{X})$ and Y is a new variable. We consider the resultant of the previous system that eliminates $\boldsymbol{X}$. Then,

$$R(Y) := \mathrm{res}_{\boldsymbol{X}}(F_0, \ldots, F_n) = \mathrm{lc}(R) \prod_{\boldsymbol{a}\in V_{\mathbb{C}}(\mathcal{I})} (Y - b(\boldsymbol{a}))^{\mu_{\mathcal{I}}(\boldsymbol{a})} \in \mathbb{Z}[Y].$$

We will find an upper bound on the bitsize of R. To this scope, we follow the proof of the DMM bound in [11] (see also the proof of the sparse resultant's height bound in [18]). For $i = 0, \ldots, n$, let Q_i be the Newton polytope of F_i. We denote by $\#Q_i$ the number of lattice points in the closed polytope Q_i and by M_i the mixed volume of all these polytopes except from Q_i. The resultant R is a univariate polynomial in Y, with coefficients homogeneous polynomials in the coefficients of the polynomials of the system in Eq. (4.1):

$$R(Y) = \ldots + \rho_k Y^k \mathbf{c}_{0,k}^{M_0-k} \mathbf{c}_{1,k}^{M_1} \cdots \mathbf{c}_{n,k}^{M_n} + \ldots,$$

where $\rho_k \in \mathbb{Z}$, $\mathbf{c}_{i,k}^{M_i}$ is a monomial in the coefficients of F_i with total degree M_i, for $i \in [n]$, and $\mathbf{c}_{0,k}^{M_0-k}$ is a monomial in the coefficients of F_0 of total degree $M_0 - k$. It holds that

$$\begin{aligned} |\mathbf{c}_{1,k}^{M_1} \cdots \mathbf{c}_{n,k}^{M_n}| &\leq \prod_{i=1}^{n} \|F_i\|_{\infty}^{M_i} \leq 2^{nM^{n-1}\delta\Lambda}, \\ |\mathbf{c}_{0,k}^{M_0-k}| &\leq \|F_0\|_{\infty}^{M_0-k} \leq 2^{\sigma(M^n-k)}, \\ \rho_k &\leq \prod_{i=0}^{n} (\#Q_i)^{M_i} \leq ((\delta+1)^n+1)^{M^n} (M+1)^{nM^{n-1}\delta}. \end{aligned}$$

So, $\|R\|_{\infty} \leq 2^{nM^{n-1}\delta(\Lambda+\log M+1))+M^n(\sigma+n\log\delta+n+1)}$. Since $R(Y)$ is a polynomial with integer coefficients, from the Cauchy bound [1, Lem. 10.3], we have that the absolute value of any of its roots is $\geq |\mathrm{tc}(R)|\|R\|_{\infty}^{-1}$, where $\mathrm{tc}(R)$ is the tailing coefficient of R. □

Lemma 4.3. *For all $\boldsymbol{x} \in V_{\mathbb{C}}(\mathcal{I})$, we compute the degree of $F_{\boldsymbol{x}}(Y)$ in bit-complexity in*

$$\widetilde{O}_B\left(\max(M,d)^{n-1}\left\lceil\frac{M}{d}\right\rceil(nd^2(M^n(\tau+n)+n\tau)+n^2d^3\Lambda(M^{n-1}+n))\right).$$

Proof. To determine which is the first non-zero coefficient of $F_{\boldsymbol{x}}(Y) = \sum_{i=0}^{d} f_i(\boldsymbol{x})Y^i$, it suffices to approximate its coefficients up to L bits, where $L \in \widetilde{O}(M^n(\tau+n) + nM^{n-1}d\Lambda)$ (Lem. 4.2). From Prop. 2.4, this can be done, for all $f_i(X)$ and all $\boldsymbol{x} \in V_{\mathbb{C}}(\mathcal{I})$, using multipoint evaluation, in

$$\widetilde{O}_B\left(\max(M,d)^{n-1}\left\lceil\frac{M}{d}\right\rceil(nd^2L+n^2d^2\tau+n^3\,d^3\Lambda)\right) \tag{4.2}$$

bit-operations. This requires approximations of every $\boldsymbol{x}$ to bit-accuracy at most $\widetilde{O}\left(L+nd+n^2d\Lambda\right)$, which is done for all $\boldsymbol{x} \in \mathcal{V}_{\mathbb{C}}(\mathcal{I})$ in $\widetilde{O}_B(M^3+M^2\Lambda+ML+nMd+n^2dM\Lambda)$ [19, Thm.5]. The total bit-complexity is dominated by the one in Eq. (4.2). We substitute the upper bound for L to conclude. □

Lemma 4.4. *For all $\boldsymbol{x} \in V_{\mathbb{C}}(\mathcal{I})$, we compute the number of distinct complex roots of $F_{\boldsymbol{x}}(Y)$ in bit-complexity in*

$$\widetilde{O}_B\left(\max(M,d^2)^{n-1}\left\lceil\frac{M}{d^2}\right\rceil(nd^5(\tau+n)(M^n+n)+n^2d^6\Lambda(M^{n-1}+n))\right).$$

Proof. We define the polynomials $F_\ell(\boldsymbol{X}, Y) := \sum_{i=0}^{\ell} f_i(\boldsymbol{X})Y^i$, for $\ell = 0, \ldots, d$, which are truncated versions of $F(\boldsymbol{X}, Y)$. Since the degree of $F_{\boldsymbol{x}}$ is not the same for every $\boldsymbol{x} \in V_{\mathbb{C}}(\mathcal{I})$, we have to repeat the following steps for every $\ell = 0, \ldots, d$:

(1) We compute the principal subresultant coefficients of $F_\ell(\boldsymbol{X}, Y)$, $\frac{\partial F_\ell}{\partial Y}(\boldsymbol{X}, Y)$ with respect to Y. The j-th principal subresultant coefficient is a polynomial in $\mathbb{Z}[\boldsymbol{X}]$, denoted by $\mathrm{sres}_j(\boldsymbol{X})$, of total degree $O(\ell(\ell-j))$ and bitsize $\widetilde{O}((\tau+n)(\ell-j))$ [1, Prop.8.72]. The computation of all principal subresultant coefficients is done in $\widetilde{O}_B\left(\ell^{2n+2}\tau\right)$ [17, Lem.4].

(2) The index of the first non-zero $\mathrm{sres}_j(\boldsymbol{x})$ gives the degree of the $\gcd(F_\ell(\boldsymbol{x}, Y), \frac{\partial F_\ell}{\partial Y}(\boldsymbol{x}, Y))$. So, for every $\boldsymbol{x} \in V_{\mathbb{C}}(\mathcal{I})$, we approximate $\mathrm{sres}_j(\boldsymbol{x})$, for $j = 0, \ldots, \ell$, up to L bits, with $L \in \widetilde{O}(M^n\ell(\tau+n) + nM^{n-1}\ell^2\Lambda)$ (Lem. 4.2), so as to determine if it zero or not. From Prop. 2.4, this can be done, for all $j \in \{0, \ldots, \ell\}$ and all $\boldsymbol{x} \in V_{\mathbb{C}}(\mathcal{I})$, using multi-point evaluation, in

$$\widetilde{O}_B\left(\max(M,\ell^2)^{n-1}\left\lceil\frac{M}{\ell^2}\right\rceil(n\ell^4(\tau+n)(M^n+n)+n^2\ell^5\Lambda(M^{n-1}+n))\right).$$

Repeating the previous steps for all $\ell \in \{0, \ldots, d\}$, yields a total bit-complexity in

$$\widetilde{O}_B\left(\max(M,d^2)^{n-1}\left\lceil\frac{M}{d^2}\right\rceil(nd^5(\tau+n)(M^n+n)+n^2d^6\Lambda(M^{n-1}+n))\right). \tag{4.3}$$

As we can see in the proof of Prop. 2.4, to compute these evaluations, we need to approximate each $\boldsymbol{x}$ to bit accuracy

$$\widetilde{O}\left(M^nd(\tau+n)+n^2d^2M\Lambda+nd(\tau+n)\right).$$

This costs for all $\boldsymbol{x} \in \mathcal{V}_{\mathbb{C}}(\mathcal{I})$, $\widetilde{O}_B(n\,M\,(M^nd(\tau+n)+n^2d^2M\Lambda+nd(\tau+n)))$ [19, Thm.5], which does not overcome the bit-complexity of Eq. (4.3). □

Lemma 4.5. *For every $\boldsymbol{x} \in V_{\mathbb{C}}(\mathcal{I})$ let $\rho_{\boldsymbol{x}}$ be a positive integer. We compute $\rho_{\boldsymbol{x}}$-approximations of $F_{\boldsymbol{x}}(Y)$ for all $\boldsymbol{x} \in V_{\mathbb{C}}(\mathcal{I})$ in*

$$\widetilde{O}_B\left(n\left(M^3+M^2\Lambda+M\max_{\boldsymbol{x}\in V_{\mathbb{C}}(\mathcal{I})}\rho_{\boldsymbol{x}}\right)+d^{n+1}\left(M^n(\tau+d\Lambda)+\sum_{\boldsymbol{x}\in V_{\mathbb{C}}(\mathcal{I})}\rho_{\boldsymbol{x}}\right)\right).$$

Proof. For a $\rho_{\boldsymbol{x}}$-approximation of $F_{\boldsymbol{x}}(Y)$, it suffices to consider an $L_{\boldsymbol{x}}$-approximation of $\boldsymbol{x}$, where $L_{\boldsymbol{x}} \in \widetilde{O}(\rho_{\boldsymbol{x}}+\tau+d\Lambda)$. This follows from [2, Lem.1], since the coefficients of $F(\boldsymbol{x}, Y)$ are polynomials in $\mathbb{Z}[X_1, \ldots, X_n]$ of size (d, τ), and $\max(1, \|\boldsymbol{x}\|_{\infty}) \in 2^{O(\Lambda)}$. To get the desired approximation of each $\boldsymbol{x} \in V_{\mathbb{C}}(\mathcal{I})$, we compute isolating discs of the roots of each F_i, for $i = 1, \ldots, n$, of size less than $2^{-L_{\max}}$, where $L_{\max} = \max_{\boldsymbol{x}\in V_{\mathbb{C}}(\mathcal{I})} L_{\boldsymbol{x}}$. This costs [19, Thm.5]

$$\widetilde{O}_B\left(n(M^3+M^2\Lambda+ML_{\max})\right). \tag{4.4}$$

For a given $\boldsymbol{x} \in V_{\mathbb{C}}(\mathcal{I})$, to compute the $\rho_{\boldsymbol{x}}$-approximation of $F(\boldsymbol{x}, Y)$, we evaluate its coefficients at the $L_{\boldsymbol{x}}$-approximation of $\boldsymbol{x}$. This costs $\widetilde{O}_B(nd^{n+1}(\tau + L_{\boldsymbol{x}}))$: When we regard $F(\boldsymbol{X}, Y)$ as a polynomial in Y, it has at most $d+1$ coefficients which are polynomials in $\mathbb{Z}[\boldsymbol{X}]$ of size (d, τ). Using [4, Lem.6], we evaluate each one of them in $\widetilde{O}_B(d^n(\tau + L_{\boldsymbol{x}}))$, and then we multiply the latter bound by d. To obtain the final cost, we sum the latter bound for all $\boldsymbol{x} \in V_{\mathbb{C}}(\mathcal{I})$ and add the cost in Eq. (4.4) and use the fact that $L_{\max} \in \widetilde{O}(\max_{\boldsymbol{x} \in V_{\mathbb{C}}(\mathcal{I})} \rho_{\boldsymbol{x}} + \tau + d\Lambda)$. □

Now, by putting everything together, we can prove our main theorem.

Proof of Thm.4.1. (i) For any $\boldsymbol{x} = (x_1, \ldots, x_n) \in V_{\mathbb{C}}(\mathcal{I})$, we compute a $\tau_{\boldsymbol{x}}$ such that $2^{-\tau_{\boldsymbol{x}}-2} \leq \mathrm{lc}(F_{\boldsymbol{x}}) \leq 2^{-\tau_{\boldsymbol{x}}}$. Then, the polynomial $\tilde{F}_{\boldsymbol{x}}(Y) := 2^{-\tau_{\boldsymbol{x}}} F_{\boldsymbol{x}}(Y)$ satisfies the conditions of Prop. 2.2, which we then use to isolate its roots (it has the same roots as $F_{\boldsymbol{x}}(Y)$). From Prop. 2.2, we can solve every $\tilde{F}_{\boldsymbol{x}}(Y)$ in

$$\widetilde{O}_B(d(d^2 + d\log \mathcal{M}(\tilde{F}_{\boldsymbol{x}}) + \mathrm{lGDisc}(\tilde{F}_{\boldsymbol{x}})). \tag{4.5}$$

For that, we require an approximation of the coefficients of $\tilde{F}_{\boldsymbol{x}}(Y)$ to an absolute precision bounded by

$$\rho_{\boldsymbol{x}} \in \widetilde{O}(d\log \mathcal{M}(\tilde{F}_{\boldsymbol{x}}) + \mathrm{lsep}(\tilde{F}_{\boldsymbol{x}}) + \mathrm{lGDisc}(\tilde{F}_{\boldsymbol{x}})).$$

Repeating the arguments as in the proof of [10, Prop. 3.13] we have that $\sum_{\boldsymbol{x} \in V_{\mathbb{C}}(\mathcal{I})} \tau_{\boldsymbol{x}} \in O(\sum_{\boldsymbol{x} \in V_{\mathbb{C}}(\mathcal{I})} \rho_{\boldsymbol{x}})$ and that the computation of $\tau_{\boldsymbol{x}}$ does not affect the total complexity. Then, using Lem. 4.5, we compute $\rho_{\boldsymbol{x}}$-approximations of the coefficients of $\tilde{F}_{\boldsymbol{x}}$ for all $\boldsymbol{x} \in V_{\mathbb{C}}(\mathcal{I})$ in

$$\widetilde{O}_B\big(n(M^3 + M\Lambda(M+d) + M \max_{\boldsymbol{x} \in V_{\mathbb{C}}(\mathcal{I})} \rho_{\boldsymbol{x}}) + \\ + nd^{n+1}M^n(\tau + d\Lambda) + nd^{n+1} \sum_{\boldsymbol{x} \in V_{\mathbb{C}}(\mathcal{I})} \rho_{\boldsymbol{x}}\big). \tag{4.6}$$

From Cor. 3.3 and Cor. 3.5, we have that

$$\sum_{\boldsymbol{x} \in V_{\mathbb{C}}(\mathcal{I})} \mu_{\mathcal{I}}(\boldsymbol{x}) \rho_{\boldsymbol{x}} \in \widetilde{O}(M^n d(n + \tau + d) + nd^2 M^{n-1}\Lambda)).$$

Therefore, Eq. (4.6) becomes

$$\widetilde{O}_B\big(n(M^3 + M^2\Lambda) + nM^{n+1}d(n+\tau+d) + \\ + nM^n d^2(n\Lambda + d^n(n+\tau+d+\Lambda)) + n^2 d^{n+3} M^{n-1}\Lambda\big). \tag{4.7}$$

By summing Eq. (4.5) for all $\boldsymbol{x} \in V_{\mathbb{C}}(\mathcal{I})$ yields

$$\widetilde{O}_B\big(M^n d^2(n+\tau+d) + nd^3 M^{n-1}\Lambda\big). \tag{4.8}$$

We add the bounds in Eq. (4.7) and Eq. (4.8) to conclude (the bound in Eq. (4.7) dominates).

(ii) When the number of distict roots of $F_{\boldsymbol{x}}(Y)$ for every $\boldsymbol{x} \in V_{\mathbb{C}}(\mathcal{I})$ is not already known, then we have to compute it using Lem. 4.4. We add the cost of this computation to the bound of part (i) of the theorem. □

5 APPLICATION: SUM OF SQUARE ROOTS OF INTEGERS PROBLEM

We consider the problem of determining the minimum non-zero difference between two sums of square roots of integers. It appears as Problem 33 in 'The Open Problems Project' and was originally addressed by Joseph O'Rouke [9].

Let $a_i, b_i \in \mathbb{Z}^{\geq 0}$ for $i = 1, \ldots, n$ of bitsize τ. We want to decide if $\sum_{i=1}^n \sqrt{a_i}$ is less than, equal to, or greater than $\sum_{i=1}^n \sqrt{b_i}$. This problem is also related to the Euclidean Travel Salesman Problem (TSP): Given a set of points in the plane with integer coordinates and $L \in \mathbb{N}$, decide if there exists a circuit passing through all these points and having total length (with respect to the Euclidean distance) at most L. The length of the path is a sum of square-roots of integers.

Comparing $\sum_{i=1}^n \sqrt{a_i}$ with $\sum_{i=1}^n \sqrt{b_i}$ in the real-RAM model, can be done trivially. However, in the bit-complexity setting, one has to determine the number of bits that is sufficient to obtain a correct result. We by $r(n, \tau)$ denote the minimum positive value of $\left|\sum_{i=1}^n \sqrt{a_i} - \sum_{i=1}^n \sqrt{b_i}\right|$. Lower bounds on $r(n, \tau)$, and in turn upper bounds on $-\log r(n, \tau)$, give upper bounds on the precision needed to compare $\sum_{i=1}^n \sqrt{a_i}$ with $\sum_{i=1}^n \sqrt{b_i}$. In particular, if $-\log r(n, \tau)$ is bounded above by a polynomial in k and n, then the sign of $\sum_{i=1}^n \sqrt{a_i} - \sum_{i=1}^n \sqrt{b_i}$ can be computed in polynomial time. Nevertheless, existing upper bounds on $-\log r(n, \tau)$ are exponential. In [6, 20] they prove that $-\log r(n, \tau) \in \widetilde{O}(\tau 2^{2n})$, through studying separation bounds.

Here, we apply the results of Sec. 3 to derive bounds that, however, remain exponential in n. Nonetheless, the same bounds apply to the sum of all the roots of the associated system that has as root the two quantities that we have to compare. We consider the system

$$\left\{\begin{array}{c} F_i(X_i) := X_i^2 - a_i = 0\,, i \in [n] \\ G_i(Y_i) := Y_i^2 - b_i = 0\,, i \in [n] \\ H(\boldsymbol{X}, \boldsymbol{Y}, Z) := (Z - X_1 - \cdots - X_n)(Z - Y_1 - \cdots - Y_n) = 0 \end{array}\right\}$$

Let $\mathcal{K} = \langle F_1, \ldots, F_n, G_1, \ldots, G_n \rangle$. From Cor.3.5 we have that

$$\sum_{(\boldsymbol{x}, \boldsymbol{y}) \in V_{\mathbb{C}}(\mathcal{K})} \mu_{\mathcal{K}}(\boldsymbol{x}, \boldsymbol{y}) \mathrm{lsep}(H(\boldsymbol{x}, \boldsymbol{y}, \cdot)) \in \widetilde{O}(n 2^{2n} \tau)\,,$$

or equivalently, since all the multiplicities are equal to one,

$$\sum_{(\boldsymbol{x}, \boldsymbol{y}) \in V_{\mathbb{C}}(\mathcal{K})} \Big|\log\Big|\sum_{i=1}^n x_i - \sum_{i=1}^n y_i\Big|\Big| \in \widetilde{O}(n 2^{2n} \tau)\,. \tag{5.1}$$

We see that, as Eq. (5.1) shows, not only $-\log r(n, \tau)$ is in $\widetilde{O}(n\tau 2^{2n})$, but also the sum of the differences associated to all the 2^{2n} roots of the system $\{F_1 = \cdots = F_n = G_1 = \cdots = G_n = 0\}$.

6 CONCLUDING REMARKS

We provided amortized bounds on the separation of a polynomial with coefficients in a multiple field extension. We used these bounds to estimate the bit-complexity of isolating its roots and applied them to the 'sum of square roots of integers' problem. For the root isolation, we followed an adaptive approach that we juxtaposed to a theoretical one that uses maximal precision, but leads to better bit-complexity estimates. In a future work, we set our sights on developing an adaptive method that matches the bit-complexity of the theoretical one.

Acknowledgements. The authors would like to thank Elias Tsigaridas for the several discussions on the subject and Michael Sagraloff for his feedback and the suggestion of the theoretical approach.

REFERENCES

[1] S. Basu, R. Pollack, and M-F. Roy. 2006. *Algorithms in Real Algebraic Geometry.* Algorithms and Computation in Mathematics, Vol. 10. Springer-Verlag.

[2] Ruben Becker and Michael Sagraloff. 2017. Counting Solutions of a Polynomial System Locally and Exactly. *ArXiv* abs/1712.05487 (2017).

[3] François Boulier, Changbo Chen, François Lemaire, and Marc Maza. 2010. Real Root Isolation of Regular Chains. *Proc. Asian Symposium on Computer Mathematics (ASCM)* (09 2010). https://doi.org/10.1007/978-3-662-43799-5_4

[4] Yacine Bouzidi, Sylvain Lazard, Marc Pouget, and Fabrice Rouillier. 2014. Separating linear forms and Rational Univariate Representations of bivariate systems. *Journal of Symbolic Computation* 68 (08 2014). https://doi.org/10.1016/j.jsc.2014.08.009

[5] Cornelius Brand and Michael Sagraloff. 2016. On the Complexity of Solving Zero-Dimensional Polynomial Systems via Projection. In *Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation - ISSAC '16.* ACM Press, Waterloo, ON, Canada, 151–158. https://doi.org/10.1145/2930889.2930934

[6] Christoph Burnikel, Rudolf Fleischer, Kurt Mehlhorn, and Stefan Schirra. 2000. A Strong and Easily Computable Separation Bound for Arithmetic Expressions Involving Radicals. *Algorithmica* 27 (2000), 87–99.

[7] Changbo Chen and Marc Moreno Maza. 2012. Algorithms for computing triangular decomposition of polynomial systems. *Journal of Symbolic Computation* 47, 6 (2012), 610–642. https://doi.org/10.1016/j.jsc.2011.12.023 Advances in Mathematics Mechanization.

[8] X. Dahan, É. Schost, M. Moreno Maza, W. Wu, and Y. Xie. 2005. On the Complexity of the D5 Principle. *SIGSAM Bull.* 39, 3 (sep 2005), 97–98. https://doi.org/10.1145/1113439.1113457

[9] Erik D. Demaine, Joseph B. M. Mitchell, and Joseph O'Rourke. 2007. Problem 33. *The Open Problems Project.* https://topp.openproblem.net/p33

[10] Daouda Niang Diatta, Sény Diatta, Fabrice Rouillier, Marie-Françoise Roy, and Michael Sagraloff. 2022. Bounds for Polynomials on Algebraic Numbers and Application to Curve Topology. *Discrete & Computational Geometry* (15 Feb 2022). https://doi.org/10.1007/s00454-021-00353-w

[11] Ioannis Emiris, Bernard Mourrain, and Elias Tsigaridas. 2020. Separation bounds for polynomial systems. *Journal of Symbolic Computation* 101 (2020), 128–151.

[12] Rémi Imbach, Marc Pouget, and Chee-Keng Yap. 2021. Clustering Complex Zeros of Triangular Systems of Polynomials. *Mathematics in Computer Science* 15 (2021), 271–292.

[13] J. R. Johnson and Werner Krandick. 1997. Polynomial Real Root Isolation Using Approximate Arithmetic. In *Proceedings of the 1997 International Symposium on Symbolic and Algebraic Computation* (Kihei, Maui, Hawaii, USA) *(ISSAC '97).* Association for Computing Machinery, New York, NY, USA, 225–232. https://doi.org/10.1145/258726.258790

[14] Michael Kerber and Michael Sagraloff. 2012. A worst-case bound for topology computation of algebraic curves. *J. Symb. Comput.* 47, 3 (2012), 239–258.

[15] Alexander Kobel and Michael Sagraloff. 2013. Fast Approximate Polynomial Multipoint Evaluation and Applications. *CoRR* abs/1304.8069 (2013). arXiv:1304.8069 http://arxiv.org/abs/1304.8069

[16] Alexander Kobel and Michael Sagraloff. 2015. On the complexity of computing with planar algebraic curves. *Journal of Complexity* 31, 2 (2015), 206–236. https://doi.org/10.1016/j.jco.2014.08.002

[17] Sylvain Lazard, Marc Pouget, and Fabrice Rouillier. 2017. Bivariate triangular decompositions in the presence of asymptotes. *Journal of Symbolic Computation* 82 (2017), 123–133. https://doi.org/10.1016/j.jsc.2017.01.004

[18] Angelos Mantzaflaris, Eric Schost, and Elias Tsigaridas. 2017. Sparse Rational Univariate Representation. In *Proceedings of the 2017 ACM on International Symposium on Symbolic and Algebraic Computation* (Kaiserslautern, Germany) *(ISSAC '17).* Association for Computing Machinery, New York, NY, USA, 301–308. https://doi.org/10.1145/3087604.3087653

[19] Kurt Mehlhorn, Michael Sagraloff, and Pengming Wang. 2015. From approximate factorization to root isolation with application to cylindrical algebraic decomposition. *Journal of Symbolic Computation* 66 (2015), 34–69. https://doi.org/10.1016/j.jsc.2014.02.001

[20] K. Mehlhorn and S. Schirra. 2000. *A Generalized and Improved Constructive Separation Bound for Real Algebraic Expressions.* MPI Informatik, Bibliothek & Dokumentation. https://books.google.fr/books?id=S8c1vwEACAAJ

[21] Victor Y. Pan. 2002. Univariate Polynomials: Nearly Optimal Algorithms for Numerical Factorization and Root-finding. *Journal of Symbolic Computation* 33, 5 (2002), 701–733. https://doi.org/10.1006/jsco.2002.0531

[22] Siegfried M. Rump. 1977. Real Root Isolation for Algebraic Polynomials. *SIGSAM Bull.* 11, 2 (may 1977), 2–3. https://doi.org/10.1145/1088240.1088241

[23] Adam Strzebonski and Elias Tsigaridas. 2019. Univariate real root isolation in an extension field and applications. *Journal of Symbolic Computation* 92 (2019), 31–51. https://doi.org/10.1016/j.jsc.2017.12.001

[24] Adam Strzeboński and Elias P. Tsigaridas. 2012. Univariate Real Root Isolation in Multiple Extension Fields. In *Proceedings of the 37th International Symposium on Symbolic and Algebraic Computation* (Grenoble, France) *(ISSAC '12).* Association for Computing Machinery, New York, NY, USA, 343–350. https://doi.org/10.1145/2442829.2442878

[25] Joris van der Hoeven and Grégoire Lecerf. 2020. Fast multivariate multi-point evaluation revisited. *Journal of Complexity* 56 (2020), 101405. https://doi.org/10.1016/j.jco.2019.04.001

[26] Bican Xia and Lu Yang. 2002. An Algorithm for Isolating the Real Solutions of Semi-algebraic Systems. *J. Symb. Comput.* 34 (2002), 461–477.

[27] Zhihai Zhang, Tian Fang, and Bican Xia. 2009. Real Solution Isolation with Multiplicity of Zero-Dimensional Triangular Systems. *Science China Information Sciences* 54 (06 2009). https://doi.org/10.1007/s11432-010-4154-y