On the Complexity of Solving a Bivariate Polynomial System

Pavel Emeliyanenko
Max-Planck-Institut für Informatik, Germany
asm@mpi-inf.mpg.de

Michael Sagraloff
Max-Planck-Institut für Informatik, Germany
msagralo@mpi-inf.mpg.de

ABSTRACT

We study the complexity of computing the real solutions of a bivariate polynomial system using the recently presented algorithm BISOLVE [2]. BISOLVE is an elimination method which first projects the solutions of a system onto the x- and y-axes and, then, selects the actual solutions from the so induced candidate set. However, unlike similar algorithms, BISOLVE requires no genericity assumption on the input nor it needs any change of the coordinate system. Furthermore, extensive benchmarks from [2] confirm that the algorithm outperforms state of the art approaches by a large factor. In this paper, we show that, for two polynomials $f, g \in \mathbb{Z}[x, y]$ of total degree at most n with integer coefficients bounded by 2^τ, BISOLVE computes isolating boxes for all real solutions of the system $f = g = 0$ using $\tilde{\mathcal{O}}(n^8 + n^7\tau)$ bit operations[1], thereby improving the previous record bound by four magnitudes.

1. INTRODUCTION

Systems of polynomial equations naturally arise in many fields of science and engineering. In computational geometry and computer graphics, there is a particular interest in the study of polynomial systems in two or three variables: Almost all existing exact and complete algorithms for computing the topology or an arrangement of algebraic curves [4, 10] (and surfaces [3]) are crucially based on determining so-called critical points (extremal points, singularities, etc.), which are in turn the solutions of a bivariate polynomial system. In this work, we investigate in the bit complexity analysis of the recently presented algorithm BISOLVE [2] to isolate the real solutions of a polynomial system

$$f(x,y) = \sum_{i+j\le n} f_{ij}x^i y^j = 0, \quad g(x,y) = \sum_{i+j\le n} g_{ij}x^i y^j = 0, \tag{1.1}$$

where $f,\ g \in \mathbb{Z}[x, y]$ are polynomials of *magnitude* (n, τ), that is, their total degrees are bounded by n, and their coefficients are integers of modulus 2^τ or less. Henceforth, we assume that f and g share no common non-trivial factor in $\mathbb{Z}[x, y]\backslash\mathbb{Z}$ which, due to Bézout's Theorem, is equivalent to the existence of finitely many complex solutions of (1.1). BISOLVE computes a set of disjoint boxes $B_k \subset \mathbb{R}^2$, $k = 1, \ldots, m$, such that the union of all B_k contains

$$V_{\mathbb{R}} := \{(x, y) \in \mathbb{R}^2 | f(x, y) = g(x, y) = 0\},$$

the set of all *real* solutions of (1.1), and each B_k is *isolating* (i.e. each B_k contains exactly one solution). We show that BISOLVE demands for $\tilde{\mathcal{O}}(n^8+n^7\tau)$ bit operations, thus improving the previous record bound $\tilde{\mathcal{O}}(n^{12} + n^{10}\tau^2)$ from [6] by four magnitudes. In comparison to [6], our analysis uses two recently presented asymptotically fast algorithms for isolating [18] and refining [11] the real roots of a univariate polynomial. In comparison to the much more involved algorithms from A. Schönhage [20] and V. Pan [15], which achieve comparable complexity bounds, the recently presented algorithms are very practical, a crucial property for our principle object to provide methods which are efficient in practice as well as in theory.

We would also like to stress the fact that the obtained complexity result for BISOLVE is not only due to the use of asymptotically fast methods to isolate and refine the roots of a univariate polynomial, but rather due to the effectiveness of the novel inclusion predicate which is used in the *validation step* (or lifting step) of BISOLVE in order to certify or to discard candidate solutions. In fact, we consider the achieved improvement in the projection step to be incremental[2] whereas the non-trivial analysis of the lifting step, and thus of the novel inclusion predicate, constitutes the main contribution of this paper. As a byproduct, which may be of some independent interest, our analysis yields an upper bound for $\Sigma(F) := \sum_{z:F(z)=0} \log \mathrm{sep}(z, F)^{-1}$ which generalizes the well-known bounds for arbitrary *square-free* polynomials $F \in \mathbb{Z}[x]$ to the general case.[3]

The lifting step in BISOLVE which is based on a homotopy argument is completely different from previous approaches which rely on the evaluation of signed remainder sequences (SRs) to identify the common roots of $f(\alpha, y)$ and $g(\alpha, y)$,

[1] $\tilde{\mathcal{O}}$ indicates that polylogarithmic factors in τ and n are omitted.

[2] The results from [18, 11] can certainly also be applied to previous elimination methods in order to improve the complexity results for the *projection step*.

[3] The sum is taken over all complex roots z of F (counted with multiplicity), where $\mathrm{sep}(z, F)$ denotes the separation (i.e. the minimal distance of z to a root $z' \neq z$ of F) of z.

with α being the projection of a solution onto the x-axis. The cost of computing SRs often becomes dominating in practice. Instead, BISOLVE completely avoids such computations and, as confirmed by the experiments in [2], outperforms other state of the art approaches such as LGP [5] or Maple's ISOLATE by large factors. This shows that the efficiency of our algorithm in theory as proven in the present paper is not just the product of purely theoretical manipulations with the single goal to achieve the best asymptotic complexity disregarding many aspects of practical value.

Related work. An early result on the complexity analysis appears in [9], where the closely related problem of computing the topology of an algebraic curve is considered. The authors analyze the algorithm TOP and derive a complexity bound of $\tilde{\mathcal{O}}(N^{14})$ bit operations, with $N = \max(n, \tau)$. Another work [6] discusses three methods to solve a bivariate polynomial system. All of them are based on the computation of signed remainder sequences. The first method, GRID, projects the solutions onto orthogonal axes and, then, matches them by means of a SIGN_AT procedure which computes the signs of a subresultant sequence. The complexity of GRID is bounded by $\tilde{\mathcal{O}}(N^{14})$ bit operations, where the overall cost is dominated by that of the SIGN_AT operations. It should be noted that, despite the fact that BISOLVE follows the same algorithmic idea as GRID (i.e. to project onto the x- and y-axes and to choose the right solutions from the induced candidate set), the final validation steps of the two methods are completely different. The second approach called M_RUR assumes that the system is in generic position (i.e. no two solutions share a common x-coordinate). It is based on the computation of a rational univariate representation (RUR) and achieves a bit complexity of $\tilde{\mathcal{O}}(n^{10}(n^2+\tau^2)) = \tilde{\mathcal{O}}(N^{12})$. The third approach, G_RUR, achieves the same bit complexity as M_RUR but relies on computing $H_\alpha \in \mathbb{Z}(\alpha)[y]$, the greatest common divisor of the square-free parts of $f(\alpha, y)$ and $g(\alpha, y)$, where α is a projected solution of the system. It seems that using asymptotically fast algorithms for the tasks of isolating and refining the roots of a polynomial also leads to a considerable improvement of the overall complexity of the algorithms M_RUR (only in a sheared system) and G_RUR. However, since only the bound for the projection step improves, the so obtained bounds for the overall bit complexity are considerably weaker (at least two magnitudes) than the bound achieved by BISOLVE. For instance, the computational complexity of the final steps (e.g. the sign evaluations of H_α at candidate intervals) of M_RUR and G_RUR is at least by a factor n^2 larger. In addition, the analysis of the lifting step in G_RUR is based on the study [21] of a modular GCD algorithm over an extension field. Besides the fact that we do not consider the computation of the polynomials H_α to be very practical, we remark that [21] only provides a bound on the *expected* number of bit operations; see Section 3.2 in [21]. In his dissertation, M. Kerber describes randomized algorithms to analyze the topology of a single algebraic curve and to compute arrangements of such curves. The algorithm also uses SRs and applies a coordinate transformation to ensure generic position. A detailed analysis of the "curve-pair analysis" which solves the subproblem of finding the solutions of a bivariate system shows that the corresponding complexity is bounded by an *expected number* of $\tilde{\mathcal{O}}(n^{10}(n+\tau)^2)$ bit operations; see [10, Section 3.3.4]. Recent work [12] improves the latter analysis for the task of computing the topology of a single algebraic curve. As a result, the topology of a single curve can be deterministically computed using $\tilde{\mathcal{O}}(n^8\tau(n+\tau))$ bit operations.

Outline. Section 2 introduces some notations which are used throughout the argument. In Section 3, we briefly review the algorithm BISOLVE. Here, we omit some technical details and filtering techniques to keep the presentation simple. The complexity analysis is given in Section 4. We analyze the three main steps of the algorithm separately, and then combine the results yielding the overall complexity. Finally, in Section 5, we give some concluding remarks.

2. SETTING

We express the input polynomials f and g in (1.1) as *univariate* polynomials in x and y of degrees n_x and n_y, respectively:

$$f(x,y) = \sum_{i=0}^{n_x} f_i^{(x)}(y)x^i = \sum_{i=0}^{n_y} f_i^{(y)}(x)y^i,$$
$$g(x,y) = \sum_{i=0}^{n_x} g_i^{(x)}(y)x^i = \sum_{i=0}^{n_y} g_i^{(y)}(x)y^i,$$

where $f_i^{(y)}$, $g_i^{(y)} \in \mathbb{Z}[x]$, and $f_i^{(x)}$, $g_i^{(x)} \in \mathbb{Z}[y]$. Throughout the paper, it is assumed that n_x, $n_y \le n$. We denote the *Sylvester matrix* associated with the polynomials f and g by $S^{(y)} = S^{(y)}(f,g)$. Its entries are the coefficients $\{f_i^{(y)}\}$ and $\{g_i^{(y)}\}$; see [8, p. 286] for the definition. The resultant $R^{(y)}(x) = \operatorname{res}(f,g;y) \in \mathbb{Z}[x]$ of f and g with respect to y is the determinant of $S^{(y)}$. By analogy, $R^{(x)}(y) = \operatorname{res}(f,g;x) \in \mathbb{Z}[y]$ defines the resultant with respect to x and $S^{(x)}(f,g)$ the associated Sylvester's matrix with entries $\{f_i^{(x)}\}$ and $\{g_i^{(x)}\}$. If this causes no ambiguity, we also write R omitting the variable index and by R^* the square-free part of R.

For a (not necessarily square-free) polynomial $F(x) = \sum_{i=0}^{n} F_i x^i \in \mathbb{R}[x]$ of degree $n := \deg(F)$, $\operatorname{lcf}(F) := F_n$ denotes the *leading coefficient of* F. Let $z_1 \dots z_m \in \mathbb{C}$ be the distinct roots of F, then $\operatorname{mult}(z_i, F)$ denotes the multiplicity of the root z_i and $\operatorname{sep}_i := \operatorname{sep}(z_i, F)$ the separation of z_i (i.e. the minimal distance of z_i to any $z_j \neq z_i$). The separation $\operatorname{sep}(F)$ of F is the minimum of all sep_i, $\Sigma^*(F) := \sum_{i=1}^{m} \log \operatorname{sep}_i^{-1}$, and $\Sigma(F) := \sum_{i=1}^{n} \log \operatorname{sep}_i^{-1} = \sum_{i=1}^{m} \operatorname{mult}(z_i, F) \cdot \log \operatorname{sep}_i^{-1}$. Finally, we denote $\Gamma(F) := \max_i |z_i|$ the maximal absolute value of all z_i, and $\mathcal{M}(F) := |\operatorname{lcf}(F)| \prod_{i=0}^{k} \max\{1, |z_i|\}$ the *Mahler measure* of F.

For an interval $I = (a,b) \subset \mathbb{R}$, $w_I := b - a$ denotes the *width*, $m_I := (a+b)/2$ the *center* and $r_I := (b-a)/2$ the *radius* of I. A disc in $\mathbb{C}$ is denoted by $\Delta := \Delta_r(m)$, where $m \in \mathbb{C}$ defines the center of Δ and $r \in \mathbb{R}^+$ its radius.

3. REVIEW OF THE ALGORITHM

In this section, we briefly review the algorithm BISOLVE to make the paper self-contained; for further details and filtering techniques used in the actual realization, we refer the interested reader to [2]. At the highest level, BISOLVE comprises three subroutines which we consider separately:

PROJECT: We first project the complex solutions of (1.1)

onto the x- and y-axes. That is, we consider the two sets:

$$V_{\mathbb{C}}^{(x)} := \{x \in \mathbb{C} | \exists y \in \mathbb{C} \wedge f(x,y) = g(x,y) = 0\},$$
$$V_{\mathbb{C}}^{(y)} := \{y \in \mathbb{C} | \exists x \in \mathbb{C} \wedge f(x,y) = g(x,y) = 0\}$$

and compute their restrictions $V_{\mathbb{R}}^{(x)} := V_{\mathbb{C}}^{(x)} \cap \mathbb{R}$ and $V_{\mathbb{R}}^{(y)} := V_{\mathbb{C}}^{(y)} \cap \mathbb{R}$ to the real values. The real solutions $V_{\mathbb{R}}$ of (1.1) are then contained in the product

$$\mathcal{C} := V_{\mathbb{R}}^{(x)} \times V_{\mathbb{R}}^{(y)} \subset \mathbb{R}^2, \tag{3.1}$$

which we denote the set of *candidate solutions* for (1.1). For computing $V_{\mathbb{R}}^{(x)}$ and $V_{\mathbb{R}}^{(y)}$, we first compute the resultants $R^{(y)}$ and $R^{(x)}$, respectively, and extract the square-free part R^* of either polynomial ($R = R^{(y)}$ or $R = R^{(x)}$ for short). Then, we isolate the real roots α_i of R^* using the algorithm NEWDSC from [18]. NEWDSC is a subdivision approach based on the combination of Descartes' Rule of Signs and Newton iteration. With respect to bit complexity, it achieves the best bound that is so far known for this problem; see also [15] for an overview of asymptotically fast numerical algorithms to isolate all complex roots. Yet, in contrast to the latter mentioned asymptotically fast methods, NEWDSC concentrates on the real roots only and is much easier to access and to implement.

SEPARATE: In this step, the real roots of R are further separated from the complex ones. That is, for each real root α, we refine a corresponding isolating interval $I := I(\alpha)$ until the disc $\Delta_{8r_I}(m_I)$ contains no root of R except α. In order to guarantee the latter property, we refine I until the following inequality holds (see [2, Thm. 2] for a proof):

$$|(R^*)'(m_I)| - \frac{3}{2} \sum\nolimits_{k \geq 2} \left| \frac{(R^*)^{(k)}(m_I)}{k!} \right| (8r_I)^k > 0. \tag{3.2}$$

In the next step, we compute

$$LB(\alpha) := 2^{-2 \deg R} |R(m_I - 2r_I)|, \tag{3.3}$$

which constitutes a lower bound for $|R(x)|$ on the boundary $\partial\Delta(\alpha)$ of $\Delta(\alpha) := \Delta_{2r_I}(m_I)$, that is, $|R(x)| > LB(\alpha)$ for all $x \in \partial\Delta(\alpha)$; see [2, Thm. 3.2] for a proof.

Finally, for each real root α of $R^{(y)}$ (and β of $R^{(x)}$), we have isolating intervals $I(\alpha)$ (and $I(\beta)$) and isolating discs $\Delta(\alpha) = \Delta_{2r_{I(\alpha)}}(m_{I(\alpha)})$ (and $\Delta(\beta)$). Hence, each real solution of the system (1.1) is contained in a polydisc $\Delta(\alpha,\beta) := \Delta(\alpha) \times \Delta(\beta) \subset \mathbb{C}^2$, and each of these polydiscs contains at most one solution. In addition, for each point (x,y) on the boundary of a polydisc $\Delta(\alpha,\beta)$, we have $|R^{(y)}(x)| > LB(\alpha)$ or $|R^{(x)}(y)| > LB(\beta)$.

VALIDATE: The goal of this final stage is to determine all candidates $(\alpha,\beta) \in \mathcal{C}$ which are actually solutions of (1.1) and to exclude the remaining ones. Again, in order to facilitate the complexity analysis, we assume that the actual solutions are chosen exclusively based on the *inclusion test* outlined below. We remark that the efficiency of the actual implementation is further due to a series of filtering techniques to rapidly exclude the majority of candidates. This, for instance, includes an interval Descartes algorithm [7] to approximate the roots of $f(\alpha,y)$ and $g(\alpha,y)$.

In SEPARATE, we have already computed lower bounds $LB(\alpha)$ and $LB(\beta)$ for the values of $|R^{(y)}|$ and $|R^{(x)}|$ at the boundaries of $\Delta(\alpha)$ and $\Delta(\beta)$, respectively. We now (conceptually) rewrite $R^{(y)}$ in terms of cofactors $u^{(y)}$ and $v^{(y)}$ (see [8, p. 287] for more details):

$$R^{(y)}(x) = u^{(y)}(x,y) f(x,y) + v^{(y)}(x,y) g(x,y), \tag{3.4}$$

where $u^{(y)}$ and $v^{(y)}$ are determinants of "Sylvester-like" matrices $U^{(y)}$ and $V^{(y)}$. These matrices are obtained from the matrix $S^{(y)}(f,g)$ by replacing the last column with vectors $(y^{n_y-1} \dots\ y\ 1\ 0 \dots\ 0)^T$ and $(0 \dots\ 0\ y^{m_y-1} \dots\ y\ 1)^T$ of size $n_y + m_y$, respectively. Now, *without explicitly computing* the cofactors (which are typically very large expressions), we determine upper bounds $UB(\alpha,\beta,u^{(y)})$ and $UB(\alpha,\beta,v^{(y)})$ for $|u^{(y)}|$ and $|v^{(y)}|$ on $\Delta(\alpha,\beta)$, respectively. This is achieved by bounding the absolute values of the entries in $U^{(y)}$ and $V^{(y)}$ and, then, applying Hadamard's inequality to $U^{(y)}$ and $V^{(y)}$. Cofactor polynomials $u^{(x)}$, $v^{(x)}$ and respective upper bounds $UB(\alpha,\beta,u^{(x)})$, $UB(\alpha,\beta,v^{(x)})$ are defined in an analogous way for the resultant polynomial $R^{(x)}$. The inclusion test based on a homotopy argument is now formulated as follows (see [2, Thm. 4] for a proof):

Theorem 1 *If there exists a* $\xi := (x_0,y_0) \in \Delta(\alpha,\beta)$ *with*

$$UB(\alpha,\beta,u^{(y)}) \cdot |f(\xi)| + UB(\alpha,\beta,v^{(y)}) \cdot |g(\xi)| < LB(\alpha), \tag{3.5}$$

$$UB(\alpha,\beta,u^{(x)}) \cdot |f(\xi)| + UB(\alpha,\beta,v^{(x)}) \cdot |g(\xi)| < LB(\beta), \tag{3.6}$$

then $\Delta(\alpha,\beta)$ *contains a solution of (1.1), and* $f(\alpha,\beta) = 0$.□

The candidate solutions $(\alpha,\beta) \in \mathcal{C}$ are now treated as follows: Let $B(\alpha,\beta) = I(\alpha) \times I(\beta) \subset \mathbb{R}^2$ be the corresponding *candidate box*. Each candidate box is then refined until we can ensure that $f(\alpha,\beta) \neq 0$ or $g(\alpha,\beta) \neq 0$ (using interval arithmetic on $B(\alpha,\beta)$), or, for an arbitrary point $(x_0,y_0) \in B(\alpha,\beta)$, the inequalities (3.5) and (3.6) are fulfilled. In the latter case, Theorem 1 guarantees that (α,β) is a solution of (1.1). We refer to Section 4.3.2 for the details of the evaluation using interval arithmetic.

4. COMPLEXITY ANALYSIS

Throughout the analysis, we assume that the multiplication of two integers is always done in *asymptotically fast* way. In other words, the bit complexity to multiply two k-bit integers is assumed to be $M(k) = \mathcal{O}(k \log k \log\log k) = \tilde{\mathcal{O}}(k)$.

4.1 PROJECT

For computing the resultant $R = R^{(x)}$ (or $R = R^{(y)}$), we use an asymptotically fast subresultant algorithm based on Half-GCD computation from [16]. Thus, both resultant computations need $\tilde{\mathcal{O}}(n^4\tau)$ bit operations, and the resulting polynomials have magnitude

$$(n^2, \mathcal{O}(n(\log n + \tau))).$$

Next, we compute $R/\gcd(R,R')$ to extract the square-free part R^* of R. According to [13, 16], this operation demands for $\tilde{\mathcal{O}}(n^5(\tau + \log n))$ bit operations, and R^* is of magnitude

$$(n^2, \mathcal{O}(n(n+\tau))). \tag{4.1}$$

Finally, the real roots of R^* are isolated using NEWDSC as outlined in Section 3. Given a square-free polynomial

$F \in \mathbb{Z}[x]$ of magnitude (N, μ) and an integer $L \in \mathbb{N}$, we can compute isolating intervals (for all real roots) of width 2^{-L} using no more than $\tilde{O}(N^3\mu + N^2 L)$ bit operations; see [18, Theorem 10]. Hence, the cost for the considered isolation step is bounded by $\tilde{O}(n^8 + n^7\tau)$. We remark that the same complexity bound can be achieved when using an asymptotically fast numerical solver (e.g. [15]) to approximate all complex roots of R^*.

4.2 Separate

Before we start with the actual analysis of Separate, we provide an upper bound for $\Sigma(F) = \sum_z \log \operatorname{sep}(z, F)^{-1}$, where F denotes an arbitrary (not necessarily square-free) polynomial F of magnitude (N, μ), and the sum is taken over all roots of F counted with multiplicity. In the case where F is square-free, we have $\Sigma(F) = \tilde{\mathcal{O}}(N\mu)$; e.g. see [17, Lemma 19] or [19]. However, to the best of our knowledge, there exists no comparable bound in the literature which applies to polynomials F with multiple roots. The following Theorem provides such a bound which may be of independent interest.

Theorem 2 *Let $F \in \mathbb{Z}[x]$ be a polynomial of magnitude (N, μ). We denote $z_1, \ldots, z_d$ the distinct complex roots of F and $s_i := \operatorname{mult}(z_i, F)$ the multiplicity of z_i. Then, for arbitrary non-negative integers m_i, with $m_i \leq s_i$, we have*

$$\sum_{i=1}^{d} m_i \log \operatorname{sep}(z_i, F)^{-1} = \tilde{\mathcal{O}}(N^2 + N\mu).$$

□

Proof. We consider the factorization of F (over $\mathbb{Z}$) into square-free and pair-wise coprime factors:

$$F(x) = \prod_{i=1}^{k} Q_i(x)^{s_i}, \quad d_i := \deg(Q_i) \geq 1,$$

such that the polynomials $Q_i(x)$ and $F(x)/Q_i(x)^{s_i}$ are coprime, and $N = \sum_{i=1}^{k} d_i s_i$. We further denote F^* the square-free part of F and $d := \deg(F^*) = \sum_{i=1}^{k} d_i$ its degree. Then, for arbitrary roots α and β of F^*, it holds that

$$\begin{aligned}
|(F^*)'(\alpha)| &= |\operatorname{lcf}(F^*)| \cdot |\alpha - \beta| \prod_{\gamma \neq \alpha, \beta : F^*(\gamma) = 0} |\gamma - \alpha| \\
&\leq |\operatorname{lcf}(F^*)| \cdot |\alpha - \beta| \prod_{\gamma \neq \alpha, \beta : F^*(\gamma) = 0} 2\max(1, |\alpha|, |\gamma|) \\
&\leq 2^{d-2} |\alpha - \beta| \max(1, |\alpha|)^{d-3} \mathcal{M}(F^*)
\end{aligned}$$

since $\mathcal{M}(F^*) = |\operatorname{lcf}(F^*)| \cdot \prod_{z : F^*(z) = 0} \max(1, |z|)$. Suppose, w.l.o.g., that α is a root of Q_i and β is a root of F^* closest to α. Then, according to the above inequality, we have

$$\operatorname{sep}(\alpha, F) = |\alpha - \beta| \geq \frac{|(F^*)'(\alpha)|}{2^{d-2} \max(1, |\alpha|)^{d-3} \mathcal{M}(F^*)}$$

We now apply this inequality to the product over all $\operatorname{sep}(\alpha_j, F)$, $j = 1, \ldots, d_i$, where $\alpha_1, \ldots, \alpha_{d_i}$ denote the roots of Q_i:

$$\begin{aligned}
\prod_{j=1}^{d_i} \operatorname{sep}(\alpha_j, F) &\geq 2^{(2-d)d_i} \mathcal{M}(Q_i)^{3-d} \mathcal{M}(F^*)^{-d_i} \prod_{j=1}^{d_i} |(F^*)'(\alpha_j)| \\
&= 2^{(2-d)d_i} \mathcal{M}(Q_i)^{3-d} \mathcal{M}(F^*)^{-d_i} \prod_{j=1}^{d_i} |(Q_i)'(\alpha_j) \cdot \frac{F^*}{Q_i}(\alpha_j)|
\end{aligned} \tag{4.2}$$

since

$$(F^*)'(\alpha_j) = \underbrace{Q_i(\alpha_j)}_{=0} \cdot \left(\frac{F^*}{Q_i}\right)'(\alpha_j) + (Q_i)'(\alpha_j) \cdot \frac{F^*}{Q_i}(\alpha_j)$$

In addition, we have

$$\prod_{j=1}^{d_i} |Q_i'(\alpha_j)| = |\operatorname{lcf}(Q_i)^{2-d_i} \operatorname{Disc}(Q_i)| \geq |\operatorname{lcf}(Q_i)^{2-d_i}|, \text{ and}$$

$$\prod_{j=1}^{d_i} |\frac{F^*}{Q_i}(\alpha_j)| = |\operatorname{lcf}(Q_i)^{d_i - d} \operatorname{res}(Q_i, \frac{F^*}{Q_i})| \geq |\operatorname{lcf}(Q_i)^{d_i - d}|$$

since $\operatorname{Disc}(Q_i)$ and $\operatorname{res}(Q_i, \frac{F^*}{Q_i})$ are non-zero integers. Applying the latter two inequalities to (4.2) now yields:

$$\prod_{j=1}^{d_i} \operatorname{sep}(\alpha_j, F) \geq 2^{(2-d)d_i} \mathcal{M}(Q_i)^{3-d} \mathcal{M}(F^*)^{-d_i} |\operatorname{lcf}(Q_i)^{2-d}|$$

Finally, we consider the product of the separations of all roots to the respective powers s_i:

$$\begin{aligned}
\prod_{i=1}^{k} \prod_{j=1}^{d_i} \operatorname{sep}(\alpha_j, F)^{s_i} &\geq \prod_{i=1}^{k} 2^{(2-d)d_i s_i} \mathcal{M}(Q_i)^{(3-d)s_i} \\
&\quad \cdot \mathcal{M}(F^*)^{-d_i s_i} \cdot \prod_{i=1}^{k} |\operatorname{lcf}(Q_i)|^{-s_i} \\
&= 2^{(2-d)N} \mathcal{M}(F)^{3-d} \mathcal{M}(F^*)^{-N} |\operatorname{lcf}(F)|^{-1} = 2^{-\tilde{\mathcal{O}}(N^2 + N\mu)}
\end{aligned}$$

where we used that $\prod_{i=1}^{k} \mathcal{M}(Q_i)^{s_i} = \mathcal{M}(F)$ by the multiplicativity of the Mahler measure and $\mathcal{M}(F^*) \leq \mathcal{M}(F) = 2^{\tilde{\mathcal{O}}(\mu)}$. Hence, in the case where $m_i = s_i$ for all $i = 1, \ldots, d$, the claim eventually follows by taking the logarithm on both sides. Since for each root z of F, $\operatorname{sep}(z, F)$ is upper bounded by two times the maximal absolute value of all roots of F, we have $\operatorname{sep}(z, F) < 2^{\mu+2}$ according to the Cauchy root bound (see e.g. [23]). Thus, the claim also follows for arbitrary integers m_i with $0 \leq m_i \leq s_i$.

We now turn to the analysis of Separate: In the projection step, we have already determined intervals $I := I(\alpha)$ which isolate the real roots α of R^*. Now, each I has to be refined until the inequality (3.2) holds. This ensures that $\Delta_{8r_I}(m_I)$ isolates $\alpha \in I$ from all other roots of R^*, and thus the value $LB(\alpha)$ as defined in (3.3) constitutes a lower bound for $|R(\alpha)|$ on the boundary of $\Delta(\alpha) = \Delta_{2r_I}(m_I)$. In each iteration, we approximate α to a certain number L of bits after the binary point. Then, we check whether the inequality (3.2) holds. If the latter inequality does not hold, we double L and proceed. According to [19, Lemma 2], we have

$$|(R^*)'(z) - \frac{3}{2} \sum\nolimits_{k \geq 2} \left| \frac{(R^*)^{(k)}(m_I)}{k!} \right| r^k > 0$$

if $r < \operatorname{sep}(z_i, R^*)/(4n^4) \leq \operatorname{sep}(z_i, R^*)/(4 \deg(R^*)^2)$.[4] It follows that (3.2) holds for sure if $r_I < \operatorname{sep}(z_i, R^*)/(32n^4) = \operatorname{sep}(z_i, R)/(32n^4)$, thus we have to approximate α to at most $2\log(32n^4/\operatorname{sep}(\alpha, R)) = \mathcal{O}(\log(\operatorname{sep}(\alpha, R)^{-1} + \log n)$ many

[4] In [19, Lemma 2], we considered a constant $\sqrt{2}$ instead of $3/2$. However, the same proof as given for [19, Lemma 2] also applies to the "3/2-case".

bits after the binary point. Due to Theorem 2, $\log \operatorname{sep}(\alpha, R)^{-1}$ is bounded by $\tilde{\mathcal{O}}(n^4 + n^3\tau)$, and thus α has to be approximated to at most $\tilde{\mathcal{O}}(n^4+n^3\tau)$ many bits. For all real roots of R, the latter computation demands for $\tilde{\mathcal{O}}(n^4(n^4 + n^3\tau)) = \tilde{\mathcal{O}}(n^8 + n^7\tau)$ many bit operations according to [18, Theorem 10] (or alternatively [15]). It remains to estimate the cost for evaluating the left side of (3.2). In order to do so, we first compute the Taylor expansion of $(R^*)'$ at $x = m_I$ (i.e. $(R^*)'(x + m_I)$). Since m_I is a dyadic number that is representable by $\mathcal{O}(n^2+n\tau+\log \operatorname{sep}(\alpha, R)^{-1})$ many bits, the cost for this computation is bounded by $\tilde{\mathcal{O}}(\deg(R^*)^2(n^2 + n\tau + \log \operatorname{sep}(\alpha, R)^{-1})) = \tilde{\mathcal{O}}(n^4(n^2 + n\tau + \log \operatorname{sep}(\alpha, R)^{-1}))$, where we use asymptotically fast Taylor shift [22]. Then, x is replaced by $8r_I$ yielding $(R^*)'(m_I + 8r_I x)$. This step constitutes a shift of the k-th (dyadic) coefficient of $f(m_I + x)$ by $k \log(8r_I)$ many bits. The resulting polynomial has dyadic coefficients of bitsize $\mathcal{O}(n^2+n\tau+n^2 \log \operatorname{sep}(\alpha, R)^{-1})$, hence the final evaluation demands for $\mathcal{O}(n^2(n^2 + n\tau + n^2 \log \operatorname{sep}(\alpha, R)^{-1}))$ many bit operations. Summing up over all real roots α of R thus yields the bound

$$\sum_{\alpha} \tilde{\mathcal{O}}(n^4(n^2 + n\tau + \log \operatorname{sep}(\alpha, R)^{-1})) = \tilde{\mathcal{O}}(n^8 + n^7\tau)$$

for the overall cost since there at most n^2 many real roots and $\Sigma(R^*) = \tilde{\mathcal{O}}(n^4 + n^3\tau)$.

It remains to consider the cost for the computation of $LB(\alpha) = 2^{-2\deg R}|R(m_I - 2r_I)|$: We have to evaluate a polynomial of magnitude $(n^2, n(n+\tau))$ at a dyadic number of bitsize $\mathcal{O}(n^2 + n\tau + \log \operatorname{sep}(\alpha)^{-1})$. Namely, the binary representation of m_I needs at most $O(n(n + \tau))$ bits before and $O(\log r_I^{-1}) = O(\log n + \log \operatorname{sep}(\alpha)^{-1})$ bits after the binary point. Hence, for computing $LB(\alpha)$ for all real roots α, we need a number of bit operations bounded by

$$\sum_{\alpha} \tilde{\mathcal{O}}(n^4(n^2 + n\tau + \log \operatorname{sep}(\alpha)^{-1})) = \tilde{\mathcal{O}}(n^8 + n^7\tau).$$

4.3 Validate

4.3.1 Estimating lower and upper bounds

In the final stage, Validate, we have a set of candidate solutions $\mathcal{C}$ and corresponding disjoint polydiscs $\Delta(\alpha, \beta) := \Delta(\alpha) \times \Delta(\beta) \subset \mathbb{C}^2$. Each of the polydiscs contains at most one solution of (1.1), that is, (α, β). The actual solutions of the system are chosen from $\mathcal{C}$ based on the inclusion test from Theorem 1, while the other candidates are excluded using interval arithmetic. We split the complexity analysis of Validate into two parts: First, we estimate $LB(\alpha)$, our lower bound for $|R|$ on the boundary of $\Delta(\alpha)$, as well as the upper bounds for the values of $|u^{(y)}|$ and $|v^{(y)}|$ on $\Delta(\alpha, \beta)$ as needed by the inclusion predicate. This eventually yields a bound on how good each candidate (α, β) must be approximated in order to certify it as a solution or to discard it.

Estimating the lower bounds. We first compute lower and upper bounds for $LB(\alpha) = 2^{-2\deg R}|R(m_I-2r_I)|$ which, in turn, constitutes a lower bound for the values of $|R(z)|$ on the boundary of the disc $\Delta(\alpha) := \Delta_{2r_I}(m_I)$, where $I := I(\alpha)$ is the isolating interval for α obtained in the separation phase; then, similar bounds also apply to $LB(\beta)$, the lower bound for $|R^{(x)}|$ on the boundary of $\Delta(\beta)$, see Section 3 (Separate).

In the analysis of Separate, we have already argued that approximating α to an error of $\operatorname{sep}(\alpha, R)/(32n^4)$ or less guarantees that the inequality (3.2) holds, and thus the disc $\Delta_{8r_I}(m_I)$ isolates α. In each iteration of the refinement, we double the number of bits to which α is approximated and check whether (3.2) holds. Hence, it follows that the so-obtained interval $I(\alpha)$ has width $w_I > (\operatorname{sep}(\alpha, R)/(32n^4))^2$. In addition, since the disc $\Delta_{8r_I}(m_I)$ isolates α, we have $w_I < \operatorname{sep}(\alpha, R)/7$. We fix these bounds for w_I:

$$\frac{\operatorname{sep}(\alpha, R)^2}{1024n^4} < w_I \le \frac{\operatorname{sep}(\alpha, R)}{7}. \tag{4.3}$$

Let us now consider the factorization of R into linear factors, that is, $R(z) = \operatorname{lcf}(R) \cdot \prod_{i=1}^{d}(z-z_i)^{s_i}$, where $z_1, \ldots, d$ denote the distinct complex roots of R and s_i the corresponding multiplicities. Then, with $\alpha = z_j$, we have

$$\frac{\operatorname{sep}(z_j, R)}{4} > |(m_I - 2r_I) - z_j| > \frac{\operatorname{sep}(z_j, R)^2}{2048n^8}$$

and

$$2|z_j - z_i| > |(m_I - 2r_I) - z_i| > \frac{|z_j - z_i|}{2}$$

for all $i \neq j$. Hence, it follows that

$$\begin{aligned}
&LB(\alpha) = LB(z_j) = 2^{-2\deg R} \cdot |R(m_I - 2r_I)| \\
&= 2^{-2\deg R}|\operatorname{lcf}(R)| \cdot |(m_I - 2r_I) - z_j|^{s_j} \prod_{i\neq j} |(m_I - 2r_I) - z_i|^{s_i} \\
&< 2^{-2\deg R}|\operatorname{lcf}(R)| \cdot (\operatorname{sep}(z_j, R)/4)^{s_j} \prod_{i\neq j} |2(z_j - z_i)|^{s_i} \\
&< \operatorname{sep}(z_j, R)^{s_j} \cdot |\operatorname{lcf}(R)| \cdot \prod_{i\neq j} |z_j - z_i|^{s_i} < \operatorname{sep}(z_j, R)^{s_j} \frac{|R^{(s_j)}(z_j)|}{s_j!} \\
&= 2^{\mathcal{O}(n^2+n\tau)} \max(1, |z_j|)^{n^2} \operatorname{sep}(z_j, R)^{s_j} \\
&= 2^{\mathcal{O}(s_j(n^2+n\tau))} \max(1, |z_j|)^{n^2}
\end{aligned} \tag{4.4}$$

since $R^{(s_j)}/(s_j!) \in \mathbb{Z}[x]$ has magnitude $(n^2, n(n+\tau))$, and $\operatorname{sep}(z_j, R) < 2\max_i |z_i| = 2^{\mathcal{O}(n(n+\tau))}$ according to Cauchy's Bound. We can also compute a lower bound for $LB(\alpha)$:

$$\begin{aligned}
LB(\alpha) &> 2^{-2\deg R}|\operatorname{lcf}(R)| \cdot \left(\frac{\operatorname{sep}(z_j, R)^2}{2048n^8}\right)^{s_j} \prod_{i\neq j} \left(\frac{|z_j - z_i|}{2}\right)^{s_i} \\
&> \frac{2^{-3\deg R}}{(2048n^8)^{s_j}} \cdot |\operatorname{lcf}(R)| \operatorname{sep}(z_j, R)^{2s_j} \prod_{i\neq j} |z_j - z_i|^{s_i}
\end{aligned} \tag{4.5}$$

Since we are mainly interested in a bound for the product of all $LB(\alpha)$, we first consider the product

$$\Pi := \prod_{j=1}^{d} \left(\frac{2^{-3\deg R}}{(2048n^8)^{s_j}} \cdot |\operatorname{lcf}(R)| \operatorname{sep}(z_j, R)^{2s_j} \prod_{i\neq j} |z_j - z_i|^{s_i} \right)$$

of the bound in (4.5) over all $j = 1, \ldots, d$. Since $\sum_j s_j = d \le \deg R \le n^2$, it follows that $\prod_{j=1}^{d} \frac{2^{-3\deg R}}{(2048n^8)^{s_j}} = 2^{-\mathcal{O}(n^4)}$. For the product of the remaining factors, we first write $R = \prod_{s=1}^{s_0} Q_s^s$ with square-free, pairwise coprime $Q_s \in \mathbb{Z}[x]$. Since $R^{(s)}/s!$ has integer coefficients, we have

$$1 \le |\operatorname{res}(Q_s, \frac{R^{(s)}}{s!})| = |\operatorname{lcf}(Q_s)|^{\deg(R)-s} \prod_{z:Q_s(z)=0} R^{(s)}(z),$$

and thus

$$\prod_{j=1}^{d}\left(|\operatorname{lcf}(R)|\operatorname{sep}(z_j,R)^{2s_j}\prod_{i\neq j}|z_j-z_i|^{s_i}\right)$$
$$>|\operatorname{lcf}(R)|^d 2^{-2\Sigma(R)}\prod_j\prod_{i\neq j}|z_i-z_j|^{s_j}=2^{-2\Sigma(R)}\prod_j\frac{|R^{(s_j)}(z_i)|}{s_j!}$$
$$=2^{-2\Sigma(R)}\prod_{s=1}^{s_0}|\operatorname{lcf}(Q_s)|^{s-\deg(R)}|\operatorname{res}(Q_s,\frac{R^{(s)}}{s!})|$$
$$>2^{-2\Sigma(R)}|\operatorname{lcf}(R)|\cdot|\operatorname{lcf}(R^*)|^{-\deg(R)}=2^{-\tilde{\mathcal{O}}(n^4+n^3\tau)},$$

where we used that $|\operatorname{lcf}(R)|\leq 2^{\mathcal{O}(n(\log n+\tau))}$, $\deg R\leq n^2$, and $\Sigma(R)=\tilde{\mathcal{O}}(n^4+n^3\tau)$. Hence, Π is lower bounded by $2^{-\tilde{\mathcal{O}}(n^4+n^3\tau)}$. Similar to the computation in (4.4), we can also determine an upper bound for the j-th factor in Π. Namely, we have $\frac{2^{-3\deg R}}{(2048n^8)^{s_j}}<1, \operatorname{sep}(z_j,R)^{s_j}=2^{\mathcal{O}(s_jn(\log n+\tau))}$ and

$$\operatorname{lcf}(R)\prod_{i\neq j}|z_j-z_i|^{s_i}=\frac{|R^{(s_j)}(z_j)|}{s_j!}<2^{\mathcal{O}(n(\log n+\tau))}\max(1,|z_j|)^{n^2}.$$

Thus, for an arbitrary subset $J\subset\{1,\ldots,d\}$, the partial product

$$\Pi':=\prod_{j\in J}\left(\frac{2^{-3\deg R}}{(2048n^8)^{s_j}}\cdot|\operatorname{lcf}(R)|\operatorname{sep}(z_j,R)^{2s_j}\prod_{i\neq j}|z_j-z_i|^{s_i}\right)$$

is smaller than $2^{\mathcal{O}(n^4+n^3\tau)}\prod_{j\in J}\max(1,|z_j|)^n=2^{\mathcal{O}(n^4+n^3\tau)}$ since $\prod_{j\in J}\max(1,|z_j|)^n\leq\mathcal{M}(R)=2^{\mathcal{O}(n(\log n+\tau))}$. Finally, since the product over all $LB(\alpha)$ is lower bounded by a partial product of Π, it follows that $\prod_\alpha LB(\alpha)=2^{-\tilde{\mathcal{O}}(n^4+n^3\tau)}$. The same argument further shows that each $LB(\alpha)$ is lower bounded by $2^{-\tilde{\mathcal{O}}(n^4+n^3\tau)}$ as well.

Estimating the upper bounds. For computing the upper bounds $UB(\alpha,\beta,u^{(y)})$ and $UB(\alpha,\beta,v^{(y)})$ for $|u^{(y)}|$ and $|v^{(y)}|$ on $\Delta(\alpha,\beta)$, we apply Hadamard's inequality to the matrices $U^{(y)}$ and $V^{(y)}$, see Section 3.

In the actual realization, we use interval arithmetic for a box in $\mathbb{C}^2$ which contains $\Delta(\alpha,\beta)$ in order to estimate the absolute values of the respective matrix entries U_{ij} and V_{ij}, and then apply Hadamard's bound. For the complexity analysis, we follow a slightly different but even simpler approach: From the construction of $\Delta(\alpha,\beta)$, the disc $\Delta(\alpha)$ has radius less than $\operatorname{sep}(\alpha,R^{(y)})/4$, and $\Delta(\beta)$ has radius less than $\operatorname{sep}(\beta,R^{(x)})/4$ according to (4.3). Hence, the latter two radii are upper bounded by $2\max\{1,|\alpha|\}$ and $2\max\{1,|\beta|\}$, respectively. Recall that the matrix $U^{(y)}$ is of the form:

$$U^{(y)}=\begin{vmatrix} f_{m_y}^{(y)} & f_{m_y-1}^{(y)} & \cdots & f_0^{(y)} & 0\ \ldots & y^{n_y-1}\\ \vdots & \ddots & \ddots & & \ddots & \vdots\\ 0 & \ldots\ 0 & f_{m_y}^{(y)} & f_{m_y-1}^{(y)} & \cdots & 1\\ g_{n_y}^{(y)} & g_{n_y-1}^{(y)} & \cdots & g_0^{(y)} & 0\ \ldots & 0\\ \vdots & \ddots & \ddots & & \ddots & \vdots\\ 0 & \ldots\ 0 & g_{n_y}^{(y)} & g_{n_y-1}^{(y)} & \cdots & 0 \end{vmatrix},$$

where the polynomials $f_i^{(y)}(x)$ and $g_i^{(y)}(x)$ are of magnitude (n,τ) (see Section 2). Thus, for each point $(\hat{x},\hat{y})\in\Delta(\alpha,\beta)$, the following inequality holds:

$$|f_i^{(y)}(\hat{x})|\leq(n+1)\cdot 2^\tau(2\max(1,|\alpha|))^n,$$

and a similar bound applies to $|g_i^{(y)}(\hat{x})|$ as well. For the last column of $U^{(y)}$, we have: $(\hat{y})^{n_y-1}\leq(2\max(1,|\beta|))^n$, and thus $|U_{ij}^{(y)}(\hat{x},\hat{y})|\leq(n+1)\cdot 2^{\tau+n}\max\{1,|\alpha|,|\beta|\}^n$. By Hadamard's inequality, $|u^{(y)}|=|\det(U^{(y)})|<\prod_i|U_i^{(y)}|_2$ where $|U_i^{(y)}|_2$ is the 2-norm of the i-th row vector of $U^{(y)}$. Hence, when using the latter bounds for the entries of $U^{(y)}$, we obtain an upper bound $UB(\alpha,\beta,u^{(y)})$ for $|u^{(y)}|$ on the polydisc $\Delta(\alpha,\beta)$, such that $UB(\alpha,\beta,u^{(y)})\geq 1$ and

$$\begin{aligned}\log|UB(\alpha,\beta,u^{(y)})|&=\mathcal{O}(n(\tau+n)+n^2\log\max(1,|\alpha|,|\beta|))\\&=\mathcal{O}(n^4+n^3\tau).\end{aligned}\tag{4.6}$$

Again, we are looking for amortization effects: Taking the product of the latter bounds over all candidates (α,β) yields:

$$\begin{aligned}&\sum_{\alpha,\beta}\log UB(\alpha,\beta,u^{(y)})\\&\quad=\sum_{\alpha,\beta}\mathcal{O}(n^2+n\tau)+\sum_{\alpha,\beta}n^2\log\max(1,|\alpha|,|\beta|)\\&\quad\leq\mathcal{O}(n^6+n^5\tau)+n^2\sum_\beta\sum_\alpha\log\max(1,|\alpha|)\\&\quad+n^2\sum_\alpha\sum_\beta\log\max(1,|\beta|))\\&\quad\leq\mathcal{O}(n^6+n^5\tau)+n^2\log\mathcal{M}(R^{(y)})+n^2\log\mathcal{M}(R^{(x)})\\&\quad=\tilde{\mathcal{O}}(n^6+n^5\tau)\end{aligned}\tag{4.7}$$

since there are at most n^2 many α and β. A completely similar argument shows that the bounds in (4.6) and (4.7) are also valid for $UB(\alpha,\beta,v^{(y)})$, $UB(\alpha,\beta,u^{(x)})$ and $UB(\alpha,\beta,v^{(x)})$.

4.3.2 The inclusion test

For a given candidate $(\alpha,\beta)\in\mathcal{C}$ and $\mathcal{B}:=B(\alpha,\beta)=I(\alpha)\times I(\beta)\subset\mathbb{R}^2$ the corresponding candidate box, we define

$$\delta(\mathcal{B}):=\frac{\min(LB(\alpha),LB(\beta))}{\max_{w\in\{u^{(x)},u^{(y)},v^{(x)},v^{(y)}\}}UB(\alpha,\beta,w)}.$$

From the bounds that we have computed in the previous section, we conclude that $\log\delta(\mathcal{B})^{-1}=\tilde{\mathcal{O}}(n^4+n^3\tau)$. According to Theorem 1, $\mathcal{B}$ is isolating for a solution of (1.1) if and only if there exists an $(x_0,y_0)\in\mathcal{B}$ with

$$|f(x_0,y_0)|+|g(x_0,y_0)|<\delta(\mathcal{B}).\tag{4.8}$$

Hence, by contraposition, we must have

$$|f(x_0,y_0)|+|g(x_0,y_0)|\geq\delta(\mathcal{B})\tag{4.9}$$

for all $(x_0,y_0)\in\mathcal{B}$ if $\mathcal{B}$ contains no solution. In order to certify or discard (α,β) as a solution of the system, we evaluate f and g on $\mathcal{B}$ using *interval arithmetic* with precision $\rho:=\rho(\mathcal{B})=\lceil-\log s\rceil$, where $s:=\max(w_{I(\alpha)},w_{I(\beta)})$ is the size of $\mathcal{B}$. As a result of this evaluation, we obtain intervals $\mathfrak{B}(f(\alpha,\beta),\rho)$ and $\mathfrak{B}(g(\alpha,\beta),\rho)$ which contain $f(\mathcal{B})$ and $g(\mathcal{B})$, respectively. The above consideration shows that it suffices to use a precision ρ such that both intervals $\mathfrak{B}(f(\alpha,\beta),\rho)$ and $\mathfrak{B}(g(\alpha,\beta),\rho)$ have width less than $\delta(\mathcal{B})/2$.

Namely, if this happens, then either one of the intervals does not contain zero or we must have $|f(x_0,y_0)| + |g(x_0,y_0)| < \delta(\mathcal{B})$ for all $(x_0,y_0) \in \mathcal{B}$. In the first case, we can discard (α,β), whereas, in the second case, we can guarantee that (α,β) is a solution.

The width of $\mathfrak{B}(f(\alpha,\beta),\rho)$ (and $\mathfrak{B}(g(\alpha,\beta),\rho)$) is directly related to the absolute error induced by the interval arithmetic. In order to bound this error, we briefly outline how the interval arithmetic is performed and refer the reader to [11, Section 4] for more details; cf. [14, Theorem 18] for an alternative approach when using floating point evaluation instead. For a precision $\rho \in \mathbb{N}$ and $x \in \mathbb{R}$, we define:

$$\begin{aligned} \mathrm{down}(x,\rho) &= \{k \cdot 2^{-\rho} \in \mathbb{R} : k = \lfloor x \cdot 2^{\rho} \rfloor\}, \\ \mathrm{up}(x,\rho) &= \{k \cdot 2^{-\rho} \in \mathbb{R} : k = \lceil x \cdot 2^{\rho} \rceil\}. \end{aligned} \tag{4.10}$$

That is, x is included in the interval $\mathfrak{B}(x,\rho) := [\mathrm{down}(x,\rho), \mathrm{up}(x,\rho)]$. For simplicity, we omit the precision parameter ρ and write $\mathrm{up}(x)$ or $\mathfrak{B}(x)$. Arithmetic operations on approximate numbers obey the rules of classical interval arithmetic; for $x,y \in \mathbb{R}$, we define:

$$\begin{aligned} \mathfrak{B}(x) + \mathfrak{B}(y) &:= [\mathrm{down}(x) + \mathrm{down}(y), \mathrm{up}(x) + \mathrm{up}(y)], \\ \mathfrak{B}(x) - \mathfrak{B}(y) &:= [\mathrm{down}(x) - \mathrm{up}(y), \mathrm{up}(x) - \mathrm{down}(y)], \\ \mathfrak{B}(x) \cdot \mathfrak{B}(y) &:= \Big[\mathrm{down}(\min_{i,j=\{1,2\}} \{H_i(x)H_j(y)\}), \\ &\qquad \mathrm{up}(\max_{i,j=\{1,2\}} \{H_i(x)H_j(y)\})\Big] \end{aligned}$$

with $H_1(x) = \mathrm{down}(x)$, and $H_2(x) = \mathrm{up}(x)$. Using these rules for $F \in \mathbb{R}[x]$ and $x_0 \in \mathbb{R}$, $\mathfrak{B}(F(x_0),\rho)$ can be evaluated using the Horner's scheme: $\mathfrak{B}(F(x_0)) = \mathfrak{B}(F_0) + \mathfrak{B}(x_0) \cdot (\mathfrak{B}(F_1) + \mathfrak{B}(x_0) \cdot (\mathfrak{B}(F_2) + \ldots))$. The next lemma provides a bound on the error that is induced by polynomial evaluation with precision ρ.

Lemma 1 *Let $F \in \mathbb{R}[x]$ be a polynomial of degree N with coefficients of absolute value less than 2^{μ}, $c \in \mathbb{R}$ with $|c| \leq 2^{\upsilon}$, and $\rho \in \mathbb{N}$. Then,*

$$|F(c) - H(F(c),\rho)| \leq 2^{-\rho+1} 2^{\mu} 2^{N\upsilon} (N+1)^2,$$

where $H = \{\mathrm{down}, \mathrm{up}\}$. In particular, $\mathfrak{B}(F(c),\rho)$ has width $2^{-\rho+2}(N+1)^2 2^{\mu+N\upsilon}$ or less. For a proof, see [11, Lem. 3]. □

In particular, this lemma asserts that the absolute error which results from approximate polynomial evaluation is *linear* in $2^{-\rho}$ and *of degree n* in the absolute value of the input. A straight forward computation shows that evaluating $f(\alpha,\beta)$ with precision ρ induces an absolute error of less than $2^{-\rho+1}(n+1)^2 2^{\tau} \max\{1,|\alpha|,|\beta|\}^n$, thus the width of $\mathfrak{B}(f(\alpha,\beta),\rho)$ is bounded by $2^{-\rho+2}(n+1)^2 2^{\tau} \max(1,|\alpha|^n,|\beta|^n)$. The same bound also applies to $\mathfrak{B}(g(\alpha,\beta),\rho)$.

It follows that our inclusion/exclusion test must succeed for any precision ρ less than

$$\rho(\mathcal{B}) := \log(8(n+1)^2 2^{\tau} \max(1,|\alpha|^n,|\beta|^n)\delta(\mathcal{B})^{-1})$$

because, then, both intervals $\mathfrak{B}(f(\alpha,\beta),\rho)$ and $\mathfrak{B}(f(\alpha,\beta),\rho)$ have width less than $\delta(\mathcal{B})/2$. Since we double the working precision ρ in each step, we eventually succeed for a

$$\begin{aligned} \rho &< 2\rho(\mathcal{B}) = \mathcal{O}(\log n + \tau + n \log \max(1,|\alpha|,|\beta|) - \log \delta(\mathcal{B})) \\ &= \tilde{\mathcal{O}}(n^4 + n^3\tau). \end{aligned}$$

In addition, we have to refine the isolating intervals $I(\alpha)$ and $I(\beta)$ to a width $2^{-\rho} = 2^{-\tilde{\mathcal{O}}(n^4+n^3\tau)}$. In our analysis of Separate, we have already seen that refining the isolating intervals for all real roots of $R^{(y)}$ (and $R^{(x)}$) to a width of $2^{-\tilde{\mathcal{O}}(n^4+n^3\tau)}$ demands for $\tilde{\mathcal{O}}(n^8+n^7\tau)$ many bit operations.

It remains to bound the cost for evaluating $\mathfrak{B}(f(\alpha,\beta),\rho)$ and $\mathfrak{B}(f(\alpha,\beta),\rho)$: Since we have to perform $\mathcal{O}(n^2)$ many multiplications and additions with dyadic numbers whose binary representations need $\mathcal{O}(\tau + n \log \max(1,|\alpha|,|\beta|) - \delta(\mathcal{B}(\alpha,\beta)))$ many bits, the latter computation demands for

$$\tilde{\mathcal{O}}(n^2(\tau + n \log \max(1,|\alpha|,|\beta|) - \rho(\mathcal{B}(\alpha,\beta)))) \tag{4.11}$$

many bit operations. Hence, for the bit complexity of the polynomial evaluations at all (α,β), we obtain the bound

$$\begin{aligned} &\sum_{\alpha,\beta} \tilde{\mathcal{O}}(n^2(\tau + n \log \max(1,|\alpha|,|\beta|) - \delta(\mathcal{B}(\alpha,\beta)))) \\ &= \tilde{\mathcal{O}}(n^6\tau + n^3 \sum_{\alpha,\beta} \log \max(1,|\alpha|,|\beta|) - n^2 \sum_{\alpha,\beta} \delta(\mathcal{B}(\alpha,\beta))) \\ &= \tilde{\mathcal{O}}(n^7 + n^6\tau - n^2 \sum_{\alpha,\beta} \delta(\mathcal{B}(\alpha,\beta))), \end{aligned}$$

where we use the same argument as in (4.7) to bound the sum of all $\log \max(1,|\alpha|,|\beta|)$. The following computation further shows that $-\sum_{\alpha,\beta} \log \delta(\mathcal{B}(\alpha,\beta)) = \tilde{\mathcal{O}}(n^6+n^5\tau)$: Using the upper bound (4.4) for $LB(\alpha)$ and $LB(\beta)$ yields

$$\begin{aligned} &\log(\min(LB(\alpha),LB(\beta)))^{-1} \leq \log LB(\alpha)^{-1} + \log LB(\beta)^{-1} \\ &+ 2n^2 \cdot \log \max(1,|\alpha|,|\beta|) + \mathcal{O}((s_\alpha + s_\beta)(n^2+n\tau)), \end{aligned}$$

where s_α denotes the multiplicity of α as a root of $R^{(y)}$, and s_β the multiplicity of β as a root of $R^{(x)}$. Hence, the bound $\tilde{\mathcal{O}}(n^6+n^5\tau)$ for the sum over all $\log(\min(LB(\alpha),LB(\beta)))^{-1}$ follows from

$$\begin{aligned} &\sum_{\alpha,\beta} \log LB(\alpha)^{-1} + \log LB(\beta)^{-1} = \sum_{\beta} \sum_{\alpha} \log LB(\alpha)^{-1} + \\ &+ \sum_{\alpha} \sum_{\beta} \log LB(\beta)^{-1} \leq -n^2 (\sum_{\alpha} \log LB(\alpha) + \sum_{\beta} \log LB(\beta)) \\ &= -n^2 (\log \prod_{\alpha} LB(\alpha) + \log \prod_{\beta} LB(\beta)) = \tilde{\mathcal{O}}(n^6+n^5\tau), \end{aligned}$$

and

$$\begin{aligned} &\sum_{\alpha,\beta} 2n^2 \log \max(1,|\alpha|,|\beta|) + \mathcal{O}((s_\alpha+s_\beta)(n^2+n\tau)) \\ &= \tilde{\mathcal{O}}(n^6 + n^5\tau + (n^4+n^3\tau) \cdot (\sum_{\alpha} s_\alpha + \sum_{\beta} s_\beta)) = \tilde{\mathcal{O}}(n^6+n^5\tau). \end{aligned}$$

In addition, the result from (4.7) shows that

$$\begin{aligned} &\sum_{\alpha,\beta} \log \max_{w \in \{u^{(x)},u^{(y)},v^{(x)},v^{(y)}\}} UB(\alpha,\beta,w) \\ &\quad \leq \sum_{\alpha,\beta} \log UB(\alpha,\beta,u^{(x)}) + \sum_{\alpha,\beta} \log UB(\alpha,\beta,u^{(y)}) \\ &\quad + \sum_{\alpha,\beta} \log UB(\alpha,\beta,v^{(x)}) + \sum_{\alpha,\beta} \log UB(\alpha,\beta,v^{(y)}) \\ &\quad = \tilde{\mathcal{O}}(n^6+n^5\tau). \end{aligned} \tag{4.12}$$

Thus, the claimed bound for $-\sum_{\alpha,\beta} \log \delta(\mathcal{B}(\alpha,\beta))$ follows from our definition of $\delta(\mathcal{B}(\alpha,\beta))$.

We conclude that $\tilde{\mathcal{O}}(n^8 + n^7\tau)$ determines the overall bit complexity of BISOLVE.

5. CONCLUSIONS

We have derived the bound $\tilde{\mathcal{O}}(n^8 + n^7\tau)$ for the bit complexity of isolating the real solutions of a bivariate polynomial system. To the best of our knowledge, the latter bound considerably improves upon the best known complexity bounds for this fundamental task. However, it seems that an even more involved analysis may yield a slight improvement to $\tilde{\mathcal{O}}(n^7\tau)$ bit operations. In particular, this would require to remove the "N^2-term" in our bound for $\Sigma(F)$ as given in Theorem 2.

The bottleneck in our analysis stems from the fact that we treat the resultant polynomial R as a general polynomial of magnitude $(n^2, n(\tau + \log n))$, thus yielding a worst case separation of $2^{-\tilde{\mathcal{O}}(n^4+n^3\tau)}$ for the roots of R. Hence, as long as no improvement for the root isolation step is achieved, it seems to be very difficult to further improve upon the given bound for solving a bivariate polynomial system when using an elimination approach. In practice, we never observed that the roots of the resultant polynomial have such a bad separation, thus, the question arises whether isolating and refining the roots of an elimination polynomial is possibly easier than of a general polynomial of the same magnitude.

A recent exact and complete algorithm [1] uses BISOLVE to compute arrangements of planar algebraic curves. We consider the presented analysis as a first step to derive corresponding complexity results for the arrangement computation which improve upon the results as given in [10, 12]. Finally, it seems reasonable to extend BISOLVE for solving zero-dimensional polynomial systems with more than two variables. We aim to formulate such an algorithm and to analyze its complexity in a similar way as done for BISOLVE in this paper.

6. REFERENCES

[1] E. Berberich, P. Emeliyanenko, A. Kobel, and M. Sagraloff. Arrangement computation of planar algebraic curves. In *Proceedings of the Workshop on Symbolic an Numerical Computation (SNC)*, pages 88–99, 2011.

[2] E. Berberich, P. Emeliyanenko, and M. Sagraloff. An Elimination Method for Solving Bivariate Polynomial Systems: Eliminating the Usual Drawbacks. In *ALENEX '11*, pages 35–47. SIAM, 2011.

[3] E. Berberich, M. Kerber, and M. Sagraloff. An efficient algorithm for the stratification and triangulation of algebraic surfaces. *Computational Geometry: Theory and Applications*, 43:257–278, 2010. Special issue on SoCG'08.

[4] J. Cheng, S. Lazard, L. Penaranda, M. Pouget, F. Rouillier, and E. Tsigaridas. On the topology of planar algebraic curves. In *SCG '09: Proc. of the 25th Annual Symposium on Computational Geometry*, pages 361–370, New York, NY, USA, 2009. ACM.

[5] J.-S. Cheng, X.-S. Gao, and J. Li. Root isolation for bivariate polynomial systems with local generic position method. In *ISSAC '09*, pages 103–110, New York, NY, USA, 2009. ACM.

[6] D. I. Diochnos, I. Z. Emiris, and E. P. Tsigaridas. On the asymptotic and practical complexity of solving bivariate systems over the reals. *Journal of Symbolic Computation*, 44(7):818–835, 2009.

[7] A. Eigenwillig, L. Kettner, W. Krandick, K. Mehlhorn, S. Schmitt, and N. Wolpert. A Descartes algorithm for polynomials with bit-stream coefficients. In *CASC '05*, volume 3718 of *LNCS*, pages 138–149, 2005.

[8] K. Geddes, S. Czapor, and G. Labahn. *Algorithms for computer algebra*. Kluwer Academic Publishers, Boston/Dordrecht/London, 1992.

[9] L. González-Vega and M. E. Kahoui. An Improved Upper Complexity Bound for the Topology Computation of a Real Algebraic Plane Curve. *Journal of Complexity*, 12(4):527–544, 1996.

[10] M. Kerber. *Geometric Algorithms for Algebraic Curves and Surfaces*. PhD thesis, Universität des Saarlandes, Saarbrücken, Germany, 2009.

[11] M. Kerber and M. Sagraloff. Efficient real root approximation. In *ISSAC '11*, pages 209–216, 2011. see http://arxiv.org/abs/1104.1362v1 for an extended version.

[12] M. Kerber and M. Sagraloff. A worst-case bound for topology computation of algebraic curves. *CoRR*, abs/1104.1510, 2011. to appear in the Journal of Symbolic Computation.

[13] T. Lickteig and M.-F. Roy. Sylvester-Habicht Sequences and Fast Cauchy Index Computation. *Journal of Symbolic Computation*, 31(3):315–341, 2001.

[14] K. Mehlhorn, R. Osbild, and M. Sagraloff. A general approach to the analysis of controlled perturbation algorithms. *Comput. Geom.*, 44(9):507–528, 2011.

[15] V. Y. Pan. Solving a polynomial equation: some history and recent progress. *SIAM Review*, 39(2):187–220, 1997.

[16] D. Reischert. Asymptotically fast computation of subresultants. In *ISSAC '97*, pages 233–240, New York, NY, USA, 1997. ACM.

[17] M. Sagraloff. On the complexity of real root isolation. *CoRR*, abs/1011.0344, 2010. submitted.

[18] M. Sagraloff. When newton meets descartes - a simple and fast algorithm to isolate the real roots of a polynomial. *CoRR*, abs/1109.6279, 2011. submitted in parallel to ISSAC'12, for an online version, see also *http://www.mpi-inf.mpg.de/ msagralo/NEWDSC.pdf*.

[19] M. Sagraloff and C. Yap. A simple but exact and efficient algorithm for complex root isolation. In *ISSAC '11*, pages 353–360, 2011.

[20] A. Schönhage. The fundamental theorem of algebra in terms of computational complexity, 1982. Manuscript, Department of Mathematics, University of Tübingen. Updated 2004.

[21] M. van Hoeij and M. B. Monagan. A modular GCD algorithm over number fields presented with multiple extensions. In *ISSAC '02*, pages 109–116, 2002.

[22] J. von zur Gathen and J. Gerhard. Fast algorithms for taylor shifts and certain difference equations. In *ISSAC '97*, pages 40–47, New York, NY, USA, 1997. ACM.

[23] C. K. Yap. *Fundamental Problems in Algorithmic Algebra*. Oxford University Press, 2000.