Certifying solutions to overdetermined and singular polynomial systems over $\mathbb{Q}$

Tulay Ayyildiz Akoglu[a,1], Jonathan D. Hauenstein[b,2], Agnes Szanto[a,3]

[a] *Department of Mathematics, North Carolina State University, Campus Box 8205, Raleigh, NC, 27965, USA.*

[b] *Department of Applied and Computational Mathematics and Statistics, University of Notre Dame, Notre Dame, IN, 46556, USA.*

Abstract

This paper is concerned with certifying that a given point is near an exact root of an overdetermined or singular polynomial system with rational coefficients. The difficulty lies in the fact that consistency of overdetermined systems is not a continuous property. Our certification is based on hybrid symbolic-numeric methods to compute the exact *rational univariate representation* (RUR) of a component of the input system from approximate roots. For overdetermined polynomial systems with simple roots, we compute an initial RUR from approximate roots. The accuracy of the RUR is increased via Newton iterations until the exact RUR is found, which we certify using exact arithmetic. Since the RUR is well-constrained, we can use it to certify the given approximate roots using α-theory. To certify isolated singular roots, we use a determinantal form of the *isosingular deflation*, which adds new polynomials to the original system without introducing new variables. The resulting polynomial system is overdetermined, but the roots are now simple, thereby reducing the problem to the overdetermined case. We prove that our algorithms have complexity that are polynomial in the input plus the output size upon successful convergence, and we use worst case upper bounds for termination when our iteration does not converge to an exact RUR. Examples are included to demonstrate the approach.

[1] This research was partly supported by a Turkish grant.

[2] This research was partly supported by DARPA YFA and NSF grant DMS-1262428.

[3] This research was partly supported by NSF grant CCF-1217557.

1. Introduction

In their recent article [24], F. Sottile and the second author showed that one can get an efficient and practical root certification algorithm using α-theory (c.f. [59, 8]) for well-constrained polynomials systems. The same paper also considers overdetermined systems over the rationals and show how to use α-theory to certify that a given point is *not* an approximation of any exact roots of the system. However, to certify that a point is near an exact root, one can use universal lower bounds for the minimum of positive polynomials on a disk That paper concludes that all known bounds were "too small to be practical."

A closer look at the literature on lower bounds for the minimum of positive polynomials over the roots of zero-dimensional rational polynomial systems reveals that they all reduce the problem to the univariate case and use univariate root separation bounds (see, for example, [12, 27, 11, 28]). This led to the idea of directly using an exact univariate representation for certification of the input system instead of using universal lower bounds that are often very pessimistic. For example, the overdetermined system

$$f_1 := x_1 - \frac{1}{2}, f_2 := x_2 - x_1^2, \ldots, f_n := x_n - x_{n-1}^2, f_{n+1} := x_n$$

has no common roots, but the value of f_{n+1} on the common root of $f_1, \ldots, f_n$ is double exponentially small in n. While universal lower bounds cover these artificial cases, our approach, as we shall see, has the ability to terminate early in cases when the witness for our input instance is small.

In principle, one can compute such a univariate representation using purely algebraic techniques, for example, by solving large linear systems corresponding to resultant or subresultant matrices (see, for example, [62]). However, this purely symbolic method would again lead to worst case complexity bounds. Instead, we propose a hybrid symbolic-numeric approach, using the approximate roots of the system, as well as exact univariate polynomial remaindering over $\mathbb{Q}$. We expect that our method will make the certification of roots of overdetermined systems practical for cases when the universal lower bounds are too pessimistic, or when the actual size of our univariate representation is significantly smaller than in the worst case.

Consider an overdetermined system $\mathbf{f} = (f_1, \ldots, f_m) \in \mathbb{Q}[x_1, \ldots, x_n]$ for some $m > n$, and assume that the ideal $\mathcal{I} := \langle f_1, \ldots, f_m \rangle$ is radical and zero dimensional. With these assumptions Roullier's Rational Univariate Representation (RUR) for $\mathcal{I}$ exists [54], as well as for any component of $\mathcal{I}$

over the rationals. Since the polynomials in the RUR have also rational coefficients, we can hope to compute them exactly, unlike the possibly irrational coordinates of the common roots of $\mathcal{I}$. With the exact RUR, which is a well-constrained system of polynomials, we can use α-theory as in [24] to certify that a given point is an approximate root for the RUR, and thus for our original system $\mathbf{f}$. Alternatively, from an RUR, we can compute Hermite matrices for our system which can be used to certify that there is an exact root within ϵ of our given point.

Our numerical method to compute an RUR for $\mathcal{I}$ or for a rational component of $\mathcal{I}$ consists of three steps:

1. compute approximations of all isolated roots to a given accuracy of a random well-constrained (square) set of linear combinations of the polynomials $f_1, \ldots, f_m$ using homotopy continuation (e.g., see [4, 60]);
2. among the roots computed in Step 1, choose a subset that is a candidate to be approximations to roots of $\mathcal{I}$ or a rational component of $\mathcal{I}$ – for this step, we can only give heuristics on how to proceed;
3. construct a rational RUR from the approximate roots chosen in Step 2 using Lagrange interpolation and rational number reconstruction, and check whether the polynomials in $\mathbf{f}$ reduce to zero modulo the computed RUR. If yes, terminate, if not, continue to compute iteratively more accurate approximations of the RUR.

We propose two methods to increase the accuracy of the computed RUR. The first one, called *Local Newton Iteration*, simply repeats Step 3 with ever more accurate root approximations. The second one, called *Global Newton Iteration*, is a version of Hensel's lifting and was used in the non-Archimedean metric in, for example, [30, 63, 66, 20]. We apply it in the usual Euclidean metric on coefficient vectors with rational entries. In [23], it was shown that the two methods are equivalent when using the p-adic non-Archimedean metric but different under the Euclidean metric. Furthermore, [23] gives sequential and parallel complexity analysis for the iterations of both of these methods.

Note that termination of the above steps depends on the choice of approximate roots in Step 2. In particular, for wrong choices, the iterated RUR will never converge to an exact RUR of a rational component of $\mathcal{I}$. We will exhibit several approaches for termination: either by an *a priori* bound on the height of the coefficients of the exact RUR, or by incorporating certification of non-roots as in [24].

Another approach that eliminates the heuristic choice in Step 2 is to compute an exact RUR for all roots computed in Step 1. From this RUR, using symbolic techniques, one can compute an exact RUR for $\mathcal{I}$. This method will always converge to an exact RUR, provided that the roots computed in Step 1 are certified approximate roots of the well-constrained system and the round off error is negligible. Even though this technique is less efficient than the one described above, one may choose it if the method summarized above fails. Note that this hybrid symbolic-numeric method is highly parallelizable, which may make it preferable compared to purely symbolic methods to compute an RUR for $\mathcal{I}$ via elimination.

In the second part of the paper, we consider certifying isolated singular roots of a rational polynomial system. Due the behavior of Newton's method near singular roots (e.g., see [21]), standard techniques in α-theory can not be applied to certify such roots even if the polynomial system is well-constrained. The key tool to handle such multiple roots is called *deflation*. Deflation techniques "regularize" the system thereby creating a new polynomial system which has a simple root corresponding to the multiple root of the original system [15, 25, 39, 40, 49, 50]. In this work, we will focus on using a determinantal form of the *isosingular deflation* [25], in which one simply adds new polynomials to the original system without introducing new variables. The new polynomials are constructed based on exact information that one can obtain from a numerical approximation of the multiple root. In particular, if the original system had rational coefficients, the new polynomials which remove the multiplicity information also have rational coefficients. Thus, this technique has reduced us to the case of an overdetermined system over $\mathbb{Q}$ in the original set of variables that has a simple root.

1.1. Related work

One of the applications of certifying near-exact solutions of overdetermined and singular systems is the certification of computerized proofs of mathematical theorems, similarly as the software package `alphaCertified` of [24] was applied to confirm a conjecture of Littlewood in [9].

As we mentioned in the Introduction, for well-constrained (square) polynomial systems, the paper [24] applies α-theory (c.f. [59, 8]) to obtain an efficient and practical root certifying algorithm. It is proposed to use universal lower bounds to certify approximate roots of overdetermined systems. In [16], an α-theory for overdetermined systems was developed. However, since this approach cannot distinguish between local minimums and roots, we can not use it to certify roots of overdetermined systems. There is an extended body of literature on using interval arithmetic and optimization methods to certify the existence or non-existence of the solutions of well-constrained systems with guaranteed accuracy, e.g., [36, 46, 56, 67, 34, 48]. Techniques to certify each step of path tracking in homotopy continuation for well-constrained systems using α-theory are presented in [5, 6, 22]. Recently, a certification method of real roots of positive dimensional systems was studied in [68].

Related to the certification problems under consideration is the problem of finding certified sum of squares (SOS) decompositions of rational polynomials. In [52, 53, 31, 32], they turn SOS decompositions given with approximate (floating point) coefficients into rational ones, assuming that the feasible domain of the corresponding semidefinite feasibility problem has nonempty interior. In [45], they adapt these techniques to the degenerate case, however they also require a feasible solution with rational coefficients exists. The certification of more general polynomial, semi-algebraic and transcendental optimization problems were considered in [2]. In [57], they compute rational points in semi-algebraic sets and give a method to decide if a polynomial can be expressed as an SOS of rational polynomials. Note that we can straightforwardly translate the certification of approximate roots of overdetermined polynomial systems into polynomial optimization problems over compact convex sets (using a ball around the approximate root), however, we cannot guarantee a rational feasible solution. Instead, we propose to construct the rational representation of several irrational roots that form a rational component of the input system. Note also that the coefficient vector of the RUR of the input ideal is the solution of a linear system corresponding to multiples of the input polynomials that appears as the affine constraints in Lasserre's relaxation in [38], and thus can be computed using purely symbolic methods. However, here we propose a more efficient symbolic-numeric approach, constructing an RUR from approximate roots, and allowing RUR's of smaller rational components as well.

The idea of computing an exact solution from numerical approximations is not new. In [13], they give a randomized algorithm that for a given approximate zero z corresponding to an exact zero ξ of a polynomial system $F_1, \ldots, F_n \in \mathbb{Q}[x_1, \ldots, x_n]$ finds the RUR of the irreducible component $V_\xi \subset V(F_1, \ldots, F_n)$ in polynomial time depending on n, the heights of F_i, the degree of the minimal polynomial of a primitive element for V_ξ, and the height of this minimal polynomial. The algorithm in [13] uses the algorithm in [33] for the construction of the minimal polynomial of a given approximate algebraic number. However, the algorithm in [33] requires an upper bound for the height of the algebraic number, and this bound is used in the construction of the lattice that they apply LLL lattice basis reduction. To get such bound *a priori*, we would need to use universal bounds for the height. In order to get an incremental algorithm with early termination for the case when the output size is small, one can modify the algorithm in [33] to be incremental, but that would require multiple application of the lattice basis reduction algorithm. Alternatively, one can apply the PSLQ algorithm as in [17], which is incremental and does not require an *a priori* height bound. The main point of the approach in this paper is that we assume to know *all* approximate roots of a rational component, so in this case we can compute the exact RUR much more efficiently, and in parallel, as we prove in this paper and in [23]. So instead of multiple LLL lattice basis reduction, we propose a cheaper lifting and checking technique.

Related literature on certification of singular zeroes of polynomial systems include [35, 55, 44, 41, 42]. However, these approaches differ from ours in the sense that they certify singular roots of some small perturbation of the input polynomial system while in the present paper we certify singular roots of an exact polynomial system with rational coefficients.

2. Certifying roots of overdetermined systems

As described in the Introduction, the philosophy behind our method is to avoid using universal worst case lower bounds as certificates. Instead, we aim to compute an exact univariate representation for our system that can be used for the certification. The Introduction outlined the steps of our proposed method to compute such a representation with details presented here.

2.1. Preliminaries

Let us start with recalling the notion of Roullier's *Rational Univariate Representation (RUR)*, originally defined in [54]. We follow here the approach and notation in [20].

Let $\mathbf{f} = (f_1, \ldots, f_m) \in \mathbb{Q}[x_1, \ldots, x_n]$ for some $m \geq n$, and assume that the ideal $\mathcal{I} := \langle f_1, \ldots, f_m \rangle$ is radical and zero dimensional. The factor ring $\mathbb{Q}[x_1, \ldots, x_n]/\mathcal{I}$ is a finite dimensional vector space over $\mathbb{Q}$, and we denote

$$\delta := \dim_{\mathbb{Q}} \mathbb{Q}[x_1, \ldots, x_n]/\mathcal{I}.$$

Furthermore, for almost all $(\lambda_1, \dots, \lambda_n) \in \mathbb{Q}^n$ (except a Zariski closed subset), the linear combination

$$u(x_1, \dots, x_n) := \lambda_1 x_1 + \cdots + \lambda_n x_n$$

is a *primitive element* of $\mathcal{I}$, i.e. the powers $1, u, u^2, \dots, u^{\delta-1}$ form a linear basis for $\mathbb{Q}[x_1, \dots, x_n]/\mathcal{I}$ (c.f. [54]). Let $q(T) \in \mathbb{Q}[T]$ be the minimal polynomial of u in $\mathbb{Q}[x_1, \dots, x_n]/\mathcal{I}$, and let $x_i = v_i(u)$ be the polynomials expressing the coordinate function as linear combinations of the powers of u in $\mathbb{Q}[x_1, \dots, x_n]/\mathcal{I}$. Note that from $u = \lambda_1 x_1 + \cdots + \lambda_n x_n$, we must have

$$u = \lambda_1 v_1(u) + \cdots + \lambda_n v_n(u).$$

and

$$\langle q(T),\ x_1 - v_1(T), \dots, x_n - v_n(T)\rangle = \langle \mathcal{I}, T - (\lambda_1 x_1 + \cdots + \lambda_n x_n)\rangle.$$

Definition 2.1. *Let $\mathcal{I} = \langle f_1, \dots, f_m\rangle \subset \mathbb{Q}[x_1, \dots, x_n]$ be as above. The Rational Univariate Representation (RUR) of $\mathcal{I}$ is given by*

- *a primitive element $u = \lambda_1 x_1 + \cdots + \lambda_n x_n$ of $\mathcal{I}$ for some $\lambda_1, \dots, \lambda_n \in \mathbb{Q}$;*
- *the minimal polynomial $q(T) \in \mathbb{Q}[T]$ of u in $\mathbb{Q}[x_1, \dots, x_n]/I$, a monic square-free polynomial of degree δ;*
- *the parametrization of the coordinates of the zeroes of $\mathcal{I}$ by the zeroes of q, given by*
$$v_1(T), \dots, v_n(T) \in \mathbb{Q}[T]$$
all degree at most $\delta - 1$ and satisfying
$$\lambda_1 v_1(T) + \cdots + \lambda_n v_n(T) \equiv T \mod q(T).$$

In the algorithms below, we compute an RUR that may not generate the same ideal $\mathcal{I}$ as our input polynomials, nevertheless it contains $\mathcal{I}$, i.e., the polynomials $f_1, \dots, f_m$ vanish modulo the RUR. In this case, the RUR will generate a *component of $\mathcal{I}$*. The common roots of the RUR of a component of $\mathcal{I}$ correspond to a subset of $V(\mathcal{I})$. For any subset $V \subset V(\mathcal{I})$, one can construct an RUR of the corresponding component of $\mathcal{I}$ that satisfies the following definition. Finally, we distinguish between RUR's of components and *rational components of $\mathcal{I}$*.

Definition 2.2. *Let $\mathcal{I} = \langle f_1, \ldots, f_m \rangle \subset \mathbb{Q}[x_1, \ldots, x_n]$ be as above and fix a primitive element $\lambda_1 x_1 + \cdots + \lambda_n x_n \in \mathbb{Q}[x_1, \ldots, x_n]$ of $\mathcal{I}$. The polynomials*

$$T - \lambda_1 x_1 + \cdots + \lambda_n x_n,\ q(T),\ v_1(T), \ldots, v_n(T) \tag{1}$$

form a Rational Univariate Representation (RUR) of a component of $\mathcal{I}$ *if it satisfies the following properties:*

- *$q(T) \in \mathbb{C}[T]$ is a monic square-free polynomial of degree $d \leq \delta$,*
- *$v_1(T), \ldots, v_n(T) \in \mathbb{C}[T]$ are all degree at most $d-1$ and satisfy*

$$\lambda_1 v_1(T) + \cdots + \lambda_n v_n(T) \equiv T \mod q(T),$$

- *for all $i = 1, \ldots, m$ we have*

$$f_i(v_1(T), \ldots, v_n(T)) \equiv 0 \mod q(T).$$

If, in addition, $q(T), v_1(T), \ldots, v_n(T) \in \mathbb{Q}[T]$ are rational polynomials, we call (1) a RUR of a rational component of $\mathcal{I}$.

First note that the set

$$\{q(T),\ x_1 - v_1(T), \ldots, x_n - v_n(T)\} \tag{2}$$

forms a Gröbner basis for the ideal it generates with respect to the lexicographic monomial ordering defined by $T < x_1 < \cdots < x_n$ and is well-constrained. When we have an RUR of $\mathcal{I}$, (2) is a Gröbner basis for

$$\langle \mathcal{I}, T - (\lambda_1 x_1 + \cdots + \lambda_n x_n) \rangle.$$

In particular, one can compute an RUR of $\mathcal{I}$ with purely symbolic elimination methods, such as Buchberger's algorithm, or resultant based methods. Moreover, since (2) is well-constrained, one can apply standard α-theoretic tools to certify solutions.

Let

$$T - (\lambda_1 x_1 + \cdots + \lambda_n x_n),\ q(T),\ v_1(T), \ldots, v_n(T)$$

be an RUR for $\mathcal{I}$ and

$$q'(T),\ v_1'(T), \ldots, v_n'(T) \in \mathbb{Q}[T]$$

be an RUR for a rational component of $\mathcal{I}$ with respect to the same primitive element. Then, we have

$$q(T) \equiv 0 \text{ and } v_i(T) \equiv v_i'(T) \mod q'(T) \text{ for } i = 1, \ldots, n.$$

This shows that we can obtain an RUR for a rational component of $\mathcal{I}$ from an RUR of $\mathcal{I}$ using symbolic univariate polynomial factorization over $\mathbb{Q}$ and univariate polynomial remainders (we only use this in our complexity estimates, not in the computation that we propose).

Next, let us recall the relationship between the RUR of a component of $\mathcal{I}$ and the corresponding (exact) roots. Let $V := \{\xi_1, \ldots, \xi_d\} \subseteq V(\mathcal{I}) \subset \mathbb{C}^n$ be the exact roots of a component of $\mathcal{I}$, and denote $\xi_i = (\xi_{i,1}, \ldots, \xi_{i,n})$ for $i = 1, \ldots, d$. Then for any n-tuple $(\lambda_1, \ldots \lambda_n) \in \mathbb{Q}^n$ such that

$$\lambda_1 \xi_{i,1} + \cdots + \lambda_n \xi_{i,n} \neq \lambda_1 \xi_{j,1} + \cdots + \lambda_n \xi_{j,n} \quad \text{ if } i \neq j,$$

we can define a primitive element $u = \lambda_1 x_1 + \cdots + \lambda_n x_n$ for V. Since all roots are distinct, such primitive element exists and a randomly chosen n-tuple from a sufficiently large finite subset of $\mathbb{Q}^n$ will have this property with high probability (c.f. [54]). Fix such $(\lambda_1, \ldots \lambda_n) \in \mathbb{Q}^n$, and define

$$\mu_i := \lambda_1 \xi_{i,1} + \cdots + \lambda_n \xi_{i,n}, \quad i = 1, \ldots d. \tag{3}$$

Then,

$$q(T) := \prod_{i=1}^{d} (T - \mu_i) \tag{4}$$

is the unique monic polynomial of degree d vanishing at the values of u corresponding to each point in V. For each j, $v_j(T)$ in the parametrization of the coordinates is the unique Lagrange interpolant satisfying

$$v_j(\mu_i) = \xi_{i,j} \quad \text{ for } i = 1, \ldots, d. \tag{5}$$

Unfortunately, the common roots of $\mathcal{I}$ may be irrational, so numerical methods will compute only approximations to them. However, the coefficients of the RUR of $\mathcal{I}$ are rational numbers, so we may compute them exactly. This is not always true for RUR's of components of $\mathcal{I}$, only for RUR's of rational components. Below, we will show how to recover the exact coefficients of the RUR of a rational component of $\mathcal{I}$ from numerical approximations of the roots of $\mathcal{I}$. We start with giving heuristics for finding a good initial RUR which we will use as a starting point for iterative methods which are locally convergent to the exact RUR.

2.2. Initialization

As shown later, the iterative methods we use are *locally convergent*, so we need an initial RUR that is sufficiently close to the exact one in order to have convergence. In this subsection, we discuss heuristics to find a good initial approximate RUR that we can use as the starting point for our iteration.

More precisely, to compute an initial RUR for a rational component of $\mathcal{I}$ we propose the following:

1. **Homotopy method.** Let $\mathbf{f} = (f_1, \ldots, f_m) \in \mathbb{Q}[x_1, \ldots, x_n]$ for some $m > n$ be the defining equations of $\mathcal{I}$. As in [24, Section 3], for any linear map $R : \mathbb{Q}^m \to \mathbb{Q}^n$ which we will also consider as a matrix $R \in \mathbb{Q}^{n\times n}$, we define the well-constrained system $R(\mathbf{f}) := R \circ f$. For almost all choices of $R \in \mathbb{Q}^{n\times m}$, the ideal generated by $R(\mathbf{f})$ is zero dimensional and radical. We fix such an $R \in \mathbb{Q}^{n\times m}$ and throughout this paper we use the notation

$$\boldsymbol{F} = (F_1, \ldots, F_n) := R(\mathbf{f}). \tag{6}$$

   In this step, we assume that by using numerical homotopy algorithms (c.f. [4, 60]), we have computed *approximate roots* for each root in $V(\boldsymbol{F})$, i.e., local Newton's method with respect to $\boldsymbol{F}$ is quadratically convergent starting from these approximate roots (see [24, Section 2] on how to certify approximate roots of well-constrained systems). Using α-theory for $\boldsymbol{F}$, we can estimate the distance from each of these approximate roots to the exact ones.

   Denote an upper bound for these distances by ε.

2. **Candidates for roots of a component of $\mathcal{I}$.** To find the subset of approximate roots to $V(\boldsymbol{F})$ that approximates the roots in (a component of) $V(\mathcal{I})$, we propose several methods. The first one is to choose the roots that has residuals for all f_i for $i = 1, \ldots, m$ up to a given tolerance t. The tolerance t can be chosen based on ε defined above in Step 1, and the height and degree of each of the polynomials in $\mathbf{f}$. Another approach could incorporate the ideas in [24, Section 3] to exclude the roots that are *not* approximations of $V(\mathcal{I})$ by comparing the approximate roots of $R(\mathbf{f})$ to the approximate roots of an other random square subsystem $R'(\mathbf{f})$ for some $R' \in \mathbb{Q}^{n\times m}$. Finally, if we know that $\mathcal{I}$ or a rational component of $\mathcal{I}$ has only a very small number of common zeroes, then we can check all subsets of the roots

computed in Step 1 that have that cardinality as candidates for the common roots of a component of $\mathcal{I}$.

Denote the cardinality of the roots selected in this step by d.

3. **Initial RUR.** In this step, we compute an approximation to the RUR of a component of $\mathcal{I}$ from the d candidates selected in the previous step using (3), (4) and (5). In Section 2.3, we will discuss the sensitivity of this step to the perturbation of the roots, and give an upper bound for the distance of the approximate RUR computed from the exact one.

As mentioned in the Introduction, we propose two iterative methods for increasng the precision of the approximate RUR: in the first we use *Local Newton Iteration* and the second we use *Global Newton Iteration.* These are the subjects of the next two subsections.

2.3. Increasing the Precision of the RUR using Local Newton Iteration

The main idea of increasing the precision of the RUR by local Newton iteration is very simple: we repeat the computation of (3), (4), and (5) in the approximate RUR with ever more accurate roots that are computed by independently applying Newton's iteration to each of the d approximate roots.

Here, we analyze the accuracy that we can achieve for the approximate RUR after k iterations.

Let $\boldsymbol{F} = (F_1, \ldots, F_n)$ be as in (6). Let $\{\boldsymbol{z}_1^{(0)}, \ldots, \boldsymbol{z}_d^{(0)}\} \subset \mathbb{C}^n$ be the the d approximate roots we identified in Step 2 of Subsection 2.2, and let $\{\boldsymbol{z}_1^*, \ldots, \boldsymbol{z}_d^*\} \subset \mathbb{C}^n$ be the corresponding exact roots in $V(\boldsymbol{F})$. For each $i = 1, \ldots, d$ and $k \geq 0$ we define the $(k+1)$-th Newton iterate by

$$\boldsymbol{z}_i^{(k+1)} := \boldsymbol{z}_i^{(k)} - J_{\boldsymbol{F}}(\boldsymbol{z}_i^{(k)})^{-1} F(\boldsymbol{z}_i^{(k)}),$$

where $J_{\boldsymbol{F}}(\boldsymbol{z}_i^{(k)})$ is the $n \times n$ Jacobian matrix of $\boldsymbol{F}$ evaluated at $\boldsymbol{z}_i^{(k)}$, which we assume to be invertible. Then, using our assumption in Step 1 of Subsection 2.2, namely that for all $i = 1, \ldots, d$,

$$\|\boldsymbol{z}_i^{(0)} - \boldsymbol{z}_i^*\|_2 \leq \varepsilon,$$

and that the Newton iteration is quadratically convergent from each $\boldsymbol{z}_i^{(0)}$ to $\boldsymbol{z}_i^*$, we get that (using for example [8])

$$\|\boldsymbol{z}_i^{(k)} - \boldsymbol{z}_i^*\|_2 \leq \varepsilon \left(\frac{1}{2}\right)^{2^k - 1}.$$

Next, we analyze the possible loss of precision when applying (4) and (5) in the computation of the approximate RUR. Note that the Lagrange interpolation step in (5) may involve exponential loss of precision if we are not careful. In particular, the condition number of the Lagrange interpolant in the monomial basis is exponential in the number of nodes d even if the nodes μ_i are in the interval $[-1,1]$ (c.f. [18]). However, using an orthogonal polynomial basis such as Chebyshev polynomials, the condition number of the Lagrange interpolant in this basis is linear in d, assuming that the nodes μ_i are in the interval $[-1,1]$ (it is $\sqrt{2}d$ for Chebyshev polynomials, c.f. [18]). Choosing an appropriate primitive element $u = \lambda_1 x_1 + \cdots + \lambda_n x_n$ ensures that all nodes μ_i are in $[-1,1]$. Finally, we can convert the Chebyshev basis back to monomial bases by solving a triangular linear system with condition number at most $d2^{d-1}$ (c.f. [19, Lemma 2]).

Denote by $q^{(k)}(T)$ and $\mathbf{v}^{(k)}(T) = (v_1^{(k)}(T), \ldots, v_n^{(k)}(T))$ the approximate RUR corresponding to $\{\boldsymbol{z}_1^{(k)}, \ldots, \boldsymbol{z}_d^{(k)}\}$, and let $q^*(T)$ and $\mathbf{v}^*(T) = (v_1^*(T), \ldots, v_n^*(T))$ be the exact RUR corresponding to $\{\boldsymbol{z}_1^*, \ldots, \boldsymbol{z}_d^*\}$. Then, using the above argument, we get the following bound for the error of the coefficients of the interpolation polynomials:

$$\|v_j^{(k)}(T) - v_j^*(T)\|_2 \leq \varepsilon d^2 \left(\frac{1}{2}\right)^{2^k - 1 - d}, \tag{7}$$

which converges to zero as $k \to \infty$, since d and ε are fixed throughout our iteration. We can use the same bound as in the right hand side of (7) for the accuracy of the polynomial $q^{(k)}(T)$. Note that to get the above bound we assume that there is no roundoff error in our computations, only the approximation error from the roots.

2.4. Increasing the Precision of the RUR using Global Newton Method

In this section, we give an adaptation of the global Newton method (also called multivariate Hensel lifting or Newton-Hensel lifting) from the non-Archimedean metric defined by a p-adic valuation used, for example, in [29, 63, 66, 20] to the Euclidean metric defined by the usual absolute value on $\mathbb{Q}$ or $\mathbb{C}$. Global Newton method increases the accuracy of the approximate RUR directly, using polynomial arithmetics, without using approximations of the roots, as shown below.

The recent work [23] includes several versions of the global Newton method that are shown to be equivalent when the coefficient ring possesses a p-adic absolute value, but different when considered over a coefficient field such as $\mathbb{Q}$ with the usual absolute value. One of these versions of the global Newton method is equivalent to that obtained from the local Newton iteration described above. Moreover, the parallel complexity of the different versions of the global Newton iteration are compared in [23], and demonstrated that they can be efficiently parallelized.

In order to make this paper self contained, we recall a version of the global Newton method presented in [23] which is not equivalent to the one in the previous section, giving an alternative method to increase the accuracy. As we will see below, this version relates to some higher dimensional local Newton iteration. As a consequence, we prove the local quadratic convergence to the exact rational univariate representation with additional details provided in [23].

Given $\boldsymbol{F} = (F_1, \ldots, F_n)$ and $u = \sum_{i=1}^{n} \lambda_i x_i$ in $\mathbb{Q}[x_1, \ldots, x_n]$ as before, we define the map

$$\Phi : \mathbb{Q}^{(n+1)d} \to \mathbb{Q}^{(n+1)d}$$

as the map of the coefficient vectors of the following degree $d-1$ polynomials:

$$\Phi : \begin{bmatrix} v_1(T) \\ \vdots \\ v_n(T) \\ \Delta q(T) \end{bmatrix} \mapsto \begin{bmatrix} F_1(\mathbf{v}(T)) \bmod q(T) \\ \vdots \\ F_n(\mathbf{v}(T)) \bmod q(T) \\ \sum_{i=1}^{n} \lambda_i v_i(T) - T \end{bmatrix}, \tag{8}$$

where

$$\Delta q(T) := q(T) - T^d.$$

If $u, q(T), v_1(T), \ldots, v_n(T)$ is an exact RUR of a component of $\langle \boldsymbol{F} \rangle$ then clearly

$$\Phi \left(v_1(T), \ldots, v_n(T), \Delta q(T) \right) = 0.$$

We apply the $(n+1)d$ dimensional Newton iteration to locally converge to the coefficient vector of an exact RUR which is a zero of Φ. Note that below we will consider the map Φ as a map between

$$\Phi : \left(\mathbb{Q}[T]/\langle q(T) \rangle \right)^{n+1} \to \left(\mathbb{Q}[T]/\langle q(T) \rangle \right)^{n+1},$$

and that $\left(\mathbb{Q}[T]/\langle q(T) \rangle \right)^{n+1}$ and $\mathbb{Q}^{(n+1)d}$ are isomorphic as vectors spaces. As was shown in [23], the Newton iteration for Φ respects the algebra structure of $\left(\mathbb{Q}[T]/\langle q(T) \rangle \right)^{n+1}$ as well.

The first lemma gives the Jacobian matrix of Φ.

Lemma 2.3 ([23]). *Let $\boldsymbol{F} = (F_1, \dots, F_n)$, u, $q(T)$, $\mathbf{v}(T)$ and Φ be as above. For $i = 1, \dots, n$ define $m_i(T)$ and $r_i(T)$ as the quotient and remainder in the following division with remainder:*

$$F_i(\mathbf{v}(T)) = m_i(T)q(T) + r_i(T). \tag{9}$$

Then the Jacobian matrix of Φ defined in (8) respects the algebra structure of $(\mathbb{Q}[T]/\langle q(T)\rangle)^{n+1}$, and is given by

$$J_\Phi(\mathbf{v}(T), \Delta q(T)) := \begin{array}{|c|c|c}
\multicolumn{1}{c}{n} & \multicolumn{1}{c}{1} & \\
\hline
J_{\boldsymbol{F}}(\mathbf{v}(T)) & \begin{matrix} -m_1(T) \\ \vdots \\ -m_n(T) \end{matrix} & n \\
\hline
\lambda_1 \quad \cdots \quad \lambda_n & 0 & 1 \\
\hline
\end{array} \quad \mod q(T). \tag{10}$$

The definition below is given using polynomial arithmetic modulo $q(T)$, but as we will see below, it is equivalent to the Newton iteration corresponding to Φ as a map defined on $\mathbb{Q}^{(n+1)d}$. For the definition to be well defined we need the following assumptions:

Assumption 2.4. Let $\boldsymbol{F}$, $u = \sum_{i=1}^n \lambda_i x_i$, $q(T)$ and $\mathbf{v}(T)$ polynomials over $\mathbb{Q}$ as above. We assume that

1. $q(T)$ is monic and has degree d,
2. $v_i(T)$ has degree at most $d - 1$,
3. $\frac{\partial q(T)}{\partial T}$ is invertible modulo $q(T)$,
4. $\lambda_1 v_1(T) + \cdots + \lambda_n v_n(T) = T$,
5. $J_{\boldsymbol{F}}(\mathbf{v}(T)) := \left[\frac{\partial F_i}{\partial x_j}(\mathbf{v}(T))\right]_{i,j=1}^n$ is invertible modulo $q(T)$.
6. $J_\Phi := J_\Phi(\mathbf{v}(T), \Delta q(T))$ defined in (10) is invertible modulo $q(T)$.

Definition 2.5. *If $\boldsymbol{F}(x_1, \dots, x_n)$, $u(x_1, \dots, x_n)$, $q(T)$ and $\mathbf{v}(T)$ are polynomials over $\mathbb{Q}$ satisfying Assumption 2.4, we define*

$$\begin{aligned}
&u = \sum_{i=1}^n \lambda_i x_i = T, \\
&\boldsymbol{w}(T) := \mathbf{v}(T) - \left(J_{\boldsymbol{F}}(\mathbf{v}(T))^{-1}\boldsymbol{F}(\mathbf{v}(T)) \mod q(T)\right),
\end{aligned}$$

$$\Delta(T) := \sum_{i=1}^{n} \lambda_i w_i(T) - T, \tag{11}$$

$$\boldsymbol{r}(T) := \boldsymbol{F}(\mathbf{v}(T)) \mod q(T), \tag{12}$$

$$\boldsymbol{U}(T) := \frac{\partial \mathbf{v}(T)}{\partial T} - \left(J_{\boldsymbol{F}}(\mathbf{v}(T))^{-1} \frac{\partial \boldsymbol{r}(T)}{\partial T} \mod q(T) \right), \tag{13}$$

$$\Lambda(T) := \sum_{i=1}^{n} \lambda_i U_i(T) \text{ that we will show to be invertible} \mod q(T), \tag{14}$$

$$\boldsymbol{V}(T) := \boldsymbol{w}(T) - \left(\frac{\Delta(T)}{\Lambda(T)} \boldsymbol{U}(T) \mod q(T) \right), \tag{15}$$

$$Q(T) := q(T) - \left(\frac{\Delta(T)}{\Lambda(T)} \frac{\partial q(T)}{\partial T} \mod q(T) \right). \tag{16}$$

The next proposition shows that $\boldsymbol{V}(T)$ and $Q(T)$ from Definition 2.5 are the Newton iterates for the function Φ.

Proposition 2.6 ([23]). *Let $\boldsymbol{F}$, u, $q(T)$, $\mathbf{v}(T)$ and Φ be such that Assumption 2.4 holds. Then, $\Lambda(T)$ defined in (14) is invertible modulo $q(T)$, and thus $\boldsymbol{V}(T)$ and $Q(T)$ are well defined in Definition 2.5. Furthermore*

$$\begin{bmatrix} \boldsymbol{V}(T) \\ Q(T) - T^d \end{bmatrix} = \begin{bmatrix} \mathbf{v}(T) \\ q(T) - T^d \end{bmatrix} - J_\Phi^{-1} \cdot \begin{bmatrix} \boldsymbol{F}(\mathbf{v}(T)) \\ \sum_{i=1}^{n} \lambda_i v_i(T) - T \end{bmatrix} \mod q(T), \tag{17}$$

where the vector on the right hand side is $\Phi(\mathbf{v}(T), q(T) - T^d)$. Finally, we also have that

$$\sum_{i=1}^{n} \lambda_i V_i(T) = T.$$

As a corollary, we know that the approximate RUR defined by the iteration in Definition 2.5 is locally quadratically and converges to an exact RUR of a component of $\langle \boldsymbol{F} \rangle$.

Corollary 2.7. *The iteration defined by Definition 2.5 is locally quadratically convergent to an exact RUR of a component of $\langle \boldsymbol{F} \rangle$, as long as Assumption 2.4 is satisfied in each iteration.*

In the next section, we will need estimates on the accuracy of our iterates after k iterations of the global Newton method. Here, we can use α-theory applied to the function Φ as described in [24, Section 2] to certify that the initial RUR computed in Subsection 2.2 has the property of quadratic convergence and bound the distance to the exact RUR it converges to.

Let $Q^{(0)}(T)$ and $\boldsymbol{V}^{(0)}(T) = (V_1^{(0)}(T), \ldots, V_n^{(0)}(T))$ denote our initial approximate RUR, and assume that it quadratically converges to the exact zero $Q^*(T)$ and $\boldsymbol{V}^*(T) = (V_1^*(T), \ldots, V_n^*(T))$ of the map Φ. Note that this exact RUR may not be an RUR of a rational component of $\mathcal{I}$, i.e. it may not have rational coefficients. At this stage, we will not be able decide its rationality. Assume that we have a number ν such that

$$\|Q^{(0)}(T) - Q^*(T)\|_2 \leq \nu, \quad \|V_i^{(0)}(T) - V_i^*(T)\|_2 \leq \nu \quad i = 1, \ldots, n,$$

where the norm is the usual Euclidean norm of the coefficient vectors of the polynomials over $\mathbb{C}$. Then, using the quadratic convergence of the global Newton iteration, according to [8], we get the following bound for the error of the coefficients of the polynomials in the k-th iteration, denoted by $Q^{(k)}(T)$ and $\boldsymbol{V}^{(k)}(T) = (V_1^{(k)}(T), \ldots, V_n^{(k)}(T))$:

$$\|Q^{(k)}(T) - Q^*(T)\|_2 \leq \nu \left(\frac{1}{2}\right)^{2^k - 1}, \quad \|V_i^{(k)}(T) - V_i^*(T)\|_2 \leq \nu \left(\frac{1}{2}\right)^{2^k - 1} \tag{18}$$

which converge to zero as $k \to \infty$.

2.5. Rational Number Reconstruction

Below, we will show how to find the exact RUR of a rational component of $\mathcal{I}$ once we computed a sufficiently close approximation of it. The main idea is that we can reconstruct the unique rational numbers that have bounded denominators and indistinguishable from the coefficients of the polynomials in our approximate RUR within their accuracy estimates. Then, we can use purely symbolic methods to check whether the RUR with the bounded rational coefficients is an exact RUR for a component of our input system $\mathbf{f}$. Here, we recall the theory behind rational number reconstruction, and in the next section we detail our "end game," i.e. conditions for termination.

Since the coordinates of the approximate roots are given as floating point complex numbers, we can consider them as Gaussian rational in $\mathbb{Q}(i)$, and the same is true for the coefficients of the approximate RUR computed from these roots in Subsection 2.3. However, since the exact RUR has rational coefficients, we will neglect the imaginary part of the coefficients of the approximate RUR. Therefore, we will assume that the coefficients of the approximate RUR are in $\mathbb{Q}$, given as floating point numbers.

In this section, we use rational number reconstruction for each coefficient of the approximate RUR that was computed in the previous subsections. The following classical result implies that if a number is sufficiently close to a rational number with small denominator, then we can find this latter rational number in polynomial time (c.f. [58, Corrolary 6.3a] or [64, Theorem 5.26]).

Theorem 2.8 ([58, 64]). *There exists a polynomial time algorithm which, for a given rational number c and a natural number B tests if there exists a pair of integers (z, d) with $1 \leq d \leq B$ and*

$$|c - z/d| < \frac{1}{2B^2},$$

and if so, finds this unique pair of integers.

To compute the pair $(z, d) \in \mathbb{Z}^2$ for each coefficient c appearing in the approximate RUR computed in the previous subsections, we use the bound $B \in \mathbb{N}$ such that $\frac{1}{2B^2} \cong E$, where E denotes our estimate of the accuracy of our approximate RUR either from (7) or from (18). Thus, we can define

$$B := \left\lceil (2E)^{-1/2} \right\rceil.$$

For efficient computation of the rational number reconstruction, we can use the extended Euclidean algorithm or, equivalently, continued fractions (e.g., see [64, 65, 51, 43, 61]), or use LLL basis reduction as in [10]. The computed rational polynomials with bounded denominators we denote by

$$\hat{q}(T) \text{ and } \hat{\mathbf{v}}(T) = (\hat{v}_1(T), \ldots, \hat{v}_n(T)). \tag{19}$$

Remark 2.9. Theorem 2.8 does not guarantee existence of the the pair (z, d) with the given properties, only uniqueness. In case the rational number reconstruction algorithm for some coefficient c returns that there is no rational number within distance E with denominators at most B, we will need to improve the precision E (which will increase the bound B on the denominator). This is done by applying further local or global Newton steps on our approximates. As described in Theorem 2.12 below, if the bound B we obtained this way is larger than an *a priori* bound, we terminate our iteration and conclude that it did not converge to a RUR of a rational component.

2.6. Termination

One key task is to decide whether to terminate our iterations or increase the accuracy of the approximate RUR as described in Sections 2.3 and 2.4.

Let $\hat{q}(T)$ and $\hat{\mathbf{v}}(T) = (\hat{v}_1(T), \dots, \hat{v}_n(T))$ be the rational polynomials with bounded denominators as computed in Subsection 2.5. In this step we will use our original overdetermined input system $\mathbf{f} = (f_1, \dots, f_m)$ in $\mathbb{Q}[x_1, \dots, x_n]$ for some $m \geq n$ with $\mathcal{I} = \langle f_1, \dots, f_m \rangle$.

First, we reduce symbolically each f_i by the Gröbner basis

$$\{\hat{q}(T), x_1 - \hat{v}_1(T), \dots, x_n - \hat{v}_n(T)\},$$

or equivalently we compute $f_i(\hat{\mathbf{v}}(T)) \mod q(T)$. If they all reduce to zero, we return (19) as the exact RUR of $\mathcal{I}$.

If not all $f_i(\hat{\mathbf{v}}(T)) \mod q(T)$ are zero, then either the accuracy of our approximate RUR was too small to "click on" the exact RUR in Subsection 2.5, or the iteration does not converge to an RUR of a rational component of $\mathcal{I}$. To decide which case we are in, we will use *a priori* upper bounds on the heights of the coefficients of an RUR of a rational component of $\mathcal{I}$.

Below, we review some of the known upper bounds that we can use in our estimates. First, we define heights of polynomials over $\mathbb{Q}$ in a way that we can utilize symbolic algorithms over $\mathbb{Z}$ to get bounds.

Definition 2.10. *Let $p(T) = T^d + a_{d-1}T^{d-1} + \cdots + a_0 \in \mathbb{Q}[T]$ where each $a_i = z_i/d_i$ for $z_i \in \mathbb{Z}$ and $d_i \in \mathbb{N}$. Let $P(T) \in \mathbb{Z}[T]$ be any intereger polynomial that is an integer multiple of $p(T)$, for example we can choose*

$$P(T) = b_d T^d + b_{d-1}T^{d-1} + \cdots + b_0 := \left(\prod_{i=0}^{d-1} d_i\right) p(T) \in \mathbb{Z}[T].$$

Then the height of p *is defined as*

$$H(p) = H(P) = \frac{\max\{|b_i| \ : \ i = 0, \dots, d\}}{\gcd(b_i \ : \ i = 0, \dots, d)}$$

which is clearly independent of representation of $p(T)$ in $\mathbb{Z}[T]$. Finally, we define the logarithmic height of p *as*

$$h(p) := \log H(p).$$

Note that if $\gcd(z_i, d_i) = 1$ *for all* i *then*

$$H(p) \geq \max_i \{|z_i|, d_i\} \tag{20}$$

so we can use the height to bound the magnitude of the numerators and denominators appearing in the coefficients of our polynomials.

The best known upper bounds for the logarithmic heights of the polynomials in the RUR of $\mathcal{I}$ are as follows [14]. Assume that the input polynomials $f_1, \dots, f_m$ have degree at most D and logarithmic height at most h. First, we give a bound for the logarithmic bound of the primitive element $u = \lambda_1 x_1 + \cdots + \lambda_n x_n$ using [54, Lemma 2.1]:

$$h(\lambda_1 x_1 + \cdots + \lambda_n x_n) \leq 2(n-1)\log(n\delta),$$

where δ is the number of roots in $V(\mathcal{I})$. To use [14], which assumes that x_1 is a primitive element, we take $(f_1, \dots, f_m, \lambda_1 x_1 + \cdots + \lambda_n x_n - T)$ as our input, with a logarithmic height upper bound

$$h' := 2(n-1)\log(n\delta) + h.$$

Then, using an arithmetic Bézout theorem in [37, Lemma 2.7] and [14, Theorem 1], the logarithmic heights of the polynomials $q(T), v_1(T), \dots, v_n(T)$ in an RUR of $\mathcal{I}$ are bounded by

$$h(q), h(v_i) \leq 6n^3 h' D^{n+1} \leq 12 n^4 h D^{n+1} \log(n\delta) \quad i = 1, \dots n. \tag{21}$$

To get a bound for the height of the polynomials in an RUR for a rational component of $\mathcal{I}$, we use Gelfund's inequality for the height of a polynomial divisor of an integer polynomial [26, Proposition B.7.3] (other bounds can also be used, a survey of the known bounds of factors in $\mathbb{Z}[x]$ be found in [1]). For $P, Q \in \mathbb{Z}[T]$ such that P is a divisor of Q, we have

$$H(P) \leq e^{\deg(Q)} H(Q).$$

This, combined with (21), gives the following upper bound for the logarithmic heights of the polynomials in an exact RUR $q^*(T), v_1^*(T), \dots, v_n^*(T)$ of a rational component of $\mathcal{I}$:

$$h(q^*), h(v_i^*) \leq 12 \delta n^4 h D^{n+1} \log(n\delta) \quad i = 1, \dots n, \tag{22}$$

where δ is the number of roots in $V(\mathcal{I})$. This yields the following bound on the heights of the polynomials in a RUR of a rational component of $\mathcal{I}$:

$$H(q^*), H(v_i^*) \leq Hn\delta e^{12\delta n^4 D^{n+1}} \quad i = 1, \ldots n, \tag{23}$$

where $H = e^h$ is an upper bound for the height of the input polynomials in $\mathbf{f}$.

Once we have an *a priori* bound for the heights of the polynomials in an exact RUR of a rational component of $\mathcal{I}$, we can use (20) and check if the bound B for the denominators used in the rational number reconstruction in Subsection 2.5 exceeds the *a priori* bound from (23). If that is the case, we conclude that the iteration did not converge to an exact RUR of a rational component of $\mathcal{I}$ and terminate our algorithm. Otherwise, continue to increase the accuracy of our approximation.

We summarize this subsection in the following theorems:

Theorem 2.11. *Let $\mathcal{I}$ be as above. Assume that $q^*(T), v_1^*(T), \ldots, v_n^*(T)$ is an exact RUR of a rational component of $\mathcal{I}$. Define the maximum height*

$$H^* := max\{H(q^*), H(v_1^*), \ldots, H(v_n^*)\}.$$

Assume that an approximate RUR, $q(T), v_1(T), \ldots, v_n(T)$, satisfies

$$\|q(T) - q^*(T)\|_2, \|v_i(T) - v_i^*(T)\|_2 \leq E < \frac{1}{2(H^*)^2} \quad i = 1, \ldots, n, \tag{24}$$

for some $E > 0$, and let $\hat{q}(T), \hat{v}_1(T), \ldots, \hat{v}_n(T)$ obtained via rational number reconstruction on the coefficients of $q(T), v_1(T), \ldots, v_n(T)$ using the bound $B := \lceil (2E)^{-1/2} \rceil > H^*$. Then

$$\hat{q}(T) = q^*(T), \hat{v}_1(T) = v_1^*(T), \ldots, \hat{v}_n(T) = v_n^*(T).$$

Proof. Note that the coefficients of $q^*(T), v_1^*(T), \ldots, v_n^*(T)$ have denominator at most $H^* < B$ by (20). Since the 2-norm gives an upper bound for the infinity norm, we know all coefficients of $q^*(T), v_1^*(T), \ldots, v_n^*(T)$ are at most distance E from the corresponding coefficient of $q(T), v_1(T), \ldots, v_n(T)$. By Theorem 2.8, for each coefficient of of $q(T), v_1(T), \ldots, v_n(T)$, there is at most one rational number with denominator bounded by $B = \lceil (2E)^{-1/2} \rceil > H^*$ within the distance of

$$\frac{1}{2B^2} = \frac{1}{2\left\lceil (2E)^{-1/2} \right\rceil^2} \leq \frac{1}{2\left((2E)^{-1/2}\right)^2} = E.$$

This proves that the rational reconstruction must equal the exact RUR. □

The next theorem considers the converse statement.

Theorem 2.12. *Let $\mathbf{f} \subset \mathbb{Q}[x_1, \ldots, x_n]$ be as above and assume that H and D are the maximum height and degree of the polynomials in $\mathbf{f}$ respectively. Also, let δ be the cardinality of the common roots of $\mathbf{f}$ in $\mathbb{C}^n$. Assume that we have an upper bound E for the accuracy of our approximate RUR $q(T), v_1(T), \ldots, v_n(T)$ either from (7) or from (18). Let $B := \lceil (2E)^{-1/2} \rceil$ and assume that*

$$B \geq Hn\delta e^{12\delta n^4 D^{n+1}}.$$

Let $\hat{q}(T), \hat{v}_1(T), \ldots, \hat{v}_n(T)$ be obtained via rational number reconstruction from the coefficients of $q(T), v_1(T), \ldots, v_n(T)$ using the bound B for the denominators. If

$$\mathbf{f}(\hat{\mathbf{v}}(T)) \not\equiv 0 \mod \hat{q}(T)$$

then there is no exact RUR of a rational component of $\mathcal{I}$ within the distance of E from $q(T), v_1(T), \ldots, v_n(T)$.

Proof. If there was an exact RUR of a rational component of $\mathcal{I}$ within E from $q(T), v_1(T), \ldots, v_n(T)$, the heights of its coefficients would be bounded by $B \geq Hn\delta e^{12\delta n^4 D^{n+1}}$ as in (23). The rational number reconstruction algorithm would have found this exact RUR, which is a contradiction. $\square$

Finally, we give the number of iteration needed (asymptotically) in the "best case" and in the "worst case".

Theorem 2.13. *Let $\mathbf{f} \in \mathbb{Q}[x_1, \ldots, x_n]^m$ and $\mathcal{I} = \langle \mathbf{f} \rangle$ be as above.*

1. *Assume that $q^*(T), v_1^*(T), \ldots, v_n^*(T)$ is an exact RUR for a component of $\mathcal{I}$ and assume that $q^{(0)}(T), v_1^{(0)}(T), \ldots, v_n^{(0)}(T)$ is an initial approximate RUR which quadratically converges to $q^*(T), v_1^*(T), \ldots, v_n^*(T)$ either using local Newton iteration as in Subsection 2.3 or global Newton iteration as in Subsection 2.4. Then the number of iterations needed to find $q^*(T), v_1^*(T), \ldots, v_n^*(T)$ is asymptotically bounded by*

   $$\mathcal{O}(\log(d) \log\log(H^* E_0))$$

   where $H^* = max\{H(q^*), H(v_1^*), \ldots, H(v_n^*)\}$ is the height of the output, $d = \deg_T(q)$, and E_0 is an upper bound on the Euclidean distance of $q^{(0)}(T), \mathbf{v}^{(0)}(T)$ from the output $q^*(T), \mathbf{v}^*(T)$.
2. *Assume that $q^{(0)}(T), v_1^{(0)}(T), \ldots, v_n^{(0)}(T)$ is an initial approximate RUR which quadratically converges to the polynomials $q^*(T), v_1^*(T), \ldots, v_n^*(T)$,*

but these polynomials have irrational coefficients, i.e. they do not form an RUR of a rational component of $\mathcal{I}$. In this case we need

$$\mathcal{O}(n\log(d\delta nD)\log\log(HE_0))$$

iterations to conclude that our iteration did not converge to an exact RUR of a rational component of $\mathcal{I}$. Here H and D are the maximum of the heights and degrees of the polynomials in $\mathbf{f}$, respectively, δ is the number of roots in $V(\mathcal{I})$, and d, E_0 are as above in 1.

Proof. 1. By Theorem 2.11 to successfully terminate the algorithm with the exact RUR we need to achieve inaccuracy of $\frac{1}{2(H^*)^2}$. We will use the error bound of the local Newton method to increase the accuracy of the RUR given in (7) because that is a weaker bound than the one we got for the global Newton method in (18), so the bound we get for the number of iterations k in (7) will also give an upper bound for the iterations that we need using the global Newton method. Thus using (7) we need that

$$d^2E_0\left(\frac{1}{2}\right)^{2^k-1-d} \leq \frac{1}{2(H^*)^2},$$

which is satisfied if

$$k \geq c_1\log(d)\log\log(E_0H^*)$$

for some constant $c_1 \leq 2$.

2. By Theorem 2.12 to terminate the algorithm in the worst case we need to achieve accuracy

$$\frac{1}{2(Hn\delta e^{12\delta n^4D^{n+1}})^2}.$$

Using the same argument as for the first claim for the error bound after k iterations of either the local or the global Newton method, using (7) we need that

$$d^2E_0\left(\frac{1}{2}\right)^{2^k-1-d} \leq \frac{1}{2(Hn\delta e^{12\delta n^4D^{n+1}})^2},$$

which is satisfied if

$$k \geq c_2(n+1)\log(d\delta nD)\log\log(E_0H)$$

for some constant $c_2 \leq 18$. □

3. Certification of isosingular points

As summarized in the Introduction, we will use isosingular deflation [25] to reduce to the case of certifying simple roots to overdetermined systems which was discussed in Section 2. Due to this reduction, we can extend this approach to all points which can be regularized by isosingular deflation, called *isosingular points.* Since every isolated multiple root is an isosingular point, this method applies to multiple roots. However, isosingular points need not be isolated as demonstrated by the origin with respect to the Whitney umbrella $x^2 - y^2 z = 0$.

3.1. Isosingular deflation

Given a system $\mathbf{g} = (g_1, \ldots, g_m) \in \mathbb{Q}[x_1, \ldots, x_n]$ and a root z of $\mathbf{g} = 0$, isosingular deflation is an iterative regularization process. For the current purposes, we will only consider the deflation operator $\mathcal{D}_{\text{det}}$ defined as follows. Let $d = \text{dnull}(\mathbf{g}, \text{z}) = \dim\ \text{null}\ \mathbf{Jg}(\text{z})$ be the dimension of the null space of the Jacobian matrix $J\mathbf{g}$ evaluated at z. Let $\ell = \binom{n}{n-d+1} \cdot \binom{m}{n-d+1}$ and $\{\sigma_1, \ldots, \sigma_\ell\}$ be the index set of all $(n-d+1) \times (n-d+1)$ submatrices of an $m \times n$ matrix. If $d = \max\{0, n-m\}$, then $\ell = 0$ in which case we know that z is a smooth point on an irreducible solution component of dimension d. Define $\mathcal{D}_{\text{det}}(\mathbf{g}, z) = (\mathbf{g}_{\text{det}}, z_{\text{det}})$ where $z_{\text{det}} = z$ and

$$\mathbf{g}_{\text{det}} = \begin{bmatrix} \mathbf{g} \\ \det J_{\sigma_1}\mathbf{g} \\ \vdots \\ \det J_{\sigma_\ell}\mathbf{g} \end{bmatrix}.$$

The matrix $J_\sigma \mathbf{g}$ is the submatrix of the Jacobian $J\mathbf{g}$ indexed by σ. Clearly, $\mathbf{g}_{\text{det}}$ consists of polynomials in $\mathbb{Q}[x_1, \ldots, x_n]$.

Since the deflation operator $\mathcal{D}_{\text{det}}$ will be repeatedly applied, we write $\mathcal{D}^k_{\text{det}}(\mathbf{g}, z)$ to mean k successive iterations with $\mathcal{D}^0_{\text{det}}(\mathbf{g}, z) = (\mathbf{g}, z)$. This naturally leads to an associated nonincreasing sequence of nonnegative integers, called the *deflation sequence*, namely $\{d_r(\mathbf{g}, z)\}_{r=0}^{\infty}$ where

$$d_k(\mathbf{g}, z) = \text{dnull}(\mathcal{D}^{\text{k}}_{\text{det}}(\mathbf{g}, \text{z})) \quad \text{for } k \geq 0.$$

Since this nonnegative sequence is decreasing, it must have a limit and it must stabilize to that limit. That is, there are integers $d_\infty(\mathbf{g}, z) \geq 0$ and $s \geq 0$ so that $d_t(\mathbf{g}, z) = d_\infty(\mathbf{g}, z)$ for all $t \geq s$. When z is isolated, s is bounded above by the depth as well as multiplicity [15, 25, 39]. The limit $d_\infty(\mathbf{g}, z)$ is called the *isosingular local dimension* of z with respect to $\mathbf{g}$. The *isosingular points* are those for which their isosingular local dimension is zero so that, after finitely many iterations, isosingular deflation has regularized the root, i.e., constructed a polynomial system for which the point is a regular root. Clearly, such a system must consist of at least n polynomials, but will typically be overdetermined.

Since the resulting polynomial systems have rational coefficients, it immediately follows that every point in a zero-dimensional rational component must have the same deflation sequence. This can be used to partition the set of points under consideration into subsets and run the certification procedure described in Section 2 independently on each subset.

The construction of the deflation sequence and the resulting regularized system is an exact process that depends upon z. In our situation where z is only known approximately, we use the numerical approximations to compute exact numbers, namely the nonnegative integers arising as the dimensions of various linear subspaces which form the deflation sequence.

One drawback with $\mathcal{D}_{\text{det}}$ is the number of minors used in each iteration, namely $\ell = \binom{n}{n-d+1} \cdot \binom{m}{n-d+1}$. Since the codimension of the set of $m \times n$ matrices of rank $n-d$ is $c = d(m+d-n)$, we will adjust $\mathcal{D}_{\text{det}}$ to use exactly c minors as follows. Since $d = \text{dnull}(\mathbf{g}, \text{z})$, we can select an invertible $(n-d) \times (n-d)$ submatrix of $J\mathbf{g}(z)$. Rather than using all of $\{\sigma_1, \dots, \sigma_\ell\}$, we only use the c many which contain our selected invertible submatrix. In particular, with this setup, the tangent space of these c many minors is equal to the tangent space of all ℓ minors at z.

With this specialized construction, one now needs to be cautious that two points with the same deflation sequence can fail to be regularized by the system constructed by the other. However, all points in the same zero-dimensional rational component will still be regularized simultaneously. In particular, by comparing ranks of various submatrices, one may be able to produce a finer partition of the points under consideration before independently certifying each collection.

Example 3.1. As a demonstration, consider the Whitney umbrella defined by $\mathbf{g}(x, y, z) = x^2 - y^2 z$. Following [25, Ex. 5.12], the deflation sequence for the origin is $\{3, 2, 0, 0, \dots\}$ showing that the origin is not isolated but is an isosingular point. In particular, it takes two iterations to construct a polynomial system for which the origin is a regular root. Since $J\mathbf{g}(0)$ is identically zero, the first iteration appends all partial derivatives, say

$$\mathbf{g}'(x,y,z) = \begin{bmatrix} \mathbf{g}(x,y,z) \\ x \\ yz \\ y^2 \end{bmatrix}.$$

Since $J\mathbf{g}'(0)$ has rank 1, the original formulation of $\mathcal{D}_{\mathrm{det}}$ will append 18 2×2 minors of Jg'. However, with our modification, we only need to add the 6 minors which arise by submatrices that, in this case, include the unique nonzero element of $Jg'(0)$. For this example, it is easy to verify that the ideal of the resulting regularizing polynomial system is equal to $\langle x, y, z\rangle$.

3.2. Certification

Given $\mathbf{f} = (f_1, \ldots, f_m) \in \mathbb{Q}[x_1, \ldots, x_n]$ and a subset $V \subset V(\mathbf{f})$ consisting of isosingular points, the process for certification proceeds as follows.

1. **Deflation sequences.** Compute the deflation sequences for each of the points in V. If each point is an isosingular point, then isosingular deflation will terminate and produce a regularized system for each point. If one is not an isosingular point, one can apply the tests developed in [25, Section 6] to determine that the sequence has stabilized with the point having a positive isosingular local dimension. Remove all such points from V and partition the remaining points based on their deflation sequences and common regularizing polynomial systems, say $V_1, \ldots, V_k$.

2. **Certify each V_i.** Associated with each V_i is a polynomial system $\mathbf{f}^{(i)}$ having rational coefficients that must be either well-constrained or overdetermined. If it is well-constrained, simply apply standard α-theoretic techniques for certification. If overdetermined, use the approach presented in Section 2 for certification.

Successfully completing the certification proves that the points under consideration are indeed isosingular points of $\mathbf{f}$, i.e., the isosingular local dimension is zero. However, as currently formulated, this does not yield any information about the local dimension of the points, e.g., deciding if the point is isolated or not. Furthermore, even if one knows that a given point is isolated, this approach currently does not yield information about its multiplicity. Both of these are topics that will be addressed in future work.

4. Examples

4.1. An illustrative example

To demonstrate the approach, consider the polynomial system

$$\mathbf{g}(x_1, x_2, x_3) = \begin{bmatrix} x_1^2 + x_2^2 - 1 \\ 8x_1 - 16x_2^2 + 17 \\ x_1 - x_2^2 - x_3 - 1 \end{bmatrix}.$$

It is easy to verify that $\mathbf{g}$ has two roots of multiplicity 2. Thus, after appending $\det J\mathbf{g}$, we are interested in the overdetermined polynomial system

$$\mathbf{f}(x_1, x_2, x_3) = \begin{bmatrix} \mathbf{g}(x_1, x_2, x_3) \\ 64x_1x_2 + 16x_2 \end{bmatrix}$$

which has two regular roots. A randomization of $\mathbf{f}$ consists of 3 quadratics which has 4 regular solutions, two of which can be shown to not correspond to roots of $\mathbf{f}$ via [24, Section 3]. We start with the following numerical approximations for the $d = 2$ points of interest:

$$\boldsymbol{z}_1 = (-0.250, 0.968, -2.188) \text{ and } \boldsymbol{z}_2 = (-0.250, -0.968, -2.188)$$

with error bound $\varepsilon = 0.002$. From these numerical approximations, we see that we can take the primitive element to be $u = x_2$.

Using exact arithmetic, the initial RUR corresponding to this setup is

$$q(T) = T^2 - 14641/15625, \quad v_1(T) = -1/4, \quad v_2(T) = T, \quad v_3(T) = -547/250.$$

At $k = 0$ with $E = \varepsilon d^2 2^{1+d-2^k}$, we have $B = \lceil (2E)^{-1/2} \rceil = 2$. Since 1/4 can not be approximated by a rational number with denominator bounded by $B = 2$ with an error of at most $(2B^2)^{-1} = 1/8$, we perform a Newton iteration on each $\boldsymbol{z}_i$. At $k = 1$, using a denominator bound of $B = 16$ with error tolerance 1/512, we obtain

$$q(T) = T^2 - 15/16, \quad v_1(T) = -1/4, \quad v_2(T) = T, \quad v_3(T) = -35/16.$$

Since

$$\mathbf{f}(-1/4, T, -35/16) = \begin{bmatrix} T^2 - 15/16 \\ 2(T^2 - 15/16) \\ -(T^2 - 15/16) \\ 0 \end{bmatrix} = 0 \mod q(T)$$

we have proven that $\mathbf{f}$ and $\mathbf{g}$ have (at least) 2 roots which form a rational component. The corresponding well-constrained system from this RUR is

$$\begin{bmatrix} x_1 + 1/4 \\ x_2^2 - 15/16 \\ x_3 + 35/16 \end{bmatrix}.$$

4.2. A well-constrained system with regular and multiple roots

A common benchmark system is the Caprasse system which has 24 regular roots and 8 roots of multiplicity four [47], namely

$$\mathbf{g} = \begin{bmatrix} x_1^3x_3 - 4x_1^2x_2x_4 - 4x_1x_2^2x_3 - 2x_2^3x_4 - 4x_1^2 - 4x_1x_3 + 10x_2^2 + 10x_2x_4 - 2 \\ x_1x_3^3 - 4x_1x_3x_4^2 - 4x_2x_3^2x_4 - 2x_2x_4^3 - 4x_1x_3 + 10x_2x_4 - 4x_3^2 + 10x_4^2 - 2 \\ 2x_1x_2x_4 + x_2^2x_3 - 2x_1 - x_3 \\ x_1x_4^2 + 2x_2x_3x_4 - x_1 - 2x_3 \end{bmatrix}.$$

Since the system is well-constrained, numerical approximations for the 24 regular roots can be certified using standard α-theory. Here, we consider certifying the multiple roots. At each of these multiple roots, $J\mathbf{g}$ has rank 2 with the lower right 2×2 block having full rank. Thus, we consider the system $\mathbf{f}$ constructed by appending the four 3×3 minors of $J\mathbf{g}$ containing the lower right block to $\mathbf{g}$.

From the numerical approximations of the 8 points $\boldsymbol{z}_i$ that we computed using `Bertini` [3], we see that $u = x_1 - x_2 + 3x_3 - 3x_4$ is a primitive element. Starting the numerical approximations correct to 10 digits, we obtain the following RUR after one Newton iteration:

$$\begin{array}{rcl} q(T) & = & (T^2 + 4/3)(T^2 + 12)(T^2 - 16T + 76)(T^2 + 16T + 76) \\ v_1(T) & = & -(4107T^7 - 347060T^5 + 15954064T^3 + 361834048T)/339935232 \\ v_2(T) & = & -(6999T^7 - 672196T^5 + 32397008T^3 + 384342848T)/679870464 \\ v_3(T) & = & (1851T^7 - 169876T^5 + 8058512T^3 + 181018688T)/169967616 \\ v_4(T) & = & (6999T^7 - 672196T^5 + 32397008T^3 + 384342848T)/679870464 \end{array}$$

This RUR shows that the 8 roots arise from 4 rational components, each of degree 2, with splitting field $\mathbb{Q}[\sqrt{3}]$.

4.3. Two cyclic system

A common family of benchmark systems are the cyclic-n roots [7]. Here, we demonstrate using the approach for problems related to $n = 4$ and $n = 9$.

For $n \geq 2$, the cyclic-n system is

$$\mathbf{f}_n = \left[\begin{array}{cc} \sum_{j=1}^{n} \prod_{k=1}^{D} x_{j+k} & \text{for } D = 1, \ldots, n-1 \\ \prod_{k=1}^{n} x_k - 1 & \end{array} \right]$$

where $x_{n+\ell} = x_\ell$ for all $\ell = 1, \ldots, n$.

The solutions of $\mathbf{f}_4 = 0$ lie on two irreducible curves, with 8 embedded points, which are isosingular points. We deflate these points simultaneously by appending in the four 3×3 minors of $J\mathbf{f}_4$ containing the first and last rows, and second and third columns. By using numerical approximations computed by `Bertini`, we see that we can use the primitive element $u = x_1 + 2x_2 - x_3 + 3x_4$. This yields the RUR:

$$\begin{array}{rcl} q(T) & = & (T-1)(T+1)(T-3)(T+3)(T^2+1)(T^2+9) \\ v_1(T) & = & (-T^5 + 121T)/120 \\ v_2(T) & = & (-T^5 + 61T)/60 \\ v_3(T) & = & (T^5 - 121T)/120 \\ v_4(T) & = & (T^5 - 61T)/60 \end{array}$$

Now, for $n = 9$, we consider the overdetermined system

$$\mathbf{f}(x) = \left[\begin{array}{c} \mathbf{f}_9(x) \\ x_1 - x_4 \\ x_1 - x_7 \end{array} \right]$$

motivated by Example 9 of [15, Section 7]. The degree of the ideal generated by $\mathbf{f}$ is 162 with `Bertini` computing 54 regular points and 54 double points. The following uses the primitive element $u = x_1 + 2x_2 - x_3 + 2x_5 + x_6 - x_8$.

For the 54 regular points, we compute

$$\begin{array}{rcl} q(T) & = & (T^2 + T - 101)(T^4 - T^3 + 102T^2 + 101T + 10201) \\ & & (T^2 + 19T + 79)(T^4 - 19T^3 + 282T^2 - 1501T + 6241) \\ & & (T^2 - 17T + 61)(T^4 + 17T^3 + 228T^2 + 1037T + 3721) \\ & & (T^{12} - 2356T^9 + 5057697T^6 - 1161599884T^3 + 243087455521) \\ & & (T^{12} - 304T^9 + 1122717T^6 + 313211504T^3 + 1061520150601) \\ & & (T^{12} + 1802T^9 + 3020223T^6 + 409019762T^3 + 51520374361). \end{array}$$

For the 54 double points, by comparing ranks of 8×8 submatrices of $J\mathbf{f}$, we are able to partition into 3 subsets of size 18. For each of these subsets, we compute the three as

$$\begin{aligned} q(T) &= (T^2-11T+19)(T^4+11T^3+102T^2+209T+361)\\ &\quad (T^{12}+704T^9+488757T^6+4828736T^3+47045881) \end{aligned}$$

$$\begin{aligned} q(T) &= (T^2+13T+31)(T^4-13T^3+138T^2-403T+961)\\ &\quad (T^{12}-988T^9+946353T^6-29433508T^3+887503681) \end{aligned}$$

$$\begin{aligned} q(T) &= (T^2+T-11)(T^4-T^3+12T^2+11T+121)\\ &\quad (T^{12}-34T^9+2487T^6+45254T^3+1771561). \end{aligned}$$

Acknowledgements

We would like to thank Eric Schost and Teresa Krick for helpful discussions on *a priori* upper bounds for the heights of RUR's, as well as Hoon Hong for his helpful suggestions.

References

[1] J. Abbott. Bounds on factors in $\mathbb{Z}[x]$. *J. Symbolic Comput.*, 50:532–563, 2013.

[2] X. Allamigeon, S. Gaubert, V. Magron, and B. Werner. Formal proofs for nonlinear optimization. arXiv:1404.7282, 2014.

[3] D.J. Bates, J.D. Hauenstein, A.J. Sommese, and C.W. Wampler. Bertini: Software for numerical algebraic geometry. Available at bertini.nd.edu.

[4] D.J. Bates, J.D. Hauenstein, A.J. Sommese, and C.W. Wampler. *Numerically solving polynomial systems with Bertini*, volume 25 of *Software, Environments, and Tools.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2013.

[5] C. Beltrán and A. Leykin. Certified numerical homotopy tracking. *Exp. Math.*, 21(1):69–83, 2012.

[6] C. Beltrán and A. Leykin. Robust certified numerical homotopy tracking. *Found. Comput. Math.*, 13(2):253–295, 2013.

[7] G. Björck. Functions of modulus 1 on Z_n whose Fourier transforms have constant modulus, and "cyclic n-roots". In *Recent advances in Fourier analysis and its applications (Il Ciocco, 1989)*, volume 315 of *NATO Adv. Sci. Inst. Ser. C Math. Phys. Sci.*, pages 131–140. Kluwer Acad. Publ., Dordrecht, 1990.

[8] L. Blum, F. Cucker, M. Shub, and S. Smale. *Complexity and real computation.* Springer-Verlag, New York, 1998.

[9] S. Bozóki, T.-L. Lee, and L. Rónyai. Seven mutually touching infinite cylinders. `arXiv`:1308.5164, 2013.

[10] C. Bright and A. Storjohann. Vector rational number reconstruction. In *ISSAC 2011—Proceedings of the 36th International Symposium on Symbolic and Algebraic Computation*, pages 51–57. ACM, New York, 2011.

[11] W.D. Brownawell and C.K. Yap. Lower bounds for zero-dimensional projections. In *ISSAC 2009—Proceedings of the 2009 International Symposium on Symbolic and Algebraic Computation*, pages 79–85. ACM, New York, 2009.

[12] J. Canny. Generalised characteristic polynomials. *J. Symbolic Comput.*, 9(3):241–250, 1990.

[13] D. Castro, L.M. Pardo, K. Hägele, and J.E. Morais. Kronecker's and Newton's approaches to solving: a first comparison. *J. Complexity*, 17(1):212–303, 2001.

[14] X. Dahan and É. Schost. Sharp estimates for triangular sets. In *ISSAC 2004*, pages 103–110. ACM, New York, 2004.

[15] B.H. Dayton and Z. Zeng. Computing the multiplicity structure in solving polynomial systems. In *ISSAC'05*, pages 116–123. ACM, New York, 2005.

[16] J.P. Dedieu and M. Shub. Newton's method for overdetermined systems of equations. *Math. Comp.*, 69(231):1099–1115, 2000.

[17] H.R.P. Ferguson, D.H. Bailey, and S. Arno. Analysis of PSLQ, an integer relation finding algorithm. *Math. Comp.*, 68(225):351–369, 1999.

[18] W. Gautschi. Questions of numerical condition related to polynomials. In *Studies in Numerical Analysis*, volume 24 of *MAA Stud. Math.*, pages 140–177. Math. Assoc. America, Washington, DC, 1984.

[19] M. Giesbrecht, G. Labahn, and W. Lee. Symbolic-numeric sparse polynomial interpolation in chebyshev basis and trigonometric interpolation. In *Proceedings of Computer Algebra in Scientific Computing (CASC 2004)*, 2004.

[20] M. Giusti, G. Lecerf, and B. Salvy. A Gröbner free alternative for polynomial system solving. *J. Complexity*, 17(1):154–211, 2001.

[21] A. Griewank and M.R. Osborne. Analysis of Newton's method at irregular singularities. *SIAM J. Numer. Anal.*, 20(4):747–773, 1983.

[22] J.D. Hauenstein, I. Haywood, and A.C. Liddell, Jr. An a posteriori certification algorithm for newton homotopies. In *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*, ISSAC '14, pages 248–255, New York, NY, USA, 2014. ACM.

[23] J.D. Hauenstein, V.Y. Pan, and A. Szanto. Global Newton iteration over archimedian and non-archimedian fields. In *Proceedings of Computer Algebra in Scientific Computing (CASC 2014)*, Lecture Notes in Computer Science. Springer-Verlag, 2014.

[24] J.D. Hauenstein and F. Sottile. Algorithm 921: alphaCertified: certifying solutions to polynomial systems. *ACM Trans. Math. Software*, 38(4):28, 2012.

[25] J.D. Hauenstein and C.W. Wampler. Isosingular sets and deflation. *Found. Comput. Math.*, 13(3):371–403, 2013.

[26] M. Hindry and J.H. Silverman. *Diophantine geometry*, volume 201 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 2000.

[27] G. Jeronimo and D. Perrucci. On the minimum of a positive polynomial over the standard simplex. *J. Symbolic Comput.*, 45(4):434–442, 2010.

[28] G. Jeronimo, D. Perrucci, and E. Tsigaridas. On the minimum of a polynomial function on a basic closed semialgebraic set and applications. *SIAM J. Optim.*, 23(1):241–255, 2013.

[29] E. Kaltofen. Polynomial-time reductions from multivariate to bi- and univariate integral polynomial factorization. *SIAM J. Comput.*, 14(2):469–489, 1985.

[30] E. Kaltofen. Sparse Hensel lifting. In *EUROCAL '85, Vol. 2 (Linz, 1985)*, volume 204 of *Lecture Notes in Comput. Sci.*, pages 4–17. Springer, Berlin, 1985.

[31] E. Kaltofen, B. Li, Z. Yang, and L. Zhi. Exact certification of global optimality of approximate factorizations via rationalizing sums-of-squares with floating point scalars. In *Proceedings of the Twenty-first International Symposium on Symbolic and Algebraic Computation*, ISSAC '08, pages 155–164, New York, NY, USA, 2008. ACM.

[32] E.L. Kaltofen, B. Li, Z. Yang, and L. Zhi. Exact certification in global polynomial optimization via sums-of-squares of rational functions with rational coefficients. *J. Symb. Comput.*, 47(1):1–15, 2012.

[33] R. Kannan, A.K. Lenstra, and L. Lovász. Polynomial factorization and nonrandomness of bits of algebraic and some transcendental numbers. *Math. Comp.*, 50(181):235–250, 1988.

[34] Y. Kanzawa, M. Kashiwagi, and S. Oishi. An algorithm for finding all solutions of parameter-dependent nonlinear equations with guaranteed accuracy. *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, 82(10):33–39, 1999.

[35] Y. Kanzawa and S. Oishi. Approximate singular solutions of nonlinear equations and a numerical method of proving their existence. *Sūrikaisekikenkyūsho Kōkyūroku*, (990):216–223, 1997. Theory and application of numerical calculation in science and technology, II (Japanese) (Kyoto, 1996).

[36] R. Krawczyk. Newton-algorithmen zur bestimmung von nullstellen mit fehlerschranken. *Computing*, 4(3):187–201, 1969.

[37] T. Krick, L.M. Pardo, and M. Sombra. Sharp estimates for the arithmetic Nullstellensatz. *Duke Math. J.*, 109(3):521–598, 2001.

[38] J. Lasserre, M. Laurent, and P. Rostalskl. A unified approach to computing real and complex zeros of zero-dimensional ideals. In M. Putinar and S. Sullivant, editors, *Emerging Applications of Algebraic Geometry*, volume 149 of *The IMA Volumes in Mathematics and its Applications*, pages 125–155. Springer New York, 2009.

[39] A. Leykin, J. Verschelde, and A. Zhao. Newton's method with deflation for isolated singularities of polynomial systems. *Theoret. Comput. Sci.*, 359(1-3):111–122, 2006.

[40] A. Leykin, J. Verschelde, and A. Zhao. Higher-order deflation for polynomial systems with isolated singular solutions. In *Algorithms in al-*

gebraic geometry, volume 146 of *IMA Vol. Math. Appl.*, pages 79–97. Springer, New York, 2008.

[41] N. Li and L. Zhi. Verified error bounds for isolated singular solutions of polynomial systems: Case of breadth one. *Theor. Comput. Sci.*, 479:163–173, 2013.

[42] N. Li and L. Zhi. Verified error bounds for isolated singular solutions of polynomial systems. *SIAM Journal on Numerical Analysis*, 52(4):1623–1640, 2014.

[43] D. Lichtblau. Half-GCD and fast rational recovery. In *ISSAC'05*, pages 231–236. ACM, New York, 2005.

[44] A. Mantzaflaris and B. Mourrain. Deflation and certified isolation of singular zeros of polynomial systems. In *Proceedings of the 36th International Symposium on Symbolic and Algebraic Computation*, ISSAC '11, pages 249–256, New York, NY, USA, 2011. ACM.

[45] D. Monniaux and P. Corbineau. On the generation of positivstellensatz witnesses in degenerate cases. In M. van Eekelen, H. Geuvers, J. Schmaltz, and F. Wiedijk, editors, *Interactive Theorem Proving*, volume 6898 of *Lecture Notes in Computer Science*, pages 249–264. Springer Berlin Heidelberg, 2011.

[46] R. Moore. A test for existence of solutions to nonlinear systems. *SIAM Journal on Numerical Analysis*, 14(4):611–615, 1977.

[47] S. Moritsugu and K. Kuriyama. On multiple zeros of systems of algebraic equations. In *Proceedings of the 1999 International Symposium on Symbolic and Algebraic Computation*, ISSAC '99, pages 23–30, New York, NY, USA, 1999. ACM.

[48] Y. Nakaya, S. Oishi, M. Kashiwagi, and Y. Kanzawa. Numerical verification of nonexistence of solutions for separable nonlinear equations and its application to all solutions algorithm. *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, 86(5):45–53, 2003.

[49] T. Ojika. Modified deflation algorithm for the solution of singular problems. I. A system of nonlinear algebraic equations. *J. Math. Anal. Appl.*, 123(1):199–221, 1987.

[50] T. Ojika, S. Watanabe, and T. Mitsui. Deflation algorithm for the multiple roots of a system of nonlinear equations. *J. Math. Anal. Appl.*, 96(2):463–479, 1983.

[51] V.Y. Pan and X. Wang. On rational number reconstruction and approximation. *SIAM J. Comput.*, 33(2):502–503, 2004.

[52] H. Peyrl and P.A. Parrilo. A Macaulay2 package for computing sum of squares decompositions of polynomials with rational coefficients. In *SNC*, pages 207–208, 2007.

[53] H. Peyrl and P.A. Parrilo. Computing sum of squares decompositions with rational coefficients. *Theor. Comput. Sci.*, 409(2):269–281, Dec. 2008.

[54] F. Rouillier. Solving zero-dimensional systems through the rational univariate representation. *Journal of Applicable Algebra in Engineering, Communication and Computing*, 9(5):433–461, 1999.

[55] S. Rump and S. Graillat. Verified error bounds for multiple roots of systems of nonlinear equations. *Numerical Algorithms*, 54(3):359–377, 2010.

[56] S.M. Rump. Solving algebraic problems with high accuracy. In *Proc. Of the Symposium on A New Approach to Scientific Computation*, pages 51–120, San Diego, CA, USA, 1983. Academic Press Professional, Inc.

[57] M. Safey El Din and L. Zhi. Computing rational points in convex semialgebraic sets and sum of squares decompositions. *SIAM J. on Optimization*, 20(6):2876–2889, 2010.

[58] A. Schrijver. *Theory of linear and integer programming.* Wiley-Interscience Series in Discrete Mathematics. John Wiley & Sons Ltd., Chichester, 1986. A Wiley-Interscience Publication.

[59] S. Smale. Newton's method estimates from data at one point. In *The merging of disciplines: new directions in pure, applied, and computational mathematics (Laramie, Wyo., 1985)*, pages 185–196. Springer, New York, 1986.

[60] A.J. Sommese and C.W. Wampler, II. *The numerical solution of systems of polynomials.* World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2005.

[61] D.E. Steffy. Exact solutions to linear systems of equations using output sensitive lifting. *ACM Communications in Computer Algebra*, 44(4):160–182, Dec. 2010.

[62] A. Szanto. Solving over-determined systems by the subresultant method. *J. Symbolic Comput.*, 43(1):46–74, 2008. With an appendix by Marc Chardin.

[63] W. Trinks. On improving approximate results of Buchberger's algorithm by Newton's method. In B. Caviness, editor, *EUROCAL '85*, volume 204 of *Lecture Notes in Computer Science*, pages 608–612. Springer Berlin Heidelberg, 1985.

[64] J. von zur Gathen and J. Gerhard. *Modern computer algebra.* Cambridge University Press, New York, 1999.

[65] X. Wang and V.Y. Pan. Acceleration of Euclidean algorithm and rational number reconstruction. *SIAM J. Comput.*, 32(2):548–556, 2003.

[66] F. Winkler. A p-adic approach to the computation of Gröbner bases. *J. Symbolic Comput.*, 6(2-3):287–304, 1988.

[67] K. Yamamura, H. Kawata, and A. Tokue. Interval solution of nonlinear equations using linear programming. *BIT Numerical Mathematics*, 38(1):186–199, 1998.

[68] Z. Yang, L. Zhi, and Y. Zhu. Verified error bounds for real solutions of positive-dimensional polynomial systems. In *Proceedings of the 38th International Symposium on International Symposium on Symbolic and Algebraic Computation*, ISSAC '13, pages 371–378, New York, NY, USA, 2013. ACM.