Verified Error Bounds for Isolated Singular Solutions of Polynomial Systems: Case of Breadth One[☆]

Nan Li[a,b], Lihong Zhi[a]

[a] *KLMM, Academy of Mathematics and Systems Science, CAS, 100190 Beijing, China*
[b] *INRIA Paris-Rocquencourt, 78153 Le Chesnay Cedex, France*

Abstract

In this paper we describe how to improve the performance of the symbolic-numeric method in [19, 20] for computing the multiplicity structure and refining approximate isolated singular solutions in the breadth-one case. By introducing a parameterized deflated system with smoothing parameters, we generalize the algorithm in [33] to compute verified error bounds such that a slightly perturbed polynomial system is guaranteed to have a breadth-one multiple root within the computed bounds.

Keywords: polynomial systems, isolated singular solutions, multiplicity structure, verification, error bounds.

1. Introduction

It is a challenge problem to solve polynomial systems with singular solutions. In [28], Rall studied the convergence properties of Newton's method at singular solutions, and many modifications of Newton's method to restore the quadratic convergence have been proposed in [29, 30, 5, 6, 7, 10, 12, 27, 25, 11, 26, 1, 34]. Recently, some symbolic-numeric methods have been proposed for refining approximate isolated singular solutions to high accuracy [2, 14, 4, 8, 9, 15, 17, 36, 37, 3, 21]. Especially, in [15, 17], they modified the symbolic deflation method in [27, 26] and provided a numerically stable implementation to compute isolated singular solutions accurately to the full machine precision. For many benchmark problems from the PHCpack demos by Jan Verschelde (available at `http://www.math.uic.edu/~jan/demo.html`), one deflation always suffices to restore the quadratic convergence. However, for the special case of breadth one,

[☆]This research is supported by a NKBRPC 2011CB302400 and the Chinese National Natural Science Foundation under Grants: 91118001, 60821002/F02, 60911130369, 10871194 and the Sino-French Lab for Computer Science, Automation and Applied Mathematics LIAMA through the ECCA project. Some results of this paper have been presented at the Symbolic and Numerical Computation (SNC 2011) conference held June 7-9, 2011 in San Jose, California.

Email addresses: `linan08@amss.ac.cn` (Nan Li), `lzhi@mmrc.iss.ac.cn` (Lihong Zhi)

each deflation only reduces the multiplicity exactly by one [20]. In addition, a modified deflation method based on computing the multiplicity structure was proposed in [4] to reduce the size of deflated systems in [15, 17] to a linear function in the multiplicity for the case of breadth one. Furthermore, in [19, 20], we described a new symbolic-numeric method based on regularized Newton iterations and computations of multiplicity structures for refining approximate isolated singular solutions with quadratic convergence in the breadth-one case. The size of matrices involved in our algorithm is bounded by the number of variables and polynomials and does not increase with the multiplicity. A preliminary implementation performs well in most cases. However, it may suffer from computing and storing dense multiplicity structures caused by linear transformations or dense expressions of differential functionals even for sparse input polynomials. In [18], we show briefly how to compute evaluations of differential functionals more efficiently by avoiding linear transformations and solving a sequence of least squares problems.

Since arbitrary small perturbations of coefficients may transform an isolated singular solution into a cluster of simple roots (or even make it disappear), it is more difficult to certify whether a polynomial system has a multiple root. In [33], by introducing a smoothing parameter, they described a verification method for computing guaranteed error bounds such that a slightly perturbed system is proved to have a double root within the computed bounds. In [21], they proposed a method to verify a multiple root of a nearby system with a computed local structure, which depends on accuracy of the given approximate singular point. By adding a perturbed univariate polynomial in one selected variable with some smoothing parameters to one selected equation of the input system, we are able to use the parameterized deflated system in [18] to compute guaranteed error bounds, such that a slightly perturbed system has a breadth-one multiple root within the computed bounds. Moreover, if the original input system has an exact breadth-one multiple root near the given approximate singular solution, we can refine the multiple root and the multiplicity structure simultaneously to arbitrary accuracy. The techniques for the construction of a parameterized deflated system and evaluations of differential functionals are similar to those introduced in [15, 16].

Main results. In this paper, we still focus on the special case where the Jacobian matrix has corank one. First, we describe a method to preserve sparse structures of input polynomial systems by avoiding linear transformations. Next, we present an algorithm to compute the recursive evaluation of differential functionals without constructing and storing their dense structure. Furthermore, we show that the parameterized deflated system introduced in [18] can be used to generalize the algorithm in [33] to compute verified error bounds, therefore, a slightly perturbed polynomial system is guaranteed to have a breadth-one multiple root within the computed bounds. We also prove that it is always possible to construct a regular augmented system to compute an inclusion of the multiple root by choosing properly smoothing parameters and renumbering the polynomials. We provide numerical experiments to demon-

strate the effectiveness of our method.

Organization of the paper. Section 2 is devoted to recall some notations and well-known facts. In Section 3, we describe an improved algorithm for computing the multiplicity structure at an isolated singular solution when the Jacobian matrix has corank one. Some experimental results are given to show the efficiency of the new algorithm. In Section 4, we show how to construct a parameterized deflated system to refine and compute verified error bounds for a breadth-one multiple root. Some numerical examples are given to demonstrate the performance of our algorithm.

2. Notation and Preliminaries

Let $R = \mathbb{K}[\mathbf{x}]$ denote a polynomial ring over the field $\mathbb{K}$ of characteristic zero. Let $I = (f_1, \ldots, f_n)$ be an ideal of R, $\hat{\mathbf{x}} \in \mathbb{K}^n$ an isolated root of I, $m_{\hat{\mathbf{x}}} = (x_1 - \hat{x}_1, \ldots, x_n - \hat{x}_n)$ the maximal ideal at $\hat{\mathbf{x}}$. Suppose $Q_{\hat{\mathbf{x}}}$ is the isolated primary component whose associate prime is $m_{\hat{\mathbf{x}}}$, then the multiplicity μ of $\hat{\mathbf{x}}$ is defined as the dimension of the quotient ring $R/Q_{\hat{\mathbf{x}}}$.

Let $\mathbf{d}^{\alpha}_{\hat{\mathbf{x}}} : R \to \mathbb{K}$ denote the differential functional defined by

$$\mathbf{d}^{\alpha}_{\hat{\mathbf{x}}}(g) = \frac{1}{\alpha_1! \cdots \alpha_n!} \cdot \frac{\partial^{|\alpha|} g}{\partial x_1^{\alpha_1} \cdots \partial x_n^{\alpha_n}}(\hat{\mathbf{x}}), \quad \forall g(\mathbf{x}) \in R, \tag{1}$$

for a point $\hat{\mathbf{x}} \in \mathbb{K}^n$ and an array $\alpha \in \mathbb{N}^n$. The normalized differentials have a useful property: when $\hat{\mathbf{x}} = \mathbf{0}$, we have $\mathbf{d}^{\alpha}_{\mathbf{0}}(\mathbf{x}^{\beta}) = 1$ if $\alpha = \beta$ or 0 otherwise. We may occasionally write $\mathbf{d}^{\alpha} = d_1^{\alpha_1} d_2^{\alpha_2} \cdots d_n^{\alpha_n}$ instead of $\mathbf{d}^{\alpha}_{\hat{\mathbf{x}}}$ for simplicity if $\hat{\mathbf{x}}$ is clear from the context, where $d_i^{\alpha_i} = \frac{1}{\alpha_i!} \frac{\partial^{\alpha_i}}{\partial x_i^{\alpha_i}}$.

Definition 2.1. *The local dual space of I at $\hat{\mathbf{x}}$ is the subspace of elements of $\mathfrak{D}_{\hat{\mathbf{x}}} = \mathrm{Span}_{\mathbb{K}}\{\mathbf{d}^{\alpha}_{\hat{\mathbf{x}}}, \alpha \in \mathbb{N}^n\}$ that vanish on all the elements of I*

$$\mathcal{D}_{\hat{\mathbf{x}}} := \{\Lambda \in \mathfrak{D}_{\hat{\mathbf{x}}} \mid \Lambda(f) = 0, \ \forall f \in I\}, \tag{2}$$

where $\dim(\mathcal{D}_{\hat{\mathbf{x}}}) = \mu$.

Computing a closed basis of the local dual space is done essentially by matrix-kernel computations [22, 24, 4, 36, 38], which are based on the *stability property* of $\mathcal{D}_{\hat{\mathbf{x}}}$:

$$\forall \Lambda \in \mathcal{D}^t_{\hat{\mathbf{x}}}, \ \Phi_{x_i}(\Lambda) \in \mathcal{D}^{t-1}_{\hat{\mathbf{x}}}, \ i = 1, \ldots, n, \tag{3}$$

where $\mathcal{D}^t_{\hat{\mathbf{x}}}$ denotes the subspace of $\mathcal{D}_{\hat{\mathbf{x}}}$ of the degree less than or equal to t, for $t \in \mathbb{N}$, and $\Phi_{x_i} : \mathfrak{D}_{\hat{\mathbf{x}}} \to \mathfrak{D}_{\hat{\mathbf{x}}}$ are the linear *anti-differentiation operators* defined by

$$\Phi_{x_i}(\mathbf{d}^{\alpha}_{\hat{\mathbf{x}}}) := \begin{cases} \mathbf{d}^{(\alpha_1, \ldots, \alpha_i - 1, \ldots, \alpha_n)}_{\hat{\mathbf{x}}}, & \text{if } \alpha_i > 0, \\ 0, & \text{otherwise.} \end{cases}$$

Moreover, $\mathcal{D}^0_{\hat{\mathbf{x}}} = \mathrm{Span}_{\mathbb{K}}\{\Lambda_0 := 1\}$ and $\Phi^k_{x_i}(\mathbf{d}^{\alpha}_{\hat{\mathbf{x}}}) := \Phi_{x_i} \overbrace{\circ \cdots \circ}^{k-1} \Phi_{x_i}(\mathbf{d}^{\alpha}_{\hat{\mathbf{x}}})$.

Lemma 2.2. *[35, Theorem 8.36] Suppose* $\{\Lambda_0, \ldots, \Lambda_s\}$ *is a closed basis of* $\mathcal{D}^{t-1}_{\hat{\mathbf{x}}}$*, then an element* $\Lambda \in \mathfrak{D}_{\hat{\mathbf{x}}}$ *lies in* $\mathcal{D}^{t}_{\hat{\mathbf{x}}}$ *if and only if it satisfies (3) and* $\Lambda(f_i) = 0$ *for* $i = 1, \ldots, n$.

In fact, (3) is equivalent to finding $\lambda_{i,k} \in \mathbb{K}$ such that $\Lambda \in \mathfrak{D}_{\hat{\mathbf{x}}}$ satisfies

$$\Phi_{x_i}(\Lambda) = \lambda_{i,0}\Lambda_0 + \lambda_{i,1}\Lambda_1 + \cdots + \lambda_{i,s}\Lambda_s, \text{ for } i = 1, \ldots, n. \tag{4}$$

If these $\lambda_{i,j}$ are known, we can compute Λ by the following formula [24]

$$\Lambda = \sum_{j=0}^{s} \lambda_{1,j}\Psi_{x_1}(\Lambda_j) + \sum_{j=0}^{s} \lambda_{2,j}\Psi_{x_2}(\Lambda_j) + \cdots + \sum_{j=0}^{s} \lambda_{n,j}\Psi_{x_n}(\Lambda_j), \tag{5}$$

where the *differentiation operators* $\Psi_{x_i} : \mathfrak{D}_{\hat{\mathbf{x}}} \to \mathfrak{D}_{\hat{\mathbf{x}}}$ are defined by

$$\Psi_{x_i}(\mathbf{d}^{\alpha}_{\hat{\mathbf{x}}}) := \begin{cases} \mathbf{d}^{(\alpha_1,\ldots,\alpha_i+1,\ldots,\alpha_n)}_{\hat{\mathbf{x}}}, & \text{if } \alpha_1 = \cdots = \alpha_{i-1} = 0, \\ 0, & \text{otherwise.} \end{cases}$$

Here and hereafter, $J_F(\hat{\mathbf{x}})$ denotes the Jacobian matrix of the polynomial system F evaluated at $\hat{\mathbf{x}}$. It is well known that when the corank of $J_F(\hat{\mathbf{x}})$ is one, $\mathcal{D}_{\hat{\mathbf{x}}}$ has the important property:

$$\dim(\mathcal{D}^{t}_{\hat{\mathbf{x}}}) - \dim(\mathcal{D}^{t-1}_{\hat{\mathbf{x}}}) = 1, \text{ for } 1 \le t \le \mu - 1.$$

Hence, it is also called the breadth-one case in [4]. For this special case, in [19], under the assumption that the first column of $J_F(\hat{\mathbf{x}})$ is zero, we employ both normalization and reduction techniques to compute a closed basis of $\mathcal{D}_{\hat{\mathbf{x}}}$ very efficiently by solving $\mu - 1$ linear systems with the size bounded by $n \times (n-1)$.

Theorem 2.3. *[19, Theorem 3.1] Suppose* $\hat{\mathbf{x}}$ *is an isolated breadth-one root of a given polynomial system* $F = \{f_1, \ldots, f_n\}$*, and the first column of* $J_F(\hat{\mathbf{x}})$ *is a zero vector. Set* $\Lambda_0 = 1$ *and* $\Lambda_1 = d_1$*, we can construct* Λ_k *incrementally for* k *from* 2 *by*

$$\Lambda_k = \Delta_k + a_{k,2}d_2 + a_{k,3}d_3 + \cdots + a_{k,n}d_n, \tag{6}$$

where Δ_k *is a differential functional which has no free parameters and can be obtained from previously computed* $\{\Lambda_0, \Lambda_1, \ldots, \Lambda_{k-1}\}$ *by*

$$\Delta_k = \Psi_{x_1}(\Lambda_{k-1}) + \sum_{j=1}^{k-2} a_{k-j,2}\Psi_{x_2}(\Lambda_j) + \cdots + \sum_{j=1}^{k-2} a_{k-j,n}\Psi_{x_n}(\Lambda_j). \tag{7}$$

The parameters $a_{k,i}$*, for* $i = 2, \ldots, n$*, are determined by solving*

$$\tilde{J}_F(\hat{\mathbf{x}}) \cdot \begin{pmatrix} a_{k,2} \\ \vdots \\ a_{k,n} \end{pmatrix} = - \begin{pmatrix} \Delta_k(f_1) \\ \vdots \\ \Delta_k(f_n) \end{pmatrix}, \tag{8}$$

where $\tilde{J}_F(\hat{\mathbf{x}})$ *consists of the last* $n-1$ *columns of* $J_F(\hat{\mathbf{x}})$*. The process stops when there is no solution for (8) and returns the multiplicity* $\mu := k$ *and* $\{\Lambda_0, \Lambda_1, \ldots, \Lambda_{\mu-1}\}$ *a closed basis of* $\mathcal{D}_{\hat{\mathbf{x}}}$.

In [19], if the first column of $J_F(\hat{\mathbf{x}})$ is not a zero vector, we apply a linear transformation of variables to obtain a new system and a new root, which satisfy the assumptions of Theorem 2.3. Finally, we can derive a closed basis of the local dual space of the original system at the original root by transforming back the computed basis. Unfortunately, these transformations always result in dense systems even if the original ones are sparse.

EXAMPLE 2.1. *[26] Consider a polynomial system*

$$F = \left\{ x_1^2 + x_2 - 3, x_1 + \frac{1}{8}x_2^2 - \frac{3}{2} \right\}.$$

The system F has $(1, 2)$ as a 3-fold isolated zero.

The Jacobian matrix of F at $(1, 2)$ is

$$J_F(1, 2) = \begin{bmatrix} 2 & 1 \\ 1 & \frac{1}{2} \end{bmatrix},$$

which has a non-trivial null vector $\mathbf{r} = (-\frac{1}{2}, 1)^T$. Then we apply a linear transformation of the variables

$$x_1 = -\frac{1}{2}y_1 + 2y_2, x_2 = y_1 + y_2,$$

to obtain a new dense polynomial system

$$G = \left\{ \frac{1}{4}y_1^2 - 2y_1y_2 + 4y_2^2 + y_1 + y_2 - 3, \frac{1}{8}y_1^2 + \frac{1}{4}y_1y_2 + \frac{1}{8}y_2^2 - \frac{1}{2}y_1 + 2y_2 - \frac{3}{2} \right\}.$$

The returned closed basis of the local dual space of G at the new point $(\frac{6}{5}, \frac{4}{5})$ by Theorem 2.3 is

$$\Lambda_0 = 1, \Lambda_1 = d_1, \Lambda_2 = d_1^2 - \frac{1}{20}d_2,$$

which can be transformed back to a closed basis of F at $(1, 2)$

$$\Lambda_0 = 1, \Lambda_1 = -\frac{1}{2}d_1 + d_2, \Lambda_2 = \frac{1}{4}d_1^2 - \frac{1}{2}d_1d_2 + d_2^2 - \frac{1}{10}d_1 - \frac{1}{20}d_2.$$

3. A Modified Algorithm for Computing a Closed basis of $\mathcal{D}_{\hat{\mathbf{x}}}$

In this section, we show how to avoid linear transformations in computing a closed basis $\{\Lambda_0, \Lambda_1, \ldots, \Lambda_{\mu-1}\}$ of the local dual space $\mathcal{D}_{\hat{\mathbf{x}}}$.

Let $\mathbf{r} = (r_1, r_2, \ldots, r_n)^T$ be a non-trivial null vector of $J_F(\hat{\mathbf{x}})$. Without loss of generality, we assume

$$|r_1| \geq |r_j|, \text{ for } 1 \leq j \leq n. \tag{9}$$

Otherwise, one can permute two variables to guarantee (9) is satisfied. Then we normalize $\mathbf{r}$ by r_1 and derive that

$$\mathbf{a}_1 = (1, a_{1,2}, \ldots, a_{1,n})^T = \left(1, \frac{r_2}{r_1}, \ldots, \frac{r_n}{r_1}\right)^T \tag{10}$$

is also a non-trivial null vector of $J_F(\hat{\mathbf{x}})$. We set $\Lambda_0 = 1$ and

$$\Lambda_1 = d_1 + a_{1,2}d_2 + \cdots + a_{1,n}d_n,\ |a_{1,2}| \leq 1, \ldots, |a_{1,n}| \leq 1. \tag{11}$$

Lemma 3.1. *Under the assumption of (11), the differential functional monomial d_1^k belongs to the support of Λ_k, for $k = 1, \ldots, \mu - 1$.*

Proof. By (11), it is clear that d_1 belongs to the support of Λ_1. If $\mu \geq 3$, Λ_2 will satisfy (4), and at least one of $\lambda_{1,1}$,$\lambda_{2,1}$,..., $\lambda_{n,1}$ will not be zero. Let t be the integer such that $\lambda_{t,1} \neq 0$, then the differential functional monomial $d_1 d_t$ belongs to the support of Λ_2 according to (5). It follows that d_t has a non-zero coefficient in $\Phi_{x_1}(\Lambda_2)$. Then from (4) and (11), we derive that $\lambda_{1,1} \neq 0$. Therefore, d_1^2 belongs to the support of Λ_2.

The rest proof is done by induction. Assume that the lemma is true for k and $k < \mu - 1$, then similar to the analysis above, Λ_{k+1} will satisfy (4) and at least one of $\lambda_{1,k}$,$\lambda_{2,k}$,...,$\lambda_{n,k}$ will not be zero. Let t be the integer such that $\lambda_{t,k} \neq 0$, then $d_1^k d_t$ belongs to the support of Λ_{k+1}. It follows that $d_1^{k-1} d_t$ has a non-zero coefficient in $\Phi_{x_1}(\Lambda_{k+1})$. Since $\dim(\mathcal{D}^{k+1}_{\hat{\mathbf{x}}}) - \dim(\mathcal{D}^{k}_{\hat{\mathbf{x}}}) = 1$ and $\text{degree}(\Lambda_k) = k$, we derive that $\lambda_{1,k} \neq 0$. Therefore, d_1^{k+1} belongs to the support of Λ_{k+1}. □

Remark 3.2. *According to Lemma 3.1, the coefficient of d_1^k in Λ_k is not zero, therefore, it can be normalized to be one. Moreover, we can assume that Λ_k does not have terms $\{1, d_1, d_1^2, \ldots, d_1^{k-1}\}$. Otherwise, one can reduce them by $\{\Lambda_0, \Lambda_1, \Lambda_2, \ldots, \Lambda_{k-1}\}$. Finally, the number of free parameters in (4) is reduced to $n - 1$.*

Lemma 3.3. *Under the assumption of (11), after performing the normalization and reductions above, we have*

$$\begin{cases} \Phi_{x_1}(\Lambda_k) = \Lambda_{k-1}, \\ \Phi_{x_i}(\Lambda_k) = a_{k,i}\Lambda_0 + a_{k-1,i}\Lambda_1 + \cdots + a_{1,i}\Lambda_{k-1},\ \textit{for } i = 2, \ldots, n, \end{cases} \tag{12}$$

where $a_{j,i}$ is the coefficient of d_i in Λ_j, for $1 \leq j \leq k$ and $k < \mu$.

Proof. By Lemma 3.1 and Remark 3.2, we know that Λ_k has a term d_1^k and there are no terms of $\{1, d_1, d_1^2, \ldots, d_1^{k-1}\}$ in Λ_k. Hence, according to (4), we derive that $\Phi_{x_1}(\Lambda_k) = \Lambda_{k-1}$. Furthermore, since $\Phi_{x_i}(\Lambda_k) \in \mathcal{D}^{k-1}_{\hat{\mathbf{x}}}$, we have

$$\Phi_{x_i}(\Lambda_k) = \lambda_{i,0}\Lambda_0 + \lambda_{i,1}\Lambda_1 + \cdots + \lambda_{i,k-1}\Lambda_{k-1},\ \text{for } i = 2, \ldots, n.$$

Using (5), we claim that $\lambda_{i,j}$ is equal to the coefficient of $d_1^j d_i$ in Λ_k. On the other hand, we know that $\Phi^j_{x_1}(\Lambda_k) = \Lambda_{k-j}$, hence the coefficient of $d_1^j d_i$ in

Λ_k is equal to the coefficient of d_i in Λ_{k-j}, which is equal to $a_{k-j,i}$. Hence, $\lambda_{i,j} = a_{k-j,i}$ for $1 \le j \le k-1$ and we prove the second equality in (12). $\square$

According to Lemma 3.3, from a closed basis $\{\Lambda_0, \Lambda_1, \ldots, \Lambda_{k-1}\}$ of $\mathcal{D}^{k-1}_{\hat{\mathbf{x}}}$ to compute a new element Λ_k in $\mathcal{D}^{k}_{\hat{\mathbf{x}}}/\mathcal{D}^{k-1}_{\hat{\mathbf{x}}}$, the only $n-1$ free parameters are $a_{k,i}$. Now we modify Theorem 2.3 under the assumption (11), in order to avoid linear transformations.

Theorem 3.4. *[18] Suppose $\hat{\mathbf{x}}$ is an isolated breadth-one root of a given polynomial system $F = \{f_1, \ldots, f_n\}$. Under the assumption (11), i.e., $\Lambda_0 = 1$ and $\Lambda_1 = d_1 + a_{1,2}d_2 + \cdots + a_{1,n}d_n$, we can construct Λ_k incrementally for k from 2 by (6), where*

$$\Delta_k = \Psi_{x_1}(\Lambda_{k-1}) + \sum_{j=1}^{k-1} a_{k-j,2}\Psi_{x_2}(\Lambda_j) + \cdots + \sum_{j=1}^{k-1} a_{k-j,n}\Psi_{x_n}(\Lambda_j), \qquad (13)$$

and the parameters $a_{k,2}, \ldots, a_{k,n}$ are determined by solving (8). The process stops when there is no solution for (8) and returns the multiplicity $\mu := k$ and $\{\Lambda_0, \Lambda_1, \ldots, \Lambda_{\mu-1}\}$ a closed basis of $\mathcal{D}_{\hat{\mathbf{x}}}$.

Proof. According to Lemma 3.3 and formulas (5),(6) and (13), the constructed Λ_k satisfies the *stability property* (3). Moreover, by solving (8), we guarantee that $\Lambda_k(f_i) = 0$ for $i = 1, \ldots, n$. Therefore, by Lemma 2.2, the set $\{\Lambda_0, \Lambda_1, \ldots, \Lambda_{\mu-1}\}$ is a closed basis of $\mathcal{D}_{\hat{\mathbf{x}}}$. $\square$

Remark 3.5. *In Theorem 3.4, the first column of $J_F(\hat{\mathbf{x}})$ is not required to be zero. As showed in [19], when solving (8), only the vector on the right side is updated, while the matrix of the size $n \times (n-1)$ on the left side is fixed. So we apply the LU decomposition to $\tilde{J}_F(\hat{\mathbf{x}})$, then solve two triangular systems instead of solving (8).*

Now we consider Example 2.1 again. Since $J_F(1,2)$ has a non-trivial null vector $\mathbf{r} = (-\frac{1}{2}, 1)^T$, we permute two variables $x_1 \leftrightarrow x_2$, then apply the method described in Theorem 3.4 for computing a closed basis of the local dual space of F at $(1,2)$. We derive that

$$\Lambda_0 = 1, \Lambda_1 = -\frac{1}{2}d_1 + d_2, \Lambda_2 = \frac{1}{4}d_1^2 - \frac{1}{2}d_1d_2 + d_2^2 - \frac{1}{8}d_1.$$

In [19], in order to compute Λ_k of $\mathcal{D}^{k}_{\hat{\mathbf{x}}}/\mathcal{D}^{k-1}_{\hat{\mathbf{x}}}$, we need to construct Δ_k by (7) and evaluate $\Delta_k(f_i)$, for $i = 1, \ldots, n$. Even if the input system F is sparse, the differential functional Δ_k could still be very dense. Hence, the evaluation of the vector on the right side of (8) could be very expensive sometimes.

EXAMPLE 3.1. *Consider a polynomial system $F = \{f_1, \ldots, f_s\}$*

$$\begin{aligned} f_i &= x_i^3 + x_i^2 - x_{i+1}, \text{ if } i < s, \\ f_s &= x_s^2, \end{aligned}$$

with a breadth-one zero $(0, \ldots, 0)$ of multiplicity 2^s.

As shown in [19], for $s = 6$, about 17MB of memory is used to store the local dual bases and it takes about 3 hours to compute all of them. Moreover, for $s = 7$, we are not able to obtain all Λ_k in 2 days, and for $s = 9$, the estimated store space is about 1GB. It is not a surprise that the computation is dominated by the evaluation of $\Delta_k(F)$ in (8).

Let P_k and L_k be the differentiation operators corresponding to Δ_k and Λ_k respectively. The evaluation of $\Delta_k(F)$ is divided into two steps. First, for $2 \leq k \leq \mu$, we construct polynomial systems $P_k(F) = \{P_k(f_1), \ldots, P_k(f_n)\}$ iteratively by

$$P_k(F) = \sum_{j=1}^{k-1} \frac{j}{k} \cdot J_{L_{k-j}(F)} \cdot \mathbf{a}_j \text{ and } L_k(F) = P_k(F) + J_F \cdot \mathbf{a}_k, \tag{14}$$

where $L_1(F) = J_F \cdot \mathbf{a}_1$, $\mathbf{a}_1 = (1, a_{1,2}, \ldots, a_{1,n})^T$, $J_{L_j(F)}$ is the Jacobian matrix of $L_j(F)$, and $\mathbf{a}_j = (0, a_{j,2}, \ldots, a_{j,n})^T$ for $j > 1$. Secondly, we compute the evaluation $\Delta_k(f_i)$ by

$$\Delta_k(f_i) = P_k(f_i)(\hat{\mathbf{x}}), \text{ for } i = 1, \ldots, n.$$

The routine `MSB1` below takes a polynomial system $F = \{f_1, f_2, \ldots, f_n\}$ and an isolated root $\hat{\mathbf{x}} \in \mathbb{K}^n$ of I as input, where the corank of $J_F(\hat{\mathbf{x}})$ is one, and returns the multiplicity μ of $\hat{\mathbf{x}}$ and a closed basis of $\mathcal{D}_{\hat{\mathbf{x}}}$. Besides, we take $\mathbf{a}_1$,$\mathbf{a}_2$,...,$\mathbf{a}_{\mu-1}$ as output too, since one can construct all Λ_k by (6) and (13) immediately after they are computed. Another reason for outputting $\mathbf{a}_i$'s is that these values are important for multiple root refinement and verification if $\hat{\mathbf{x}}$ is only given with limited precision, which will be discussed in the next section.

Algorithm 3.6. *MSB1*
Input: *A polynomial system $F = \{f_1, f_2, \ldots, f_n\}$ and a root $\hat{\mathbf{x}} \in \mathbb{K}^n$.*
Output: *The multiplicity μ, the parameters $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_{\mu-1}$ and a closed basis $\{\Lambda_0, \Lambda_1, \ldots, \Lambda_{\mu-1}\}$ of the local dual space $\mathcal{D}_{\hat{\mathbf{x}}}$.*

1. *Compute a non-trivial null vector $\mathbf{r} = (r_1, r_2, \ldots, r_n)^T$ of $J_F(\hat{\mathbf{x}})$. Let t be the integer s.t. $|r_t| \geq |r_j|$, $1 \leq j \leq n$.*
   Apply the permutation of variables $x_1 \leftrightarrow x_t$ to F, $\hat{\mathbf{x}}$, J_F and $\mathbf{r}$. Set

$$\mathbf{a}_1 := \left(1, \frac{r_2}{r_1}, \frac{r_3}{r_1}, \ldots, \frac{r_n}{r_1}\right)^T \text{ and } L_1(F) := J_F \cdot \mathbf{a}_1.$$

   Compute the LU Decomposition of $\tilde{J}_F(\hat{\mathbf{x}})$: $\tilde{J}_F(\hat{\mathbf{x}}) = P \cdot L \cdot U$, where P is the pivot matrix. Set $k := 2$.
2. *Compute $P_k(F)$ by (14) and evaluate it at $\hat{\mathbf{x}}$ to get $\Delta_k(F)$, and solve*

$$L \cdot \mathbf{b} = -P^{-1} \cdot \Delta_k(F).$$

If the last entry in $\mathbf{b}$ *is zero, solve* $U_{1..(n-1),:} \cdot \mathbf{c} = \mathbf{b}$*, and set*

$$\mathbf{a}_k := \begin{pmatrix} 0 \\ \mathbf{c} \end{pmatrix} \quad \text{and } L_k(F) := P_k(F) + J_F \cdot \mathbf{a}_k,$$

and repeat with $k := k + 1$*. Otherwise, set* $\mu := k$*, go to Step 3.*

3. *Construct* $\{\Lambda_0, \Lambda_1, \ldots, \Lambda_{\mu-1}\}$ *by (6) and (13) using computed vectors* $\mathbf{a}_1$,$\mathbf{a}_2$, $\ldots, \mathbf{a}_{\mu-1}$*, and permute variables* $d_1 \leftrightarrow d_t$*.*

Note that we run Step 3 in Algorithm 3.6 only when a closed basis for the local dual space is wanted. If one is only interested in refining and certifying approximate isolated singular solutions, this step can be omitted. We have implemented the algorithm in Maple. In the following table, we show the time needed for computing all $\mathbf{a}_j$ in Example 3.1, for $j = 1, \ldots, 2^s - 1$.

s	6	7	8	9	10
multiplicity	64	128	256	512	1024
time(sec.)	0.593	1.377	3.445	10.913	44.659

4. Verified Multiple Roots of Polynomial Systems

As mentioned in [21], in real-life applications it is common to work with approximate inputs, and usually we need to decide (numerically) whether an (approximate) system possesses a unique real or complex root in a given domain.

Standard verification methods for nonlinear systems are based on the following theorem [13, 23, 31].

Theorem 4.1. *Let* $F \in \mathbb{R}^n$ *be a polynomial system with* $F = \{f_1, \ldots, f_n\}$*, and* $\tilde{\mathbf{x}} \in \mathbb{R}^n$ *a real point. Given an interval domain* $X \in \mathbb{IR}^n$ *with* $\tilde{\mathbf{x}} \in X$*, and an interval matrix* $M \in \mathbb{IR}^{n \times n}$ *satisfies* $\nabla f_i(X) \subseteq M_{i,:}$*, for* $i = 1, \ldots, n$*. Denote by* I *the* $n \times n$ *identity matrix and assume*

$$-J_F(\tilde{\mathbf{x}})F(\tilde{\mathbf{x}}) + (I - J_F(\tilde{\mathbf{x}})M)X \subseteq int(X).$$

Then there is a unique $\hat{\mathbf{x}} \in X$ *with* $F(\hat{\mathbf{x}}) = 0$*. Moreover, every matrix* $\tilde{M} \in M$ *is nonsingular. In particular, the Jacobian matrix* $J_F(\hat{\mathbf{x}})$ *is nonsingular.*

In [33], they introduced a smoothing parameter to certify a double root of a slightly perturbed system using Theorem 4.1. It should be noticed that a double root is the simplest breadth-one root with multiplicity two.

In [21], they applied Theorem 4.1 to a deflated system to verify a multiple root of a nearby system with the computed local dual structure. Their method can deal with arbitrary multiple roots, but the computed local dual structure depends on the accuracy of the given approximate input.

For an isolated breadth-one root with multiplicity greater than two, in [33, Theorem 4.2], they proved that it is impossible to compute an inclusion of a

multiple root by adding only a smoothing parameter to one selected equation. We describe a method below to construct a deflated system using $\mu-1$ smoothing parameters and a parameterized basis in $\mathbf{a}_1, \ldots, \mathbf{a}_{\mu-1}$ of $\mathcal{D}_{\hat{\mathbf{x}}}$ to certify a breadth-one multiple root for $\mu \geq 2$.

Let $F = \{f_1, \ldots, f_n\} \in R$ be given. Suppose $\hat{\mathbf{x}} \in \mathbb{K}^n$ is an isolated root of F with multiplicity μ and $J_F(\hat{\mathbf{x}})$ has corank one. We show how to choose a pair of suitable variable and equation to perform the perturbation. In fact, the perturbed variable x_i can be determined by choosing a column of $J_F(\hat{\mathbf{x}})$, which can be written as a linear combination of the other $n-1$ columns. Similarly, suppose the j-th row of $J_F(\hat{\mathbf{x}})$ can be written as a linear combination of the other $n-1$ linearly independent rows, then we add the perturbed univariate polynomial in x_i to f_j. Finally, we permute

$$x_1 \leftrightarrow x_i \text{ and } f_1 \leftrightarrow f_j \tag{15}$$

to construct the deflated system (16).

Assumption 4.2. *Suppose $J_F(\hat{\mathbf{x}})$ has corank one. We assume below that the first row (column) of $J_F(\hat{\mathbf{x}})$ can be written as a linear combination of its other rows (columns). This can always be achieved by permuting variables and renumbering equations as above.*

We introduce $\mu-1$ smoothing parameters $b_0, b_1, \ldots, b_{\mu-2}$ and construct a deflated system $G(\mathbf{x}, \mathbf{b}, \mathbf{a})$ with μn variables and μn equations:

$$G(\mathbf{x}, \mathbf{b}, \mathbf{a}) = \begin{pmatrix} F_1(\mathbf{x}, \mathbf{b}) = F(\mathbf{x}) - \left(\sum_{\nu=0}^{\mu-2} \frac{b_\nu x_1^\nu}{\nu!}\right) \mathbf{e}_1 \\ F_2(\mathbf{x}, \mathbf{b}, \mathbf{a}_1) \\ F_3(\mathbf{x}, \mathbf{b}, \mathbf{a}_1, \mathbf{a}_2) \\ \vdots \\ F_\mu(\mathbf{x}, \mathbf{b}, \mathbf{a}_1, \ldots, \mathbf{a}_{\mu-1}) \end{pmatrix}, \tag{16}$$

where $\mathbf{b} = (b_0, b_1, \ldots, b_{\mu-2})$, $\mathbf{a} = (\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_{\mu-1})$, $\mathbf{a}_1 = (1, a_{1,2}, \ldots, a_{1,n})^T$, $\mathbf{a}_k = (0, a_{k,2}, \ldots, a_{k,n})^T$ for $1 < k \leq \mu$, and

$$F_k(\mathbf{x}, \mathbf{b}, \mathbf{a}_1, \ldots, \mathbf{a}_{k-1}) = L_{k-1}(F_1). \tag{17}$$

Theorem 4.3. *Suppose $G(\hat{\mathbf{x}}, \hat{\mathbf{b}}, \hat{\mathbf{a}}) = 0$. Under Assumption 4.2, if $J_G(\hat{\mathbf{x}}, \hat{\mathbf{b}}, \hat{\mathbf{a}})$ is nonsingular, then $\hat{\mathbf{x}}$ is an isolated root of the polynomial system $F_0(\mathbf{x}) = F_1(\mathbf{x}, \hat{\mathbf{b}})$ with multiplicity μ and the corank of $J_{F_0}(\hat{\mathbf{x}})$ is one.*

Proof. From $G(\hat{\mathbf{x}}, \hat{\mathbf{b}}, \hat{\mathbf{a}}) = 0$, we have $F_0(\hat{\mathbf{x}}) = 0$ and

$$F_2(\hat{\mathbf{x}}, \hat{\mathbf{b}}, \hat{\mathbf{a}}_1) = J_{F_0}(\hat{\mathbf{x}}) \cdot \hat{\mathbf{a}}_1 = 0.$$

Since $\hat{\mathbf{a}}_1 \neq 0$, we derive that

$$\operatorname{rank}(J_{F_0}(\hat{\mathbf{x}})) \leq n-1.$$

Moreover, from the expression of $\hat{\mathbf{a}}_1$, we know that the first column of $J_{F_0}(\hat{\mathbf{x}})$ can be written as a linear combination of the other $n-1$ columns. Therefore,

$$\operatorname{rank}(\tilde{J}_{F_0}(\hat{\mathbf{x}})) = \operatorname{rank}(J_{F_0}(\hat{\mathbf{x}})) \leq n-1,$$

where $\tilde{J}_{F_0}(\hat{\mathbf{x}})$ consists of the last $n-1$ columns of $J_{F_0}(\hat{\mathbf{x}})$. Similarly, since $F_k(\hat{\mathbf{x}}, \hat{\mathbf{b}}, \hat{\mathbf{a}}_1, \ldots, \hat{\mathbf{a}}_{k-1}) = 0$, by Theorem 3.4 and (6), we derive that

$$\operatorname{rank}(\Delta_{k-1}(F_0), \tilde{J}_{F_0}(\hat{\mathbf{x}})) = \operatorname{rank}(\tilde{J}_{F_0}(\hat{\mathbf{x}})) \leq n-1, \text{ for } 2 < k \leq \mu.$$

In order to prove that $\hat{\mathbf{x}}$ is a breadth-one root of $F_0(\mathbf{x}) = 0$ with multiplicity μ, we need to show that

$$\operatorname{rank}(\tilde{J}_{F_0}(\hat{\mathbf{x}})) = n-1 \text{ and } \operatorname{rank}(\Delta_{\mu}(F_0), \tilde{J}_{F_0}(\hat{\mathbf{x}})) = n. \tag{18}$$

It is interesting to see that we can use the equivalent relations

$$\frac{\partial F_k}{\partial a_{j,i}} = \frac{\partial F_{k-j}}{\partial x_i}, \text{ for } 2 \leq i \leq n \text{ and } 1 \leq j \leq k-1, \tag{19}$$

to obtain a simplified expression of J_G

$$J_G = \begin{pmatrix} J_{F_1} & \mathbf{e}_1 & x_1\mathbf{e}_1 & \cdots & \frac{x_1^{\mu-2}}{(\mu-2)!}\mathbf{e}_1 & 0 & 0 & \cdots & 0 & 0 \\ J_{F_2} & 0 & \mathbf{e}_1 & \cdots & \frac{x_1^{\mu-3}}{(\mu-3)!}\mathbf{e}_1 & \tilde{J}_{F_0} & 0 & \cdots & 0 & 0 \\ J_{F_3} & 0 & 0 & \cdots & \frac{x_1^{\mu-4}}{(\mu-4)!}\mathbf{e}_1 & \tilde{J}_{F_2} & \tilde{J}_{F_0} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ J_{F_{\mu-1}} & 0 & 0 & \cdots & \mathbf{e}_1 & \tilde{J}_{F_{\mu-2}} & \tilde{J}_{F_{\mu-3}} & \cdots & \tilde{J}_{F_0} & 0 \\ J_{F_\mu} & 0 & 0 & \cdots & 0 & \tilde{J}_{F_{\mu-1}} & \tilde{J}_{F_{\mu-2}} & \cdots & \tilde{J}_{F_2} & \tilde{J}_{F_0} \end{pmatrix}, \tag{20}$$

where J_{F_k} denotes the Jacobian matrix of $F_k(\mathbf{x}, \mathbf{b}, \mathbf{a}_1, \ldots, \mathbf{a}_{k-1})$ with respect to $\mathbf{x}$ and $\tilde{J}_{F_k}$ consists of the last $n-1$ columns of J_{F_k}, for $2 \leq k \leq \mu$.

If $\operatorname{rank}(\tilde{J}_{F_0}(\hat{\mathbf{x}})) \leq n-2$, then there exists a non-trivial vector in its kernel. Note that $\tilde{J}_{F_0}(\hat{\mathbf{x}})$ is the only non-zero element in the last column, we claim that there will exist a non-trivial vector in the kernel of $J_G(\hat{\mathbf{x}}, \hat{\mathbf{b}}, \hat{\mathbf{a}})$, which is a contradiction. Hence, we derive that

$$\operatorname{rank}(J_{F_0}(\hat{\mathbf{x}})) = \operatorname{rank}(\Delta_{k-1}(F_0), \tilde{J}_{F_0}(\hat{\mathbf{x}})) = \operatorname{rank}(\tilde{J}_{F_0}(\hat{\mathbf{x}})) = n-1.$$

On the other hand, from (13) and (17), we derive that

$$J_G(\hat{\mathbf{x}}, \hat{\mathbf{b}}, \hat{\mathbf{a}})_{:,1..(\mu-1)n+1} \cdot \mathbf{v} = (0, \ldots, 0, \Delta_{\mu}(F_0))^T,$$

where

$$\mathbf{v} = \frac{1}{\mu} \cdot (1, \hat{a}_{2,2}, \ldots, \hat{a}_{2,n}, 0, \ldots, 0, 2\hat{a}_{3,2}, \ldots, 2\hat{a}_{3,n}, \ldots, (\mu-1)\hat{a}_{\mu,2}, \ldots, (\mu-1)\hat{a}_{\mu,n})^T.$$

So that, if $\text{rank}(\Delta_\mu(F_0), \tilde{J}_{F_0}(\hat{\mathbf{x}})) \leq n-1$, then there exists a non-trivial vector in the kernel of $J_G(\hat{\mathbf{x}}, \hat{\mathbf{b}}, \hat{\mathbf{a}})$, which is a contradiction. Hence, we have

$$\text{rank}(\Delta_\mu(F_0), \tilde{J}_{F_0}(\hat{\mathbf{x}})) = n.$$

Therefore, according to Theorem 3.4, $\hat{\mathbf{x}}$ is an isolated breadth-one root of $F_0(\mathbf{x}) = 0$ with multiplicity μ. □

Theorem 4.4. *Suppose $\hat{\mathbf{x}}$ is an exact isolated root of $F(\mathbf{x}) = 0$ with multiplicity μ and the corank of $J_F(\hat{\mathbf{x}})$ is one exactly. Under Assumption 4.2, we have*

$$\text{rank}(\tilde{J}_F(\hat{\mathbf{x}}), \mathbf{e}_1) = n. \tag{21}$$

Evaluating (16) at $\hat{\mathbf{b}} = \mathbf{0}$, i.e., no perturbations for F, then $J_G(\hat{\mathbf{x}}, \mathbf{0}, \hat{\mathbf{a}})$ is non-singular.

Proof. According to Assumption 4.2, the first row of $\tilde{J}_F(\hat{\mathbf{x}})$ can be written as a linear combination of its other rows. Since the rank of $\tilde{J}_F(\hat{\mathbf{x}})$ is $n-1$, its last $n-1$ rows must be linear independent. Therefore, we have (21).

Assume $\mathbf{v}$ is a nontrivial vector in the kernel of $J_G(\hat{\mathbf{x}}, \mathbf{0}, \hat{\mathbf{a}})$. If $v_1 = 0$, by checking the columns of J_G in (20), and using (21), we can show that $\mathbf{v} = 0$. If $v_1 \neq 0$, we can assume $v_1 = 1$. Similar to the second part of proof of Theorem 4.3, we derive that $\Delta_\mu(F)$ can be written as a linear combination of the columns from $\tilde{J}_F(\hat{\mathbf{x}})$, which is a contradiction. Hence, there exists no nontrivial vector in the kernel of $J_G(\hat{\mathbf{x}}, \mathbf{0}, \hat{\mathbf{a}})$. In other word, $J_G(\hat{\mathbf{x}}, \mathbf{0}, \hat{\mathbf{a}})$ is nonsingular. □

Now, we apply Theorem 4.1 on the deflated system. If the test succeeds, we derive verified narrow error bounds with the property that a slightly perturbed system is proved to have a breadth-one multiple root within the computed bounds.

Theorem 4.5. *Suppose Theorem 4.1 is applicable to $G(\mathbf{x}, \mathbf{b}, \mathbf{a})$ in (16) and yields inclusions for $\hat{\mathbf{x}}$, $\hat{\mathbf{b}}$ and $\hat{\mathbf{a}}$ such that $G(\hat{\mathbf{x}}, \hat{\mathbf{b}}, \hat{\mathbf{a}}) = 0$. Then $\hat{\mathbf{x}}$ is an isolated breadth-one root of $F_0(\mathbf{x}) := F_1(\mathbf{x}, \hat{\mathbf{b}})$ with multiplicity μ.*

Proof. A direct result of Theorem 4.1 and Theorem 4.3. □

EXAMPLE 4.1. *[33, Example 4.11] Consider a polynomial system*

$$F = \{x_1^2 x_2 - x_1 x_2^2, x_1 - x_2^2\}.$$

The system F has $(0,0)$ as a 4-fold isolated zero.

The Jacobian matrix of F at $(0,0)$ is

$$J_F(0,0) = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}.$$

We choose x_2 as the perturbed variable and add the univariate polynomial $b_0 + b_1x_2 + \frac{b_2}{2}x_2^2$ to the first equation in F to construct the parameterized deflated system

$$G(\mathbf{x}, \mathbf{b}, \mathbf{a}) = \begin{pmatrix} x_1^2x_2 - x_1x_2^2 - b_0 - b_1x_2 - \frac{b_2}{2}x_2^2 \\ x_1 - x_2^2 \\ 2a_1x_1x_2 - a_1x_2^2 + x_1^2 - 2x_1x_2 - b_1 - b_2x_2 \\ a_1 - 2x_2 \\ a_1^2x_2 + 2a_1x_1 - 2a_1x_2 + 2a_2x_1x_2 - a_2x_2^2 - x_1 - \frac{b_2}{2} \\ a_2 - 1 \\ a_1^2 + 2a_1a_2x_2 - a_1 + 2a_2x_1 - 2a_2x_2 + 2a_3x_1x_2 - a_3x_2^2 \\ a_3 \end{pmatrix}.$$

Applying the INTLAB function verifynlss in Matlab [32] to G with the initial approximation

$$[0.002, 0.003, -0.001, 0.0015, -0.002, 0.002, 1.001, -0.01]$$

to obtain inclusions

$$\begin{bmatrix} -0.00000000000001, & 0.00000000000001] \\ [-0.00000000000001, & 0.00000000000001] \\ [-0.00000000000001, & 0.00000000000001] \\ [-0.00000000000001, & 0.00000000000001] \\ [-0.00000000000001, & 0.00000000000001] \end{bmatrix}$$

This proves that the perturbed system $F_0(\mathbf{x})$ ($|b_i| \leq 10^{-14}, i = 0, 1, 2$) has a 4-fold root $\hat{\mathbf{x}}$ with $-10^{-14} \leq \hat{\mathbf{x}} \leq 10^{-14}$.

EXAMPLE 4.2. *[33, Example 4.7] Consider a polynomial system*

$$F = \{x_1^2 - x_2^2, x_1 - x_2^2\}.$$

The system F has $(0, 0)$ as a 2-fold isolated zero.

For this example, as mentioned in [33], the iteration is sensitive to the initial approximations. Applying the INTLAB function verifynlss2 in Matlab to F with the starting point $[0.002, 0.001]$, we will obtain inclusions

$$\begin{bmatrix} -0.00000000000001, & 0.00000000000001] \\ [-0.00000000000001, & 0.00000000000001] \\ [-0.00000000000001, & 0.00000000000001] \end{bmatrix}$$

However, for the initial approximation $[0.001, 0.001]$, we obtain

$$\begin{bmatrix} 0.49999999999999, & 0.50000000000001] \\ [0.70710678118654, & 0.70710678118655] \\ [-0.25000000000001, & -0.24999999999999] \end{bmatrix}$$

which finds the double root $(0.5, 1/\sqrt{2})$ of $x_1^2 - x_2^2 + 0.25 = 0$ and $x_1 - x_2^2 = 0$.

For this reason, we prefer to use the symbolic-numeric method described in [20] to refine initial approximations first, next to construct the parameterized deflated system with smoothing parameters, finally we use the method in Theorem 4.5 to compute inclusions of multiple roots. We show the routine `MRRB1` below for refining an approximate singular solution to high precision in the breadth-one case. The input of `MRRB1` is a sequence of polynomial systems $F_1, F_2, \ldots, F_\mu$ defined in (16) (17) with $\mathbf{b} = \mathbf{0}$ (without perturbations), $F_{\mu+1} = P_\mu(F_1)$ and an approximate solution $\hat{\mathbf{x}}$ of $F_1 = 0$. The algorithm in [20] has been improved in `MRRB1` by avoiding linear transformations and constructing differential functionals repeatedly.

Algorithm 4.6. *MRRB1*
Input: *A sequence of systems $F_1, \ldots, F_{\mu+1}$, a point $\hat{\mathbf{x}} \in \mathbb{K}^n$.*
Output: *A refined point $\hat{\mathbf{x}}$ and refined parameters $\hat{\mathbf{a}}_2, \ldots, \hat{\mathbf{a}}_\mu$.*

1. `Regularized Newton Iteration`*: Solve the least squares problem*

$$\left(J_{F_1}^*(\hat{\mathbf{x}}) \cdot J_{F_1}(\hat{\mathbf{x}}) + \sigma_n I_n\right) \mathbf{y} = -J_{F_1}^*(\hat{\mathbf{x}}) \cdot F_1(\hat{\mathbf{x}}),$$

   where $J_{F_1}^*(\hat{\mathbf{x}})$ is the conjugate transpose of $J_{F_1}(\hat{\mathbf{x}})$, σ_n is the smallest singular value of $J_{F_1}(\hat{\mathbf{x}})$ and I_n is the $n \times n$ identity matrix.
   Set $\hat{\mathbf{x}} := \hat{\mathbf{x}} + \hat{\mathbf{y}}$.
2. *For $2 \le k \le \mu$, solve the least squares problem*

$$F_k(\hat{\mathbf{x}}, \hat{\mathbf{a}}_1, \ldots, \hat{\mathbf{a}}_{k-2}, \mathbf{a}_{k-1}) = 0$$

   to obtain $\hat{\mathbf{a}}_{k-1}$.
3. *Solve the linear system*

$$\left[F_{\mu+1}(\hat{\mathbf{x}}, \hat{\mathbf{a}}_1, \ldots, \hat{\mathbf{a}}_{\mu-1}), \frac{\partial F_1(\hat{\mathbf{x}})}{\partial x_2}, \ldots, \frac{\partial F_1(\hat{\mathbf{x}})}{\partial x_n}\right] \mathbf{v} = -F_\mu(\hat{\mathbf{x}}, \hat{\mathbf{a}}_1, \ldots, \hat{\mathbf{a}}_{\mu-1}),$$

   where $\mathbf{v} = (v_1, \ldots, v_n)^T$. Set $\delta := v_1/\mu$.
4. *Return $\hat{\mathbf{a}}_1, \ldots, \hat{\mathbf{a}}_{\mu-1}$ and*

$$\hat{\mathbf{x}} := \hat{\mathbf{x}} + \delta \begin{pmatrix} 1 \\ \hat{a}_{1,2} \\ \vdots \\ \hat{a}_{1,n} \end{pmatrix}.$$

Now we consider Example 4.2 again. For $[0.002, 0.001]$, after running `MRRB1` two times in Maple then applying the INTLAB function `verifynlss` to G in Matlab, we obtain

$$\begin{bmatrix} -0.00000000000001, & 0.00000000000001] \\ [\ -0.00000000000001, & 0.00000000000001] \\ [\ -0.00000000000001, & 0.00000000000001] \end{bmatrix}$$

Similarly, for $[0.001, 0.001]$, we obtain

$$\begin{bmatrix} [-0.00000000000001, & 0.00000000000001] \\ [-0.00000000000001, & 0.00000000000001] \\ [-0.00000000000001, & 0.00000000000001] \end{bmatrix}$$

EXAMPLE 4.3. *[19] Consider a system $F = \{f_1, \ldots, f_s\}$ given by*

$$\begin{aligned} f_i &= x_i^2 + x_i - x_{i+1}, \text{ if } i < s, \\ f_s &= x_s^3, \end{aligned}$$

with a breadth-one singular zero $(0, 0, \ldots, 0)$ of multiplicity 3.

We run `MRRB1` three times in Maple for initial approximate roots near the origin, whose errors are around 10^{-4}, to obtain the refined $\hat{\mathbf{x}}$ and $\hat{\mathbf{a}}$ with errors about 10^{-12}. Then we choose x_s as the perturbed variable and add the univariate polynomial $b_0 + b_1 x_s$ to the last polynomial f_s to construct the parameterized deflated system. In the following table, $|X|$ and $|B|$ denotes the interval size of inclusions for $\hat{\mathbf{x}}$ and $\hat{\mathbf{b}}$, which are computed by applying INTLAB function verifynlss in Matlab to the deflated system (16) and $(\hat{\mathbf{x}}, \mathbf{0}, \hat{\mathbf{a}})$.

s	$\lvert X\rvert$	$\lvert B\rvert$
10	10^{-14}	10^{-14}
20	10^{-14}	10^{-14}
50	10^{-14}	10^{-14}
100	10^{-14}	10^{-14}
200	10^{-12}	10^{-12}
500	10^{-12}	10^{-12}
1000	10^{-12}	10^{-12}

Acknowledgments

The first author is grateful to Fabrice Rouillier and Philippe Trebuchet from Paris VI for their constructive comments. The second author would like to thank Gilles Villard and Nathalie Revol for hosing her visit to LIP-ENS de Lyon and SCAN in 2010. The paper is stimulated by fruitful discussions during the visit.

References

[1] Chen, X., Nashed, Z., Qi, L., 1997. Convergence of Newton's method for singular smooth and nonsmooth equations using adaptive outer inverses. SIAM J. on Optimization 7 (2), 445–462.

[2] Corless, R., Gianni, P., Trager, B., 1997. A reordered Schur factorization method for zero-dimensional polynomial systems with multiple roots. In: Küchlin (Ed.), Proc. 1997 Internat. Symp. Symbolic Algebraic Comput. ISSAC'97. ACM Press, New York, pp. 133–140.

[3] Dayton, B., Li, T., Zeng, Z., 2011. Multiple zeros of nonlinear systems. Mathematics of Computation 80, 2143–2168.

[4] Dayton, B., Zeng, Z., 2005. Computing the multiplicity structure in solving polynomial systems. In: Proceedings of the 2005 international symposium on Symbolic and algebraic computation. ISSAC '05. ACM, New York, NY, USA, pp. 116–123.

[5] Decker, D. W., Kelley, C. T., 1980. Newton's method at singular points. i. SIAM Journal on Numerical Analysis 17, 66–70.

[6] Decker, D. W., Kelley, C. T., 1980. Newton's method at singular points. ii. SIAM Journal on Numerical Analysis 17, 465–471.

[7] Decker, D. W., Kelley, C. T., 1982. Convergence acceleration for Newton's method at singular points. SIAM Journal on Numerical Analysis 19, 219–229.

[8] Giusti, M., Lecerf, G., Salvy, B., Yakoubsohn, J.-C., 2005. On location and approximation of clusters of zeros of analytic functions. Found. Comput. Math. 5 (3), 257–311.

[9] Giusti, M., Lecerf, G., Salvy, B., Yakoubsohn, J.-C., 2007. On location and approximation of clusters of zeros: case of embedding dimension one. Found. Comput. Math. 7 (1), 1–58.

[10] Griewank, A., 1980. Analysis and modification of Newton's method at singularities. Thesis, Australian National University.

[11] Griewank, A., 1985. On solving nonlinear equations with simple singularities or nearly singular solutions. SIAM Review 27 (4), 537–563.

[12] Griewank, A., Osborne, M. R., 1981. Newton's method for singular problems when the dimension of the null space is > 1. SIAM Journal on Numerical Analysis 18, 145–149.

[13] Krawczyk, R., 1969. Newton-algorithmen zur bestimmung von nullstellen mit fehlerschranken. Computing, 187–201.

[14] Lecerf, G., 2002. Quadratic Newton iteration for systems with multiplicity. Foundations of Computational Mathematics 2 (3), 247–293.

[15] Leykin, A., Verschelde, J., Zhao, A., 2006. Newton's method with deflation for isolated singularities of polynomial systems. Theoretical Computer Science 359, 111–122.

[16] Leykin, A., Verschelde, J., Zhao, A., 2007. Evaluation of jacobian matrices for newton's method with deflation to approximate isolated singular solutions of polynomial systems. In: Wang, D., Zhi, L. (Eds.), Symbolic-Numeric Computation. Trends in Mathematics. Birkhäuser Basel, pp. 269–278.

[17] Leykin, A., Verschelde, J., Zhao, A., 2008. Higher-order deflation for polynomial systems with isolated singular solutions. Algorithms in algebraic geometry 146 IMA Vol. Math. Appl., 79–97.

[18] Li, N., 2011. An improved method for evaluating Max Noether conditions: case of breadth. In: Moreno Maza, M. (Ed.), SNC'11 Proc. 2011 Internat. Workshop on Symbolic-Numeric Comput. ACM Press, New York, N. Y., pp. 102–103.

[19] Li, N., Zhi, L., 2012. Compute the multiplicity structure of an isolated singular solution: case of breadth one. Journal of Symbolic Computation 47, 700–710.

[20] Li, N., Zhi, L., 2012. Computing isolated singular solutions of polynomial systems: case of breadth one. SIAM Journal on Numerical Analysis 50 (1), 354–372.

[21] Mantzaflaris, A., Mourrain, B., 2011. Deflation and certified isolation of singular zeros of polynomial systems. In: Leykin, A. (Ed.), Proceedings of the 36th international symposium on Symbolic and algebraic computation. ISSAC '11. ACM, New York, NY, USA, pp. 249–256.

[22] Marinari, M., Mora, T., Möller, H., 1995. Gröbner duality and multiplicities in polynomial solving. In: Levelt, A. (Ed.), Proc. 1995 Internat. Symp. Symbolic Algebraic Comput. (ISSAC'95). ACM Press, New York, pp. 167–179.

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

[24] Mourrain, B., 1996. Isolated points, duality and residues. J. of Pure and Applied Algebra 117 & 118, 469–493.

[25] Norio, Y., 1984. Regularization of solutions of nonlinear equations with singular jacobian matrices. Journal of information processing 7 (1), 16–21.

[26] Ojika, T., 1987. Modified deflation algorithm for the solution of singular problems. J. Math. Anal. Appl. 123, 199–221.

[27] Ojika, T., Watanabe, S., Mitsui, T., 1983. Deflation algorithm for the multiple roots of a system of nonlinear equations. J. Math. Anal. Appl. 96, 463–479.

[28] Rall, L., 1966. Convergence of the Newton process to multiple solutions. Numer. Math. 9, 23–37.

[29] Reddien, G. W., 1978. On Newton's method for singular problems. SIAM Journal on Numerical Analysis 15, 993–996.

[30] Reddien, G. W., 1980. Newton's method and high order singularities. Comput. Math. Appl 5, 79–86.

[31] Rump, S., 1983. Solving algebraic problems with high accuracy. In: Proc. of the symposium on A new approach to scientific computation. Academic Press Professional, Inc., San Diego, CA, USA, pp. 51–120.

[32] Rump, S., 1999. INTLAB - INTerval LABoratory. In: Csendes, T. (Ed.), Developments in Reliable Computing. Kluwer Academic Publishers, Dordrecht, pp. 77–104.

[33] Rump, S., Graillat, S., 2009. Verified error bounds for multiple roots of systems of nonlinear equations. Numerical Algorithms 54 (3), 359–377.

[34] Shen, Y.-Q., Ypma, T. J., 2005. Newton's method for singular nonlinear equations using approximate left and right nullspaces of the Jacobian. Applied Numerical Mathematics 54 (2), 256 – 265, 6th IMACS.

[35] Stetter, H., 2004. Numerical Polynomial Algebra. SIAM, Philadelphia.

[36] Wu, X., Zhi, L., 2008. Computing the multiplicity structure from geometric involutive form. In: Jeffrey, D. (Ed.), Proc. 2008 Internat. Symp. Symbolic Algebraic Comput. (ISSAC'08). ACM Press, New York, N. Y., pp. 325–332.

[37] Wu, X., Zhi, L., 2012. Determining singular solutions of polynomial systems via symbolic-numeric reduction to geometric involutive forms. Journal of Symbolic Computation 47 (3), 227–238.

[38] Zeng, Z., 2009. The closedness subspace method for computing the multiplicity structure of a polynomial system. `http://orion.neiu.edu/~zzeng/Papers/csdual.pdf`.