Noname manuscript No.
(will be inserted by the editor)

Improved Two-Step Newton's Method for Computing Simple Multiple Zeros of Polynomial Systems

Nan Li · **Lihong Zhi**

Abstract Given a polynomial system f that is associated with an isolated singular zero ξ whose Jacobian matrix is of corank one, and an approximate zero x that is close to ξ, we propose an improved two-step Newton's method for refining x to converge to ξ with quadratic convergence. Our new approach is based on a closed-form basis of the local dual space and a recursive reduction of the simple multiple zero. By avoiding solving several least-squares problems appeared in the previous methods, an overall $2\times$-$5\times$ acceleration is achieved. The proof of the quadratic convergence of proposed iterations is also simplified significantly. Numerical experiments demonstrate up to $100\times$ speed-up when we replace the least-squares-solving calculations with closed-form solutions for refining approximate singular solutions of large-size problems (1000 equations and 1000 variables).

Keywords polynomial system, simple multiple zero, Newton's method, quadratic convergence

1 Introduction

Consider a square polynomial system $f = \{f_1, \dots, f_n\}$, where $f_i \in \mathbb{C}[X_1, \dots, X_n]$ for $i = 1, \dots, n$, and a zero $\xi \in \mathbb{C}^n$ of $f = 0$. Let $Df(\xi)$ denote the Jacobian matrix of f evaluated at ξ.

When $Df(\xi)$ is invertible, then Newton's method

$$N(f, x) = x - Df(x)^{-1} f(x) \tag{1}$$

This research is supported by the National Key Research Project of China 2018YFA0306702 (Zhi) and the National Natural Science Foundation of China 12071467 (Zhi), 12171324 (Li).

Nan Li
College of Mathematics and Statistics, Shenzhen University, Shenzhen 518060, Guangdong, China
E-mail: nan.li@szu.edu.cn

Lihong Zhi
Key Laboratory of Mathematics Mechanization, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China & University of Chinese Academy of Sciences, Beijing 100049, China
E-mail: lzhi@mmrc.iss.ac.cn

starting from an approximate zero x that is sufficiently close to ξ will converge to ξ quadratically, namely,

$$\|N(f,x)-\xi\| = O\left(\|x-\xi\|^2\right), \tag{2}$$

where $O(g)$ denotes that the value is bounded above by g up to a positive constant.

When $Df(\xi)$ is not invertible, i.e., $\dim\ker Df(\xi) \geq 1$, then Newton's method will fail to retain local quadratic convergence, because there may exist a positive-dimensional manifold near ξ that satisfies $\det Df(X) = 0$ [?]. Many modifications [?,?] have been proposed to restore the quadratic convergence of Newton's method for isolated singular zeros from different aspects, see the related works below.

When $\dim\ker Df(\xi) = 1$, a two-step Newton's method was proposed in [?] that refines an approximate zero x to converge quadratically to ξ: the first step of the iteration projects x to x' by solving a damped least squares problem (**??**), such that $x'-\xi$ approximately belongs to the one-dimensional linear space $\mathrm{span}_{\mathbb{C}}\{v'\}$, which approximately coincides with $\ker Df(\xi)$; the second step of the iteration estimates a step length δ by solving a sequence of least squares problems (**??**) and a linear system (**??**), such that the approximate solution $x'' = x' + \delta v'$ satisfies $\|x''-\xi\| = O(\|x-\xi\|^2)$. In this work, an improved two-step Newton's method is proposed without solving any least-squares problems or linear systems while retaining the quadratic convergence.

Main contribution For a square polynomial system f that is associated with an isolated zero ξ (unknown) of multiplicity μ satisfying $\dim\ker Df(\xi) = 1$, we propose a two-step Newton's method for refining an approximate zero x (known) to converge quadratically to ξ:

Step 1. Suppose the singular value decomposition of $Df(x)$ is

$$Df(x) = (u_1,\ldots,u_n)\cdot \mathrm{diag}(\sigma_1,\ldots,\sigma_n)\cdot(v_1,\ldots,v_n)^*,$$

then the first step of the refinement is to update x by

$$x' := x - \sum_{i=1}^{n-1}\frac{1}{\sigma_i}v_i u_i^* f(x).$$

Step 2. Suppose the singular value decomposition of $Df(x')$ is

$$Df(x') = (u'_1,\ldots,u'_n)\cdot \mathrm{diag}(\sigma'_1,\ldots,\sigma'_n)\cdot(v'_1,\ldots,v'_n)^*,$$

then the second step of the refinement is to update x' by

$$x'' := x' - \frac{1}{\mu}v'_n\frac{{u'_n}^*\Delta_{\mu-1}(f)}{{u'_n}^*\Delta_{\mu}(f)},$$

where $\Delta_{\mu-1}$ and Δ_μ are two linear differential functionals of order $\mu-1$ and μ, which are calculated recursively using the closed form (**??**).

The new proposed two-step Newton's method has achieved an overall two to five times acceleration due to the following improvements:

1. It projects x to x' using only $O(n^2)$ floating-point multiplications instead of solving a damped least squares problem costing $O(n^3)$ floating-point multiplications.
2. It calculates $\Delta_{\mu-1}(f)$ and $\Delta_{\mu}(f)$ using only $O(\mu n^2)$ floating-point multiplications instead of solving μ least squares problems costing $O(\mu n^3)$ floating-point multiplications.
3. It computes a suitable step length using only $O(1)$ floating-point multiplications instead of solving a linear system using $O(n^3)$ floating-point multiplications.

A cost comparison is summarized below, where the previous Algorithms 1 & 2 [?,?] are illustrated in the beginning of Section 3 and the new Algorithms 3 & 4 are given in the end of Section 4. It should be noted that both methods execute two SVDs, which use $O(\mu n^3)$ floating-point multiplications. A more detailed comparison of the cost of three improved sub-steps is provided in Section 5.

Table 1 Cost analysis of Algorithm 1 & 2 (previous) and Algorithm 3 & 4 (current)

		cost in Algorithm 1 & 2	cost in Algorithm 3 & 4	acceleration
Step 1	decompose $Df(x)$	$O(n^3)$	$O(n^3)$	none
	update x'	$O(n^3)$	$O(n^2)$	$\sim 1/n$
Step 2	decompose $Df(x')$	$O(n^3)$	$O(n^3)$	none
	calculate $\Delta_k(f)$	$O(\mu n^3)$	$O(\mu n^2)$	$\sim 1/n$
	update x''	$O(n^3)$	$O(1)$	$\sim 1/n^3$

On the other hand, the proof of the quadratic convergence in [?] is mainly based on the deflation techniques [?,?] and the symbolic-numeric reduction to geometric involutive forms [?,?]. Although the proof is rigorous, it is rather lengthy and it is difficult to derive a quantified version of the proof. Based on the closed-form basis of the local dual space (Theorem ??) and the recursive reduction of simple multiple zeros (Theorem ??), we present a new proof of the quadratic convergence without using deflation techniques, which may lead to a quantified analysis of the quadratic convergence of the improved two-step Newton's method.

Related works There are many different numeric and symbolic approaches to compute multiple zeros of polynomial systems. In [?], Rall studied the convergence property of Newton's method for singular solutions, and many modifications of Newton's method to restore the quadratic convergence for singular solutions have been proposed in [?,?,?,?,?,?,?].

In [?], Griewank constructed a bordered system from the initial system f and the singular value decomposition of the Jacobian matrix $Df(x)$ to restore the quadratic convergence of Newton's method when $Df(x)$ has corank one. The method was extended by Shen and Ypma [?,?] to the case where $Df(x)$ has arbitrary high rank deficiency.

In [?,?,?], Ojika et al. proposed a deflation method to construct a regular system to refine an approximate isolated singular solution to high accuracy. The deflation method has been further developed and generalized by Leykin, Verschelde and Zhao [?,?] for singular solutions whose Jacobian matrix has arbitrary high rank deficiency and for overdetermined polynomial systems. Furthermore, they proved that the number of deflations needed to derive a regular solution of an augmented system is strictly less than the multiplicity. Stetter [?] considered multiple zeros and zero clusters. Dayton and Zeng [?,?] proved that the depth of the local dual space is a tighter bound for the number of deflations. In [?], Lecerf gave a deflation algorithm which outputs a regular triangular system at the singular solution. In [?], Mantzaflaris and Mourrain proposed a one-step deflation method and verified a multiple root of a nearby system with a given multiplicity structure, which depends on the accuracy of the given approximate multiple root. The method is further developed by Mantzaflaris, Mourrain and Szanto in [?]. Hauenstein, Mourrain and Szanto [?,?] proposed a novel deflation method which extends their early works [?,?] to verify the existence of an isolated singular zero with a given multiplicity structure up to a given order. More recently, in [?], Giusti and Yakoubsohn proposed a new deflation sequence using the kerneling operator defined by the Schur complement of the Jacobian matrix and proved a new γ-theorem for analytic regular systems.

Dedieu and Shub [?] gave explicitly an upper bound for separating simple double zeros of analytic functions, and a numeric criterion for separating a cluster of two zeros (counting multiplicities). In [?], we gave a computable lower bound on the minimal distance between the simple multiple zero x and other zeros of f. If x is only given with limited accuracy, we proposed a numerical criterion that f is certified to have μ zeros (counting multiplicities) in a small ball around x. When $\mu = 2, 3$, we proposed a modified Newton iteration and proved the quantified quadratical convergence of the new method in [?]. Yakoubsohn [?] extended α-theory [?,?,?,?,?,?] to clusters of zeros of univariate polynomials and provided an algorithm to compute them [?]. Giusti, Lecerf, Salvy and Yakoubsohn [?] studied criteria on point estimates for locating clusters of zeros of analytic functions in univariate case and provided bounds on the diameter of the cluster of μ zeros (counting multiplicities). They proposed an algorithm based on Schröder's iteration for approximating the cluster and a stopping criterion which guarantees that the algorithm converges to the cluster quadratically. In [?], they further generalized their results to locate and approximate clusters of zeros of analytic maps of embedding dimension one via the implicit function theorem and the symbolic deflation technique.

Structure of the Paper In Section **??**, we recall some definitions and provide two characterizations of simple multiple zeros: a closed-form basis of the local dual space and a recursive reduction that plays an important role for proving the quadratic convergence of the new method. The previous two-step Newton's method is reviewed in Section **??** and an algorithmic analysis is also presented. In Section **??**, we propose an improved method and prove its quadratic convergence. Three experiments are conducted to study numerical stability of the algorithm with respect to the initially given approximate solution, the separation bound and the tolerance. In Section **??**, we compare the performance of our algorithm with the algorithm in [?] for a list of benchmark examples.

2 Characterization of Simple Multiple Zeros

Definition 1 Given a square polynomial system $f = \{f_1, \ldots, f_n\}$, where $f_i \in \mathbb{C}[X_1, \ldots, X_n]$ for $i = 1, \ldots, n$, a point $\xi \in \mathbb{C}^n$ is a simple multiple zero of $f = 0$ of multiplicity μ if it satisfies:

(1) $f(\xi) = 0$,
(2) a ball $B(\xi, r)$ of radius $r > 0$ such that $B(\xi, r) \cap f^{-1}(0) = \{\xi\}$,
(3) a generic analytic g sufficiently close to f possesses μ simple zeros in $B(\xi, r)$,
(4) $\dim \ker Df(\xi) = 1$.

The first three conditions define an isolated singular zero while the fourth condition identifies a simple multiple zero, which has some unique characterizations, including a closed-form basis of the local dual space and a recursive reduction.

2.1 A Closed-Form Basis of the Local Dual Space

Let

$$\mathbf{d}_\xi^\alpha : \mathbb{C}[X_1, \ldots, X_n] \to \mathbb{C}$$

denote the differential functional defined by

$$\mathbf{d}_\xi^\alpha(g) = \frac{1}{\alpha_1! \cdots \alpha_n!} \cdot \frac{\partial^{|\alpha|} g}{\partial X_1^{\alpha_1} \cdots \partial X_n^{\alpha_n}}(\xi), \quad \forall g \in \mathbb{C}[X_1, \ldots, X_n], \tag{3}$$

where $\alpha = [\alpha_1, \ldots, \alpha_n] \in \mathbb{N}^n$ and I_f denote the ideal generated by $f = \{f_1, \ldots, f_n\}$, then the local dual space of I_f at an isolated zero ξ is a subspace of $\mathfrak{D}_\xi = \mathrm{span}_{\mathbb{C}}\{\mathbf{d}_\xi^\alpha\}$,

$$\mathcal{D}_{f,\xi} = \{\Lambda \in \mathfrak{D}_\xi \mid \Lambda(g) = 0, \ \forall g \in I_f\}. \tag{4}$$

Consequently, if ξ is an isolated singular zero of multiplicity μ, then $\dim \mathcal{D}_{f,\xi} = \mu$. Moreover, if ξ is a simple multiple zero, then its multiplicity structure is uniformly characterized via a parametric basis of cardinality μ.

We write $d_1^{\alpha_1} \cdots d_n^{\alpha_n}$ instead of $\mathbf{d}_\xi^\alpha$, where $d_i^{\alpha_i} = \frac{1}{\alpha_i!} \cdot \frac{\partial^{\alpha_i}}{\partial X_i^{\alpha_i}}$ and let $\Psi_i : \mathfrak{D}_\xi \to \mathfrak{D}_\xi$ be the morphism that satisfies $\Psi_i(d_1^{\alpha_1} \cdots d_n^{\alpha_n}) = d_i^{\alpha_i+1} \cdots d_n^{\alpha_n}$ if $\alpha_1 = \cdots = \alpha_{i-1} = 0$ and 0 otherwise for $i = 1, \ldots, n$.

Proposition 1 *[?, Theorem 3.4] For a square polynomial system f associated with a simple multiple zero ξ of multiplicity μ, with $(\mu - 1)n$ parameters $a_{k,i}$, where $k = 1, \ldots, \mu-1$ and $i = 1, \ldots, n$, the set $\{\Lambda_0 = 1, \Lambda_1 = a_{1,1}d_1 + \cdots + a_{1,n}d_n, \Lambda_2, \ldots, \Lambda_{\mu-1}\}$ recursively formulated by*

$$\Delta_k = \sum_{i=1}^{n} \sum_{j=1}^{k-1} a_{j,i} \Psi_i(\Lambda_{k-j}), \tag{5}$$

$$\Lambda_k = \Delta_k + \sum_{i=1}^{n} a_{k,i} d_i, \tag{6}$$

for $k = 2, \ldots, \mu - 1$ is a parametric basis of $\mathcal{D}_{f,\xi}$.

Obviously, constraints $\Lambda_k(f) = 0$ are not sufficient to guarantee a unique basis of $\mathcal{D}_{f,\xi}$ because $\dim\ker Df(\xi) = 1$. Therefore, some normalization constraint is required for deciding $a_{k,i}$ uniquely. Let

$$Df(\xi) = U \cdot \operatorname{diag}(\sigma_1, \ldots, \sigma_{n-1}, 0) \cdot V^*$$

be the Singular Value Decomposition (SVD) of $Df(\xi)$, where $\sigma_1 \geq \cdots \geq \sigma_{n-1} > 0$, $U = (u_1, \ldots, u_n)$ and $V = (v_1, \ldots, v_n)$ are unitary matrices, V^* denote the Hermitian (conjugate) transpose of V.

Theorem 1 *Consider the normalization constraint*

$$a_{1,1}^* a_{1,1} + \cdots + a_{1,n}^* a_{1,n} = 1,\ a_{1,1}^* a_{k,1} + \cdots + a_{1,n}^* a_{k,n} = 0\ (k = 2, \ldots, \mu - 1), \quad (7)$$

then a unique basis of $\mathcal{D}_{f,\xi}$ is determined incrementally by ***the closed form***

$$\begin{pmatrix} a_{1,1} \\ a_{1,2} \\ \vdots \\ a_{1,n} \end{pmatrix} = v_n, \quad \begin{pmatrix} a_{k,1} \\ a_{k,2} \\ \vdots \\ a_{k,n} \end{pmatrix} = -\sum_{i=1}^{n-1} \frac{1}{\sigma_i} v_i {u_i}^* \Delta_k(f)\ (k = 2, \ldots, \mu - 1). \quad (8)$$

Proof Since V is a unitary matrix, its columns satisfy $v_n^* v_n = 1$, and $v_n^* v_i = 0$ for $i = 1, 2, \ldots, n-1$. Therefore, the vectors $(a_{k,1}, a_{k,2}, \ldots, a_{k,n})^T$ in (**??**) satisfy (**??**) for $k = 1, \ldots, \mu - 1$.

According to the normalization constraint (**??**), we can rewrite

$$(a_{k,1}, a_{k,2}, \ldots, a_{k,n})^T = \lambda_1 v_1 + \lambda_2 v_2 + \cdots + \lambda_{n-1} v_{n-1}.$$

Then, the constraint $\Lambda_k(f) = 0$ is equivalent to

$$\begin{aligned} 0 &= U^* \Lambda_k(f) \\ &= U^*[\Delta_k(f) + a_{k,1} d_1(f) + \cdots + a_{k,n} d_n(f)] \\ &= U^* \Delta_k(f) + U^* Df(\xi)(a_{k,1}, a_{k,2}, \ldots, a_{k,n})^T \\ &= U^* \Delta_k(f) + \operatorname{diag}(\sigma_1, \ldots, \sigma_{n-1}, 0) V^* V (\lambda_1, \lambda_2, \ldots, \lambda_{n-1}, 0)^T \\ &= \begin{pmatrix} u_1^* \Delta_k(f) + \sigma_1 \lambda_1 \\ u_2^* \Delta_k(f) + \sigma_2 \lambda_2 \\ \vdots \\ u_{n-1}^* \Delta_k(f) + \sigma_{n-1} \lambda_{n-1} \\ u_n^* \Delta_k(f) \end{pmatrix}. \end{aligned} \quad (9)$$

Therefore, if $u_n^* \Delta_k(f) = 0$, then

$$\lambda_1 = -\frac{1}{\sigma_1} u_1^* \Delta_k(f), \lambda_2 = -\frac{1}{\sigma_2} u_2^* \Delta_k(f), \ldots, \lambda_{n-1} = -\frac{1}{\sigma_{n-1}} u_{n-1}^* \Delta_k(f)$$

gives a unique solution (**??**) of $\Lambda_k(f) = 0$ satisfying the normalization constraint (**??**); otherwise, the procedure is terminated and k is the multiplicity of the simple multiple zero ξ, i.e. $\mu = k$.

Remark 1 Since $\dim\ker Df(\xi)=1$, without loss of generality, one can assume the first column of $Df(\xi)$ can be written as a linear combination of other $n-1$ columns of $Df(\xi)$. Therefore, in [**?**], the following normalization constraint

$$a_{1,1}=1,\ a_{k,1}=0\ (k=2,\ldots,\mu-1). \tag{10}$$

is used to determine a unique basis of $\mathcal{D}_{f,\xi}$ by solving $\mu-1$ linear systems incrementally

$$Df(\xi)\begin{pmatrix}1\\a_{1,2}\\\vdots\\a_{1,n}\end{pmatrix}=0,\ Df(\xi)\begin{pmatrix}0\\a_{k,2}\\\vdots\\a_{k,n}\end{pmatrix}=-\Delta_k(f)\ (k=2,\ldots,\mu-1). \tag{11}$$

In Section **??**, we will demonstrate that replacing the previous constraint (**??**) by (**??**) and its corresponding closed-form solution (**??**) will save a significant amount of computational effort for calculating an approximate basis of $\mathcal{D}_{f,\xi}$ when an approximate zero x near to ξ is given.

2.2 Recursive Reduction

Suppose ξ is a simple multiple zero of $f(X)=0$ of multiplicity μ and $Df(\xi)=U\cdot\operatorname{diag}(\sigma_1,\ldots,\sigma_{n-1},0)\cdot V^*$ is the SVD of $Df(\xi)$, where $\sigma_1\geq\cdots\geq\sigma_{n-1}>0$, $U=(u_1,\ldots,u_n)$ and $V=(v_1,\ldots,v_n)$ are unitary matrices.

Since $\dim\ker Df(\xi)=1$, inspired by the implicit univariate reduction in [**?**] and formula (3.18) in [**?**, Lemma 3.13], we show the existence of the following recursive reduction of the polynomial $u_n^*f(X)$ at ξ. It is very similar to expanding a univariate polynomial at a multiple zero of multiplicity μ.

Theorem 2 *Suppose ξ is a simple multiple zero of $f(X)=0$ of multiplicity μ, then there exists a* ***recursive reduction***

$$u_n^*f(X)=u_n^*\Delta_\mu(f)\left[v_n^*(X-\xi)\right]^\mu+\sum_{|\beta|>\mu}c_\beta(X-\xi)^\beta+\sum_{i=1}^{n-1}h_i(X)(u_i^*f(X)), \tag{12}$$

*where $\Delta_\mu\in\mathfrak{D}_{f,\xi}$ is the unique differential functional satisfying (**??**),(**??**) and (**??**), $c_\beta\in\mathbb{C}$ for $\beta\in\mathbb{N}^n$, and $h_i\in\mathbb{C}[X]$ satisfying $h_i(\xi)=0$.*

Proof Since $u_i^*Df(\xi)=\sigma_i v_i^*$, Taylor's expansion of f at ξ indicates that

$$u_i^*f(X)=\sigma_i v_i^*(X-\xi)+\sum_{|\beta|\geq 2}b_{i,\beta}[V^*(X-\xi)]^\beta,\ (i=1,\ldots,n-1), \tag{13}$$

where

$$[V^*(X-\xi)]^\beta=[v_1^*(X-\xi)]^{\beta_1}\cdots[v_n^*(X-\xi)]^{\beta_n}$$

for $\beta=[\beta_1,\ldots,\beta_n]\in\mathbb{N}^n$ and $b_{i,\beta}\in\mathbb{C}$,

$$u_n^*f(X)=\sum_{|\gamma|\geq 2}b_{n,\gamma}[V^*(X-\xi)]^\gamma, \tag{14}$$

for $\gamma = [\gamma_1, \ldots, \gamma_n] \in \mathbb{N}^n$ and $b_{n,\gamma} \in \mathbb{C}$.

For $|\gamma| = \gamma_1 + \cdots + \gamma_n \geq 2$, if $\gamma_i > 0$ $(i = 1, \ldots, n-1)$, by (**??**), we can substitute

$$v_i^*(X-\xi) = \frac{1}{\sigma_i} u_i^* f(X) - \frac{1}{\sigma_i} \sum_{|\beta| \geq 2} b_{i,\beta} [V^*(X-\xi)]^{\beta}, \ (i = 1, \ldots, n-1)$$

into only one corresponding factor of $b_{n,\gamma}[V^*(X-\xi)]^{\gamma}$ to derive

$$\begin{aligned} b_{n,\gamma}[V^*(X-\xi)]^{\gamma} &= b_{n,\gamma}[V^*(X-\xi)]^{\gamma - e_i} \left(\frac{1}{\sigma_i} u_i^* f(X) - \frac{1}{\sigma_i} \sum_{|\beta| \geq 2} b_{i,\beta} [V^*(X-\xi)]^{\beta} \right) \\ &= \frac{b_{n,\gamma}}{\sigma_i} [V^*(X-\xi)]^{\gamma - e_i} (u_i^* f(X)) - \sum_{|\beta| \geq 2} \frac{b_{n,\gamma} b_{i,\beta}}{\sigma_i} [V^*(X-\xi)]^{\gamma + \beta - e_i} \end{aligned}$$

where e_i is the i-th unit vector and $|\gamma + \beta - e_i| = |\gamma| + |\beta| - 1 > |\gamma|$.

Then, recursively perform the above substitution for $|\gamma| = 2, \ldots, \mu$ where $\gamma_i > 0$ $(i = 1, \ldots, n-1)$ in (**??**), we derive

$$u_n^* f(X) = \sum_{l=2}^{\mu} c_l \left[v_n^*(X-\xi) \right]^l + \sum_{|\beta| > \mu} c_{\beta} (X-\xi)^{\beta} + \sum_{i=1}^{n-1} h_i(X)(u_i^* f(X)), \tag{15}$$

where $c_l \in \mathbb{C}, c_{\beta} \in \mathbb{C}$, $(X-\xi)^{\beta} = (X_1 - \xi_1)^{\beta_1} \cdots (X_n - \xi_n)^{\beta_n}$ for $\beta \in \mathbb{N}^n$ and $h_i \in \mathbb{C}[X]$ satisfying $h_i(\xi) = 0$.

Suppose $\Delta_k \in \mathfrak{D}_{\xi}$ are those differential functionals satisfying (**??**),(**??**) and (**??**), then we can show that they also satisfy the following conditions:

$$\Delta_k \left((X-\xi)^{\beta} \right) = 0, \quad \text{if } |\beta| > k, \tag{16}$$

$$\Delta_k \left((X-\xi)^{\beta} f(X) \right) = 0, \quad \text{if } |\beta| > 0, \tag{17}$$

$$\Delta_k \left(\left[v_n^*(X-\xi) \right]^l \right) = \begin{cases} 1, \text{ if } k = l, \\ 0, \text{ otherwise,} \end{cases} \tag{18}$$

See Appendix for the proofs of the above conditions. Therefore, by applying the above Δ_k to both sides of (**??**), we obtain

$$\Delta_k \left(u_n^* f(X) \right) = u_n^* \Delta_k(f) = c_k, \ (k = 2, \ldots, \mu).$$

According to (**??**), we have $c_k = u_n^* \Delta_k(f) = 0$ $(k = 2, \ldots, \mu)$ and $c_{\mu} = u_n^* \Delta_{\mu}(f) \neq 0$, so we conclude the recursive reduction (**??**).

In Section **??**, we will give an elegant proof of the quadratic convergence of the new proposed two-step Newton's method based on the recursive reduction (**??**).

3 Review of Two-step Newton's Method

We review the two-step Newton's method for computing simple multiple zeros under Assumption 1 in [**?**].

Assumption 1 *Suppose we are given a polynomial system $f \in \mathbb{C}[X_1, \ldots, X_n]^n$ having a simple multiple zero $\xi \in \mathbb{C}^n$ of multiplicity μ (not known prior) and a point $x \in \mathbb{C}^n$ satisfying $\|x - \xi\| = \epsilon$ where $1 > \epsilon > 0$ can be sufficiently small.*

Using Taylor's expansion of $f(x)$ at ξ, it is easy to show that Assumption **??** implies that $\|f(x)\| = O(\epsilon)$. Moreover, consider Taylor's expansion of $Df(x)$ at ξ,

$$Df(x) = Df(\xi) + D^2 f(\xi)(x - \xi) + \sum_{k \geq 2} \frac{D^{k+1} f(\xi)}{k!} (x - \xi)^k.$$

then it implies

$$\|Df(x) - Df(\xi)\| \leq \|D^2 f(\xi)\| \epsilon + \sum_{k \geq 2} \frac{\|D^{k+1} f(\xi)\|}{k!} \epsilon^k = O(\epsilon).$$

Let $\sigma_i(x)$ ($\sigma_i(\xi)$) denote the singular values of $Df(x)$ ($Df(\xi)$) in descending order, then according to Weyl's theorem [**?**],

$$|\sigma_i(x) - \sigma_i(\xi)| \leq \|Df(x) - Df(\xi)\| = O(\epsilon), \text{ for } i = 1, \ldots, n,$$

which concludes

$$\sigma_n(x) \leq \sigma_n(\xi) + O(\epsilon) = O(\epsilon), \sigma_i(x) \geq \sigma_i(\xi) - O(\epsilon) \geq c > 0, \tag{19}$$

for $i = 1, \ldots, n - 1$ where $c = \sigma_{n-1}(\xi)/2$, since $\sigma_n(\xi) = 0$, $\sigma_{n-1}(\xi) > 0$ and ϵ can be sufficiently small.

The above discussion suggests a numerical identification for an approximation x of a simple multiple zero ξ. Namely, if $\|f(x)\| < \tau$ and $\sigma_n < \tau \ll \sigma_{n-1}$ for a specified tolerance τ, then x is identified as an approximation of a simple multiple zero ξ of f.

Given a successfully identified approximation x, the task of this paper is to refine it to a more accurate approximation x'' satisfying

$$\|x'' - \xi\| = O(\|x - \xi\|^2).$$

We review the algorithms MultipleRootRefinerBreadthOne (MRRB1) in [**?**] and ApproximateMultiplicityStructure (AMS) in [**?**] for refining approximate simple multiple zeros to higher accuracies.

Algorithm 1 MultipleRootRefinerBreadthOne [**?**]

Input:

f: a polynomial system associated with a simple multiple zero ξ;
x: an identified approximation of ξ for a specified tolerance τ;

Output:

x'': the refined approximation of ξ;

1: Calculate $Df(x) = U \cdot \mathrm{diag}(\sigma_1, \ldots, \sigma_n) \cdot V^*$,
where $U = (u_1, \ldots, u_n)$, $V = (v_1, \ldots, v_n)$ are unitary and $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$.
2: Refine x to $x' = x + \mathbf{LinearSolver}(Df(x)^* Df(x) + \sigma_n I_n, -Df(x)^* f(x))$,
where $\mathbf{LinearSolver}(A, b)$ is the solution of the linear system $AX = b$.
3: Calculate $Df(x') = U' \cdot \mathrm{diag}(\sigma'_1, \ldots, \sigma'_n) \cdot V'^*$,
where $U' = (u'_1, \ldots, u'_n)$, $V' = (v'_1, \ldots, v'_n)$ are unitary and $\sigma'_1 \geq \cdots \geq \sigma'_n \geq 0$.
4: Refine x' to $x'' = x' + \delta v'_n$,

$$\text{where } \delta = \pi_{X_1}\left(\mathbf{LinearSolver}\left(\left[\Delta_\mu(f), \frac{\partial f(x)}{\partial X_2}, \ldots, \frac{\partial f(x)}{\partial X_n}\right], -\Lambda_{\mu-1}(f)\right)\right)/\mu,$$

$\pi_{X_1}(\cdot)$ is the first component of a vector, $\Lambda_{\mu-1}$ and Δ_μ are by-products of AMS.

Algorithm 2 ApproximateMultiplicityStructure [**?**]

Input:

f: a polynomial system associated with a simple multiple zero ξ;
x: an identified approximation of ξ for a specified tolerance τ;

Output:

μ: an estimation of the multiplicity of ξ;
$a_{k,i}$: an estimation of $(\mu - 1)n$ parameters for a basis of $\mathcal{D}_{f,\xi}$ approximately;

1: Let v_n be the last right singular vector of $Df(x)$ and suppose $v_{n,1} \geq v_{n,i}$, set

$$a_{1,1} = 1, \; a_{1,i} = v_{n,i}/v_{n,1} \; (i = 2, \ldots, n),$$

2: From $k = 2$, incrementally set $a_{k,1} = 0$ and calculate $a_{k,i}$ $(i = 2, \ldots, n)$ by

$$\left(a_{k,2}, \ldots, a_{k,n}\right)^T = \mathbf{LeastSquares}\left(\left[\frac{\partial f(x)}{\partial X_2}, \ldots, \frac{\partial f(x)}{\partial X_n}\right], -\Delta_k(f)\right), \tag{20}$$

where $\mathbf{LeastSquares}(A, b)$ is the least squares solution of $\min \|AX - b\|_2$ and Δ_k is formulated by (**??**,**??**). If $\|(\Delta_k + a_{k,2}d_2 + \cdots + a_{k,n}d_n)(f)\| < \tau$, then set $\Lambda_k = \Delta_k + a_{k,2}d_2 + \cdots + a_{k,n}d_n$ and repeat the step; otherwise, return $\mu = k$.

Figure **??** shows a sketch of Algorithm 1 on the following toy example.

Example 1 [**?**] Consider $f = \{X^2 + Y - 3, X + 0.125\ Y^2 - 1.5\}$ associated with $\xi = (1, 2)$ and an identified approximation $x = (1.01, 2.01)$.

Fig. 1 The red circle denotes the simple multiple zero $(1, 2)$, green and blue circles denote input and output approximations of each step which are linked by red dash-dotted lines. The solid lines are contours of $\log\sqrt{(X^2 + Y - 3)^2 + (X + 0.125\ Y^2 - 1.5)^2}$ and the dashed circles are contours of $\log\sqrt{(X - 1)^2 + (Y - 2)^2}$.

Figure **??** shows the methodology of the two-step Newton's method for computing simple multiple zeros:

1. the first step of the refinement projects x to x' such that $x' - \xi$ approximately inside $\mathrm{span}_{\mathbb{C}}\{v'_n\}$, which approximately coincides with $\ker Df(\xi)$. Outwardly, x' satisfies
$$\|x' - \xi\| = O(\epsilon),\ \|f(x')\| = O(\epsilon^2). \tag{21}$$
2. the second step of the refinement finds x'' by following the direction v'_n from x' with the step length δ, which satisfies
$$|\delta + {v'_n}^*(x' - \xi)| = O(\epsilon^2). \tag{22}$$

In fact, (**??**) and (**??**) are essential to prove the quadratic convergence of Algorithm 1:

Theorem 3 (Theorem 3.2 [**?**] & **Theorem 3.6** [**?**]**)** *Suppose y is the solution of the linear system*
$$(Df(x)^* Df(x) + \sigma_n I_n)X = -Df(x)^* f(x), \tag{23}$$
*then the projection $x' = x + y$ satisfies (**??**). Consequently, x' is already of quadratic convergence in $(\ker Df(\xi))^\perp$ approximately, namely,*
$$|{v'_i}^*(x' - \xi)| = O(\epsilon^2),\ i = 1, \ldots, n-1. \tag{24}$$

Theorem 4 (Theorem 3.16 [**?**]**)** *Suppose z is the solution of the linear system*
$$\left[\Delta_\mu(f), \frac{\partial f(x)}{\partial X_2}, \ldots, \frac{\partial f(x)}{\partial X_n}\right] X = -\Lambda_{\mu-1}(f), \tag{25}$$
*then the step length $\delta = \pi_{X_1}(z)/\mu$ satisfies (**??**). Consequently, the approximate zero $x'' = x' + \delta v'_n$ is of quadratic convergence, namely,*
$$\|x'' - \xi\| = O(\epsilon^2). \tag{26}$$

Remark 2 Two facts worth to be noted. First, for any x' satisfying (**??**),
$$\begin{aligned}
&\|\mathrm{diag}(\sigma'_1, \ldots, \sigma'_n) \cdot {V'}^*(x' - \xi)\| = \|U^* Df(x')(x' - \xi)\| \\
= &\left\| U^* f(x') - U^* f(\xi) + U^* \sum_{k \geq 2} \frac{D^k f(x')}{k!}(\xi - x')^k \right\| \leq \|U^* f(x')\| + O(\epsilon^2) = O(\epsilon^2).
\end{aligned}$$
Therefore, the condition (**??**) is satisfied. Second, for any δ satisfying (**??**), we have
$$\begin{aligned}
&\|x'' - \xi\|^2 = \|{V'}^*(x'' - \xi)\|^2 \\
= &\sum_{i=1}^{n-1} |{v'_i}^*(x' - \xi)|^2 + |\delta + {v'_n}^*(x' - \xi)|^2 = (O(\epsilon^2))^2.
\end{aligned}$$
Hence, the condition (**??**) is also satisfied.

In this paper, instead of solving linear systems (**??**) and (**??**), we propose **LinearSolver**-free estimations for computing more efficiently the projection x' and the step length δ satisfying (**??**) and (**??**); instead of solving least-squares problems (**??**), we propose **LeastSquares**-free estimations for computing more efficiently the multiplicity μ and the values of parameters $a_{k,i}$ for an approximate multiplicity structure. Both accelerations attribute to the two new characterizations of simple multiple zeros, namely, the closed-form basis of $\mathcal{D}_{f,\xi}$ (**??**) and the recursive reduction (**??**).

4 New Approach

Given a polynomial system f associated with a simple multiple zero ξ, suppose x is identified as an approximation of ξ for a specified tolerance τ, we propose an accelerated two-step Newton's method which retains the quadratic convergence under Assumption **??**.

4.1 A LinearSolver-Free Estimation for x'

Suppose x is an approximation of ξ satisfying $\|x-\xi\| = \epsilon$. Let $Df(x) = \sum_{i=1}^{n} \sigma_i u_i v_i^*$ denote the SVD of $Df(x)$, where $u_i^* u_j = v_i^* v_j = 0$ if $i \neq j$, $u_i^* u_i = v_i^* v_i = 1$ for $1 \leq i \leq n$ and $1 \leq j \leq n$.

The following theorem provides a **closed-form** estimation for the projection x' without solving any linear system comparing to Theorem **??**.

Theorem 5 *Suppose*

$$y = -\sum_{i=1}^{n-1} \frac{1}{\sigma_i} v_i u_i^* f(x), \tag{27}$$

*then the projection $x' = x + y$ satisfies (**??**). Consequently, x' is already of quadratic convergence in $(\ker Df(\xi))^{\perp}$ approximately, namely, (**??**) holds.*

Proof According to (**??**) and $\|x - \xi\| = \epsilon$, we have $\|f(x)\| = O(\epsilon)$ and

$$\|x' - \xi\| = \|(x - \xi) + y\| \leq \epsilon + \sum_{i=1}^{n-1} \frac{1}{\sigma_i} \|v_i\| \|u_i^*\| \|f(x)\| = O(\epsilon),$$

which concludes the first part of (**??**).

For the second part, let us consider Taylor's expansion of $f(\xi)$ at x and multiply U^* to both sides of the equality,

$$0 = U^* f(\xi) = U^* f(x) + U^* Df(x)(\xi - x) + U^* \sum_{k \geq 2} \frac{D^k f(x)}{k!} (\xi - x)^k.$$

It implies

$$\|U^* f(x) + U^* Df(x)(\xi - x)\| \leq \sum_{k \geq 2} \frac{\|D^k f(\xi)\|}{k!} \epsilon^k = O(\epsilon^2),$$

which concludes

$$|u_n^* f(x)| \leq |\sigma_n v_n^* (x - \xi)| + O(\epsilon^2) = O(\epsilon^2). \tag{28}$$

Similarly, let us consider Taylor's expansion of $f(x')$ at x and again multiply U^* to both sides of the equality,

$$U^* f(x') = U^* f(x) + U^* Df(x) y + U^* \sum_{k \geq 2} \frac{D^k f(x)}{k!} y^k,$$

which implies

$$\|f(x')\| = \|U^* f(x')\| \le \|U^* f(x) + U^* Df(x) y\| + O(\epsilon^2).$$

Now, we calculate $U^* f(x) + U^* Df(x)y$ component-wisely. For $i = 1, \ldots, n-1$,

$$\left|u_i^* f(x) + u_i^* Df(x)y\right| = \left|u_i^* f(x) - \sigma_i v_i^* \cdot \sum_{j=1}^{n-1} \frac{1}{\sigma_j} v_j u_j^* f(x)\right| = \left|u_i^* f(x) - u_i^* f(x)\right| = 0.$$

For the last component, according to (**??**),

$$\left|u_n^* f(x) + u_n^* Df(x)y\right| = \left|u_n^* f(x) - \sigma_n v_n^* \cdot \sum_{j=1}^{n-1} \frac{1}{\sigma_j} v_j u_j^* f(x)\right| = \left|u_n^* f(x) - 0\right| = O(\epsilon^2).$$

Therefore, we derive $\|f(x')\| = O(\epsilon^2)$, and the second part of (**??**) holds. Consequently, according to Remark **??**, (**??**) holds too.

4.2 A LeastSquares-Free Estimation for $a_{k,i}$

Let

$$Df(\xi) = \sum_{i=1}^{n} \hat{\sigma}_i \hat{u}_i \hat{v}_i^*$$

be the SVD of $Df(\xi)$, where $\hat{u}_i^* \hat{u}_j = \hat{v}_i^* \hat{v}_j = 0$ if $i \neq j$, $\hat{u}_i^* \hat{u}_i = \hat{v}_i^* \hat{v}_i = 1$ for $1 \le i \le n$ and $1 \le j \le n$. By Theorem **??**, a unique basis of $\mathcal{D}_{f,\xi}$ can be determined by assigning the values of $(\mu - 1)n$ parameters to

$$\hat{\mathbf{a}}_1 = \begin{pmatrix} \hat{a}_{1,1} \\ \hat{a}_{1,2} \\ \vdots \\ \hat{a}_{1,n} \end{pmatrix} = \hat{v}_n, \ \hat{\mathbf{a}}_k = \begin{pmatrix} \hat{a}_{k,1} \\ \hat{a}_{k,2} \\ \vdots \\ \hat{a}_{k,n} \end{pmatrix} = -\sum_{i=1}^{n-1} \frac{1}{\hat{\sigma}_i} \hat{v}_i \hat{u}_i^* \hat{\Delta}_k(f), \tag{29}$$

for $k = 2, \ldots, \mu - 1$, where $\hat{\Delta}_k$ is incrementally formulated by (**??**) and (**??**).

Suppose x' is now an approximation of ξ satisfying $\|x' - \xi\| = O(\epsilon)$. Let

$$Df(x') = \sum_{i=1}^{n} \sigma_i' u_i' {v_i'}^*$$

denote the SVD of $Df(x')$, where ${u_i'}^* u_j' = {v_i'}^* v_j' = 0$ if $i \neq j$, ${u_i'}^* u_i = {v_i'}^* v_i = 1$ for $1 \le i \le n$ and $1 \le j \le n$. Let $\Delta_2, \ldots, \Delta_{\mu-1}, \Delta_\mu \in \mathfrak{D}_{x'}$ be the $\mu - 1$ linear differential functionals, which are incrementally formulated by (**??**) and (**??**) with

$$\mathbf{a}_1' = \begin{pmatrix} a_{1,1}' \\ a_{1,2}' \\ \vdots \\ a_{1,n}' \end{pmatrix} = v_n', \ \mathbf{a}_k' = \begin{pmatrix} a_{k,1}' \\ a_{k,2}' \\ \vdots \\ a_{k,n}' \end{pmatrix} = -\sum_{i=1}^{n-1} \frac{1}{\sigma_i'} v_i' {u_i'}^* \Delta_k(f), \tag{30}$$

for $k = 2, \ldots, \mu - 1$. We show below that the above defined $\mathbf{a}'_k$ is the optimal solution of the following least squares problem:

$$\begin{aligned} &\min_{a_{k,1},\cdots,a_{k,n}} && \|\Delta_k(f) + Df(x')(a_{k,1}, \ldots, a_{k,n})^T\| \\ &\text{subject to} && {v'_n}^*(a_{k,1}, \ldots, a_{k,n})^T = 0. \end{aligned} \tag{31}$$

Let

$$(a_{k,1}, a_{k,2}, \ldots, a_{k,n})^T = \lambda_{k,1}v'_1 + \cdots + \lambda_{k,n-1}v'_{n-1},$$

then we have

$$\begin{aligned} \|\Delta_k(f) + Df(x')(a_{k,1}, \ldots, a_{k,n})^T\| &= \left\|{U'}^*\Delta_k(f) + {U'}^*Df(x')(a_{k,1}, a_{k,2}, \ldots, a_{k,n})^T\right\| \\ &= \left\|\begin{pmatrix} {u'_1}^*\Delta_k(f) + \sigma'_1\lambda_{k,1} \\ \vdots \\ {u'_{n-1}}^*\Delta_k(f) + \sigma'_{n-1}\lambda_{k,n-1} \\ {u'_n}^*\Delta_k(f) \end{pmatrix}\right\| \\ &\geq |{u'_n}^*\Delta_k(f)|. \end{aligned}$$

The equality holds when

$$\lambda_{k,1} = -\frac{{u'}^*_1\Delta_k(f)}{\sigma'_1}, \ldots, \lambda_{k,n-1} = -\frac{{u'}^*_{n-1}\Delta_k(f)}{\sigma'_{n-1}}. \tag{32}$$

Moreover, for $\mathbf{a}'_k = -\sum_{i=1}^{n-1} \frac{1}{\sigma'_i} v'_i {u'_i}^*\Delta_k(f)$, we have

$${v'_n}^*(a'_{k,1}, \ldots, a'_{k,n})^T = 0.$$

Therefore, (**??**) is the optimal solution of (**??**) and $|{u'_n}^*\Delta_k(f)|$ is the optimal value.

The following theorem shows that the value (**??**) of parameters $a_{k,i}$ is a **closed-form** approximation of (**??**) without solving any least squares problems. Moreover, the errors of $\|\mathbf{a}'_k - \hat{\mathbf{a}}_k\|$ are bounded by $O(\epsilon)$ for $k = 1, \ldots, \mu - 1$.

Theorem 6 *Suppose μ is the multiplicity of ξ, then*

$$|{v'_n}^*\hat{\mathbf{a}}_1| = 1 - O(\epsilon),\ \|\mathbf{a}'_1 - \hat{\mathbf{a}}_1\| = O(\epsilon). \tag{33}$$

$$|{v'_n}^*\hat{\mathbf{a}}_k| = O(\epsilon),\ \|\mathbf{a}'_k - \hat{\mathbf{a}}_k\| = O(\epsilon),\ |{u'_n}^*\Delta_k(f)| = O(\epsilon),\ (k = 2, \ldots, \mu - 1), \tag{34}$$

$$|{u'_n}^*\Delta_\mu(f) - \hat{u}_n^*\hat{\Delta}_\mu(f)| = O(\epsilon). \tag{35}$$

Proof For $k = 1$, suppose $\hat{\mathbf{a}}_1 = \hat{v}_n = \hat{\lambda}_{1,1}v'_1 + \cdots + \hat{\lambda}_{1,n-1}v'_{n-1} + \hat{\lambda}_{1,n}v'_n$, then

$$\mathbf{a}'_1 - \hat{\mathbf{a}}_1 = V'\begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} - V'\begin{pmatrix} \hat{\lambda}_{1,1} \\ \vdots \\ \hat{\lambda}_{1,n-1} \\ \hat{\lambda}_{1,n} \end{pmatrix} = V'\begin{pmatrix} -\hat{\lambda}_{1,1} \\ \vdots \\ -\hat{\lambda}_{1,n-1} \\ 1 - \hat{\lambda}_{1,n} \end{pmatrix},$$

where $V' = (v'_1, \ldots, v'_n)$ is a unitary matrix. By the definition of SVD,

$$\begin{aligned}\|Df(x')(\mathbf{a}'_1 - \hat{\mathbf{a}}_1)\| &\le \|Df(x')v'_n\| + \|(Df(\xi) - Df(x'))\hat{v}_n\| + \|Df(\xi)\hat{v}_n\| \\ &\le \sigma'_n + \|Df(x') - Df(\xi)\| + \hat{\sigma}_n \\ &\le O(\epsilon) + O(\epsilon) + 0 = O(\epsilon).\end{aligned}$$

On the other hand, since $U' = (u'_1, \ldots, u'_n)$ is a unitary matrix,

$$\begin{aligned}\|Df(x')(\mathbf{a}'_1 - \hat{\mathbf{a}}_1)\| &= \left\| U'\mathrm{diag}(\sigma'_1, \ldots, \sigma'_{n-1}, \sigma'_n){V'}^*V' \begin{pmatrix} -\hat{\lambda}_{1,1} \\ \vdots \\ -\hat{\lambda}_{1,n-1} \\ 1 - \hat{\lambda}_{1,n} \end{pmatrix} \right\| \\ &= \left\| \begin{pmatrix} -\sigma'_1\hat{\lambda}_{1,1} \\ \vdots \\ -\sigma'_{n-1}\hat{\lambda}_{1,n-1} \\ \sigma'_n(1 - \hat{\lambda}_{1,n}) \end{pmatrix} \right\|.\end{aligned}$$

As $\|Df(x')(\mathbf{a}'_1 - \hat{\mathbf{a}}_1)\| = O(\epsilon)$ and $\sigma'_i = O(1)$ for $i = 1, \ldots, n-1$, we get

$$|\hat{\lambda}_{1,1}| = O(\epsilon), \ldots, |\hat{\lambda}_{1,n-1}| = O(\epsilon).$$

Moreover, because $\sum_{i=1}^n |\hat{\lambda}_{1,i}|^2 = \|\hat{v}_n\|^2 = 1$, we get

$$|\hat{\lambda}_{1,n}| = 1 - O(\epsilon).$$

Hence, $|{v'_n}^*\hat{\mathbf{a}}_1| = |{v'_n}^*\hat{v}_n| = |\hat{\lambda}_{1,n}| = 1 - O(\epsilon)$, which proves the first part of (**??**).

Given an SVD $Df(\xi) = \sum_{i=1}^n \hat{\sigma}_i\hat{u}_i\hat{v}_i^*$, where $\hat{u}_i^*\hat{u}_j = \hat{v}_i^*\hat{v}_j = 0$ if $i \neq j$ and $\hat{u}_i^*\hat{u}_i = \hat{v}_i^*\hat{v}_i = 1$, then replacing $\hat{u}_n$ and $\hat{v}_n$ by $z \cdot \hat{u}_n$ and $z \cdot \hat{v}_n$ where $z \in \mathbb{C}$ satisfying $|z| = 1$ will still validate an SVD of $Df(\xi)$. Therefore, without loss of generality, we can always assume that $\hat{\mathbf{a}}_1 = \hat{v}_n \in \ker Df(\xi)$ satisfies $\|\mathbf{a}'_1 - \hat{\mathbf{a}}_1\| = O(\epsilon)$. Otherwise, we replace $\hat{v}_n$ by $z_0^* \cdot \hat{v}_n$ where $z_0 = \mathrm{argmin}_{|z|=1}|z - {v'_n}^*\hat{v}_n|$. Since $|{v'_n}^*\hat{v}_n| = 1 - O(\epsilon)$, by triangular inequality, we have $|z_0 - {v'_n}^*\hat{v}_n| = |z_0| - |{v'_n}^*\hat{v}_n| = O(\epsilon)$. At this moment,

$$\|\mathbf{a}'_1 - \hat{\mathbf{a}}_1\| = \left\| V' \begin{pmatrix} -z_0^*\hat{\lambda}_{1,1} \\ \vdots \\ -z_0^*\hat{\lambda}_{1,n-1} \\ 1 - z_0^*\hat{\lambda}_{1,n} \end{pmatrix} \right\| = \left\| z_0^* \begin{pmatrix} -\hat{\lambda}_{1,1} \\ \vdots \\ -\hat{\lambda}_{1,n-1} \\ z_0 - \hat{\lambda}_{1,n} \end{pmatrix} \right\| = O(\epsilon),$$

since $|\hat{\lambda}_{1,1}| = O(\epsilon), \ldots, |\hat{\lambda}_{1,n-1}| = O(\epsilon)$, $\hat{\lambda}_{1,n} = {v'_n}^*\hat{v}_n$. Hence, the second part of (**??**) is proved. Similarly, we can always assume the last left singular vector $\hat{u}_n \in \ker Df(\xi)^*$ satisfying $\|u'_n - \hat{u}_n\| = O(\epsilon)$.

Assume $\|\mathbf{a}'_i - \hat{\mathbf{a}}_i\| = O(\epsilon)$ is true for $i = 1, \ldots, k-1$, we prove below that $\|\mathbf{a}'_k - \hat{\mathbf{a}}_k\| = O(\epsilon)$.

First, since the constructions of Δ_k and $\hat{\Delta}_k$ depend only on the values of $\mathbf{a}'_i$ and $\hat{\mathbf{a}}_i$, $\|\mathbf{a}'_i - \hat{\mathbf{a}}_i\| = O(\epsilon)$ for $i = 1, \ldots, k-1$ and $\|x' - \xi\| = O(\epsilon)$, it is easy to verify that

$$\left\|\Delta_k(f) - \hat{\Delta}_k(f)\right\| = O(\epsilon),$$

Consequently, we get

$$\begin{aligned}\|\Delta_k(f)+Df(x')\hat{\mathbf{a}}_k\| \le \|\hat{\Delta}_k(f)+Df(\xi)\hat{\mathbf{a}}_k\|+\|\Delta_k(f)-\hat{\Delta}_k(f)\| \\ +\|Df(x')-Df(\xi)\|\|\hat{\mathbf{a}}_k\|=O(\epsilon).\end{aligned} \tag{36}$$

Now, let $\hat{\mathbf{a}}_k=\hat{\lambda}_{k,1}v_1'+\cdots+\hat{\lambda}_{k,n-1}v_{n-1}'+\hat{\lambda}_{k,n}v_n'$ and $\mathbf{a}_k'=\lambda_{k,1}'v_1'+\cdots+\lambda_{k,n-1}'v_{n-1}'$, where the values of $\lambda_{k,1}',\ldots,\lambda_{k,n-1}'$ satisfy (**??**), then

$$\mathbf{a}_k'-\hat{\mathbf{a}}_k=V'\begin{pmatrix}\lambda_{k,1}'\\ \vdots\\ \lambda_{k,n-1}'\\ 0\end{pmatrix}-V'\begin{pmatrix}\hat{\lambda}_{k,1}\\ \vdots\\ \hat{\lambda}_{k,n-1}\\ \hat{\lambda}_{k,n}\end{pmatrix}=V'\begin{pmatrix}\lambda_{k,1}'-\hat{\lambda}_{k,1}\\ \vdots\\ \lambda_{k,n-1}'-\hat{\lambda}_{k,n-1}\\ -\hat{\lambda}_{k,n}\end{pmatrix}. \tag{37}$$

Let $\tilde{\mathbf{a}}_k=\hat{\lambda}_{k,1}v_1'+\cdots+\hat{\lambda}_{k,n-1}v_{n-1}'$, then $\tilde{\mathbf{a}}_k$ is a feasible solution of (**??**) satisfying

$$\begin{aligned}\|\Delta_k(f)+Df(x')\tilde{\mathbf{a}}_k\| &\le \|\Delta_k(f)+Df(x')\hat{\mathbf{a}}_k\|+\|Df(x')(\tilde{\mathbf{a}}_k-\hat{\mathbf{a}}_k)\|\\ &=\|\Delta_k(f)+Df(x')\hat{\mathbf{a}}_k\|+\|\hat{\lambda}_{k,n}Df(x')v_n'\|=O(\epsilon).\end{aligned}$$

Because the optimal value is less than any feasible value of (**??**), we get

$$|{u'}_n^*\Delta_k(f)|=\|\Delta_k(f)+Df(x')\mathbf{a}_k'\|\le\|\Delta_k(f)+Df(x')\tilde{\mathbf{a}}_k\|=O(\epsilon), \tag{38}$$

which proves the third part of (**??**). Therefore, by (**??**) and (**??**), we have

$$\|Df(x')(\hat{\mathbf{a}}_k-\mathbf{a}_k')\|\le\|\Delta_k(f)+Df(x')\hat{\mathbf{a}}_k\|+\|\Delta_k(f)+Df(x')\mathbf{a}_k'\|=O(\epsilon).$$

On the other hand, since $U'=(u_1',\ldots,u_n')$ is a unitary matrix,

$$\begin{aligned}\|Df(x')(\mathbf{a}_k'-\hat{\mathbf{a}}_k)\| &= \left\|U'\mathrm{diag}(\sigma_1',\ldots,\sigma_{n-1}',\sigma_n'){V'}^*V'\begin{pmatrix}\lambda_{k,1}'-\hat{\lambda}_{k,1}\\ \vdots\\ \lambda_{k,n-1}'-\hat{\lambda}_{k,n-1}\\ -\hat{\lambda}_{k,n}\end{pmatrix}\right\|\\ &=\left\|\begin{pmatrix}\sigma_1'(\lambda_{k,1}'-\hat{\lambda}_{k,1})\\ \vdots\\ \sigma_{n-1}'(\lambda_{k,n-1}'-\hat{\lambda}_{k,n-1})\\ -\sigma_n'\hat{\lambda}_{k,n}\end{pmatrix}\right\|.\end{aligned}$$

As $\|Df(x')(\mathbf{a}_k'-\hat{\mathbf{a}}_k)\|=O(\epsilon)$ and $\sigma_i'=O(1)$ for $i=1,\ldots,n-1$, we get

$$|\lambda_{k,1}'-\hat{\lambda}_{k,1}|=O(\epsilon),\ldots,|\lambda_{k,n-1}'-\hat{\lambda}_{k,n-1}|=O(\epsilon). \tag{39}$$

Moreover, because $\sum_{i=1}^n\hat{\lambda}_{1,i}^*\hat{\lambda}_{k,i}=\hat{\mathbf{a}}_1^*\hat{\mathbf{a}}_k=0$ and $|\hat{\lambda}_{1,1}|=O(\epsilon),\ldots,|\hat{\lambda}_{1,n-1}|=O(\epsilon)$, $|\hat{\lambda}_{1,n}|=1-O(\epsilon)$, we get

$$|{v_n'}^*\hat{\mathbf{a}}_k|=|\hat{\lambda}_{k,n}|=O(\epsilon), \tag{40}$$

which proves the first part of (**??**).

Finally, by (**??**), (**??**) and (**??**), we have $\|\mathbf{a}_k'-\hat{\mathbf{a}}_k\|=O(\epsilon)$, which completes the inductive proof of (**??**).

The proof of (**??**) is easy since

$$|{u'}_n^*\Delta_\mu(f)-\hat{u}_n^*\hat{\Delta}_\mu(f)|\le\|{u'}_n^*-\hat{u}_n^*\|\|\Delta_\mu(f)\|+\|\hat{u}_n^*\|\|\Delta_\mu(f)-\hat{\Delta}_\mu(f))\|=O(\epsilon).$$

Hence, when x' is sufficiently close to ξ, there exists a proper value of τ such that the multiplicity is correctly estimated via the criterion $|{u'}_n^{*}\Delta_k(f)| \ll \tau < |{u'}_n^{*}\Delta_\mu(f)|$.

4.3 A LinearSolver-Free Estimation for δ

Suppose x' is an approximation of ξ satisfying $\|x'-\xi\| = O(\epsilon)$ and $\|f(x')\| = O(\epsilon^2)$ by Theorem **??**. Let $Df(x') = \sum_{i=1}^{n} \sigma'_i u'_i {v'_i}^{*}$ be the SVD of $Df(x')$ and $\Delta_2, \ldots, \Delta_{\mu-1}, \Delta_\mu \in \mathfrak{D}_{x'}$ be the $\mu-1$ linear differential functionals, which are incrementally formulated by (**??**) and (**??**), where the values of $a_{k,i}$ are given by (**??**) satisfying (**??**), (**??**) and (**??**) by Theorem **??**.

Before we give a closed-form estimation for the step length δ, let us recall the well-known result for refining approximate singular zeros of univariate polynomials. Assume that an a univariate polynomial $f(X)$ has a μ-fold zero ξ, then its Taylor's expansion at ξ can be written as

$$f(X) = (X-\xi)^\mu + \sum_{k>\mu} c_k (X-\xi)^k.$$

The iterative method

$$x' = x - \frac{1}{\mu}\frac{1}{(\mu-1)!}\frac{\partial^{\mu-1} f(X)}{\partial X^{\mu-1}} \tag{41}$$

is of quadratic convergence if the approximate zero x is sufficiently close to ξ since

$$\left\| x - \frac{1}{\mu}\frac{1}{(\mu-1)!}\frac{\partial^{\mu-1} f(x)}{\partial X^{\mu-1}} - \xi \right\| = \left\| \sum_{k\geq 2} c'_k (x-\xi)^k \right\| = O(\|x-\xi\|^2).$$

Inspired by (**??**), based on the recursive reduction (**??**) given in Theorem **??**, the following theorem provides a **closed-form** estimation of the step length δ using the differential functionals instead of solving any linear system comparing to Theorem **??** to achieve the quadratic convergence.

Theorem 7 *Suppose*

$$\delta = -\frac{1}{\mu}\frac{{u'_n}^{*}\Delta_{\mu-1}(f)}{{u'_n}^{*}\Delta_\mu(f)}, \tag{42}$$

*then it satisfies (**??**). Consequently, the approximation $x'' = x' + \delta v'_n$ is of quadratic convergence, i.e., $\|x''-\xi\| = O(\epsilon^2)$ holds.*

Proof We consider the perturbed polynomial system

$$\tilde{f}(X) = f(X) - f(x') - \sigma'_n u'_n {v'_n}^{*}(X-x') - \sum_{k=2}^{\mu-1} u'_n {u'_n}^{*}\Delta_k(f)\left[{v'_n}^{*}(X-x')\right]^k. \tag{43}$$

First, we show that the approximation x' of ξ is now a simple multiple zero of $\tilde{f}$ of multiplicity μ. It is clear that

$$\begin{aligned} \tilde{f}(x') &= 0, \\ D\tilde{f}(x') &= U' \cdot \mathrm{diag}(\sigma'_1, \ldots, \sigma'_{n-1}, 0) \cdot {V'}^{*}. \end{aligned}$$

According to (**??**), we can derive that

$$\begin{aligned}
&{u'_n}^*\Delta_k(\tilde{f}) = {u'_n}^*\Delta_k(f) - {u'_n}^* u'_n {u'_n}^*\Delta_k(f) \;=\; 0,\; (k = 2, \dots, \mu - 1)\\
&{u'_n}^*\Delta_\mu(\tilde{f}) = {u'_n}^*\Delta_\mu(f) \;\neq\; 0.
\end{aligned}$$

Therefore, according to Theorem **??**, x' is a simple multiple zero of $\tilde{f}$ of multiplicity μ and a basis of $\mathcal{D}_{\tilde{f},x'}$ is uniquely determined via (**??**), (**??**) and (**??**). Subsequently, according to (**??**), we get the recursive reduction

$$ {u'_n}^*\tilde{f}(X) = {u'_n}^*\Delta_\mu(f)\left[{v'_n}^*(X - x')\right]^\mu + \sum_{|\beta|>\mu} c_\beta (X - x')^\beta + \sum_{i=1}^{n-1} \tilde{h}_i(X)[{u'_i}^*\tilde{f}(X)], \tag{44}$$

where $c_\beta \in \mathbb{C}$ and $\tilde{h}_i \in \mathbb{C}[X]$ satisfying $\tilde{h}_i(x') = 0$.

Combining (**??**) and (**??**), we get an expansion of ${u'_n}^*f(X)$ at x',

$$\begin{aligned}
{u'_n}^*f(X) &= {u'_n}^*\tilde{f}(X) + {u'_n}^*f(x') + \sigma'_n {v'_n}^*(X - x') + \sum_{k=2}^{\mu-1} {u'_n}^*\Delta_k(f)\left[{v'_n}^*(X - x')\right]^k\\
&= {u'_n}^*f(x') + \sigma'_n {v'_n}^*(X - x') + \sum_{k=2}^{\mu-1} {u'_n}^*\Delta_k(f)\left[{v'_n}^*(X - x')\right]^k\\
&\quad + {u'_n}^*\Delta_\mu(f)\left[{v'_n}^*(X - x')\right]^\mu + \sum_{|\beta|>\mu} c_\beta (X - x')^\beta + \sum_{i=1}^{n-1} \tilde{h}_i(X)[{u'_i}^*\tilde{f}(X)].
\end{aligned}$$

We are ready to complete the proof of the quadratic convergence using the above expansion and the local dual space $\mathcal{D}_{f,\xi}$. Let $\hat{\Lambda}_1 = \hat{a}_{1,1}d_1 + \cdots + \hat{a}_{1,n}d_n \in \mathcal{D}_{f,\xi}$ satisfy $\hat{a}_{1,1}^*\hat{a}_{1,1} + \cdots + \hat{a}_{1,n}^*\hat{a}_{1,n} = 1$, according to (**??**), we have

$$\left|{v'_n}^*\hat{\mathbf{a}}_1\right| = 1 - O(\epsilon),\; \text{where } \hat{\mathbf{a}}_1 = (\hat{a}_{1,1}, \dots, \hat{a}_{1,n})^T.$$

Let $\hat{\Lambda}_{\mu-1} \in \mathcal{D}_{f,\xi}$ be the $(\mu-1)$-th differential functional that is incrementally formulated by (**??**), (**??**) and (**??**), then according to (**??**), we have

$$\left|{v'_n}^*\hat{\mathbf{a}}_k\right| = O(\epsilon),\; \text{where } \hat{\mathbf{a}}_k = (\hat{a}_{k,1}, \dots, \hat{a}_{k,n})^T,\; k = 2, \dots, \mu - 1.$$

Applying $\hat{\Lambda}_{\mu-1}$ to the above expansion, we get

$$\begin{aligned}
0 &= \hat{\Lambda}_{\mu-1}({u'_n}^*f) \qquad (45)\\
&= \hat{\Lambda}_{\mu-1}\left({u'_n}^*f(x') + \sigma'_n {v'_n}^*(X - x') + \sum_{k=2}^{\mu-1} {u'_n}^*\Delta_k(f)\left[{v'_n}^*(X - x')\right]^k\right)\\
&+ \hat{\Lambda}_{\mu-1}\left({u'_n}^*\Delta_\mu(f)\left[{v'_n}^*(X - x')\right]^\mu + \sum_{|\beta|>\mu} c_\beta (X - x')^\beta + \sum_{i=1}^{n-1} \tilde{h}_i[{u'_i}^*f - {u'_i}^*f(x')]\right).
\end{aligned}$$

Now we estimate the error of every term in the above expansion. First, it is clear that

$$\begin{aligned}
\hat{\Lambda}_{\mu-1}\left({u'_n}^* f(x')\right) &= 0, && \text{since } {u'_n}^* f(x') \in \mathbb{C},\\
\hat{\Lambda}_{\mu-1}\left(\tilde{h}_i {u'_i}^* f\right) &= 0, && \text{since } \tilde{h}_i {u'_i}^* f \in I_f,\\
\hat{\Lambda}_{\mu-1}\left(\tilde{h}_i {u'_i}^* f(x')\right) &= O(\epsilon^2), && \text{since } \left|{u'_i}^* f(x')\right| = O(\epsilon^2).\\
\hat{\Lambda}_{\mu-1}\left(c'_\beta (X-x')^\beta\right) &= O(\epsilon^2), && \text{since } |\beta| - (\mu-1) \geq 2,
\end{aligned}$$

Second, similar to the condition (??), the differential functional $\hat{\Lambda}_{\mu-1} \in \mathcal{D}_{f,\xi}$ satisfies

$$\hat{\Lambda}_{\mu-1}\left(\left[{v'_n}^*(X-x')\right]^k\right) = \begin{cases} O(\epsilon), & k \leq \mu-2,\\ ({v'_n}^* \hat{\mathbf{a}}_1)^{\mu-1} + O(\epsilon), & k = \mu-1,\\ \mu {v'_n}^*(\xi - x')({v'_n}^* \hat{\mathbf{a}}_1)^{\mu-1} + O(\epsilon^2), & k = \mu. \end{cases} \tag{46}$$

See Appendix for the proofs of the condition (??) . Therefore, we get

$$\begin{aligned}
\hat{\Lambda}_{\mu-1}\left(\sigma'_n {v'_n}^*(X-x')\right) &= O(\epsilon^2), \qquad \text{since } \sigma'_n = O(\epsilon)\\
\hat{\Lambda}_{\mu-1}\left(\sum_{k=2}^{\mu-2} {u'_n}^* \Delta_k(f)\left[{v'_n}^*(X-x')\right]^k\right) &= O(\epsilon^2), \qquad \text{since } |{u'_n}^* \Delta_k(f)| = O(\epsilon)\\
\hat{\Lambda}_{\mu-1}\left({u'_n}^* \Delta_{\mu-1}(f)\left[{v'_n}^*(X-x')\right]^{\mu-1}\right) &= {u'_n}^* \Delta_{\mu-1}(f)({v'_n}^* \hat{\mathbf{a}}_1)^{\mu-1} + O(\epsilon^2),\\
\hat{\Lambda}_{\mu-1}\left({u'_n}^* \Delta_\mu(f)\left[{v'_n}^*(X-x')\right]^{\mu}\right) &= \mu {u'_n}^* \Delta_\mu(f) {v'_n}^*(\xi-x')({v'_n}^* \hat{\mathbf{a}}_1)^{\mu-1} + O(\epsilon^2).
\end{aligned}$$

Combining all above term-wise error estimations with (??), we have

$$\left|{u'_n}^* \Delta_{\mu-1}(f)({v'_n}^* \hat{\mathbf{a}}_1)^{\mu-1} + \mu {u'_n}^* \Delta_\mu(f) {v'_n}^*(\xi-x')({v'_n}^* \hat{\mathbf{a}}_1)^{\mu-1}\right| = O(\epsilon^2).$$

Finally, since $|{v'_n}^* \hat{\mathbf{a}}_1| = 1 - O(\epsilon)$ and $|{u'_n}^* \Delta_\mu(f)| \geq \tau > 0$, we derive

$$\left|{v'_n}^*(\xi - x') + \frac{1}{\mu}\frac{{u'_n}^* \Delta_{\mu-1}(f)}{{u'_n}^* \Delta_\mu(f)}\right| = O(\epsilon^2).$$

Therefore, (??) is satisfied for $\delta = -\frac{1}{\mu}\frac{{u'_n}^* \Delta_{\mu-1}(f)}{{u'_n}^* \Delta_\mu(f)}$. Consequently, let $x'' = x' + \delta v'_n$, according to Remark ??, we have $\|x'' - \xi\| = O(\epsilon^2)$.

Based on Theorem ?? and Theorem ??, we propose a modification of Algorithm 1, where two refinement steps involving **LinearSolver** are replaced by two **closed-form** solutions (??) and (??) respectively, meanwhile the quadratic convergence is retained.

Algorithm 3 SimpleMultipleRootRefiner

Input:

f: a polynomial system associated with a simple multiple zero ξ;

x: an identified approximation of ξ for a specified tolerance τ;

Output:

x'': the refined approximation of ξ;

1: Calculate the SVD of $Df(x)$,

$$Df(x) = U \cdot \mathrm{diag}(\sigma_1, \ldots, \sigma_n) \cdot V^*,$$

where $U = (u_1, \ldots, u_n)$, $V = (v_1, \ldots, v_n)$ are unitary and $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$.

2: Refine x to x',

$$x' = N_1(f, x) \stackrel{\text{def}}{=} x - \sum_{i=1}^{n-1} \frac{1}{\sigma_i} v_i u_i^* f(x).$$

3: Calculate the SVD of $Df(x')$,

$$Df(x') = U' \cdot \mathrm{diag}(\sigma_1', \ldots, \sigma_n') \cdot V'^*,$$

where $U' = (u_1', \ldots, u_n')$, $V' = (v_1', \ldots, v_n')$ are unitary and $\sigma_1' \geq \cdots \geq \sigma_n' \geq 0$.

4: Refine x' to x'',

$$x'' = N_2(f, x') \stackrel{\text{def}}{=} x' - \frac{1}{\mu} v_n' \frac{{u_n'}^* \Delta_{\mu-1}(f)}{{u_n'}^* \Delta_\mu(f)},$$

where ${u_n'}^* \Delta_{\mu-1}$ and ${u_n'}^* \Delta_\mu$ are by-products of LeastSquares-Free ApproximateMultiplicityStructure.

Remark 3 Comparing to [**?**], whose proof of the quadratic convergence was based on the deflation techniques and the symbolic-numeric reduction to geometric involutive forms, the new proof is based on the following facts:

1. $x' = N_1(f, x)$ adjusts the given approximation x to a special position x' such that there exists a refined approximation with quadratic convergence along the direction v_n' (Theorem **??**);
2. x' is an exact simple multiple zero of the perturbed system $\tilde{f}(X)$ (**??**), where the polynomial ${u_n'}^* \tilde{f}(X)$ has a recursive reduction (**??**);
3. a suitable step length is estimated using the differential functionals such that $x'' = N_2(f, x')$ achieves the quadratic convergence (Theorem **??**).

Additionally, based on Theorem **??**, we propose a modification of Algorithm 2 below, where the parameter-estimation step involving **LeastSquares** is replaced by the **closed-form** solution (**??**).

Algorithm 4 LeastSquares-Free ApproximateMultiplicityStructure

Input:
f: a polynomial system associated with a simple multiple zero ξ;
x: an identified approximation of ξ for a specified tolerance τ;
Output:
μ: an estimation of the multiplicity of ξ;
$a_{k,i}$: an estimation of μn parameters for a basis of $\mathcal{D}_{f,\xi}$;

1: Let $Df(x) = U \cdot \mathrm{diag}(\sigma_1, \ldots, \sigma_n) \cdot V^*$, where $U = (u_1, \ldots, u_n)$, $V = (v_1, \ldots, v_n)$ are unitary and $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$, then

$$a_{1,i} = v_{n,i},$$

and set $\Lambda_1 = a_{1,1}d_1 + \cdots + a_{1,n}d_n$.

2: From $k = 2$, formulate Δ_k incrementally according to (**??**): while $|u_n^* \Delta_k(f)| < \tau$, then calculate $a_{k,i}$,

$$\begin{pmatrix} a_{k,1} \\ \vdots \\ a_{k,n} \end{pmatrix} = -\sum_{i=1}^{n-1} \frac{1}{\sigma_i} v_i u_i^* \Delta_k(f),$$

set $\Lambda_k = \Delta_k + a_{k,1}d_1 + \cdots + a_{k,n}d_n$ and repeat the step; otherwise, return $\mu = k$.

4.4 Example ?? (continued)

We illustrate the main steps of the proposed two-step Newton's method for Example **??**.

#1 SVD

$$\begin{aligned} Df(x) &= \begin{bmatrix} 2.02 & 1 \\ 1 & 0.5025 \end{bmatrix} \\ &= \begin{bmatrix} -0.8957 & -0.4447 \\ -0.4447 & 0.8957 \end{bmatrix} \cdot \begin{bmatrix} 2.5165 & 0 \\ 0 & 0.00598 \end{bmatrix} \cdot \begin{bmatrix} -0.8957 & -0.4447 \\ -0.4447 & 0.8957 \end{bmatrix} \end{aligned}$$

#1 Refinement

$$\begin{aligned} x' &= \begin{pmatrix} 1.01 \\ 2.01 \end{pmatrix} - \frac{1}{2.5165} \begin{pmatrix} -0.8957 \\ -0.4447 \end{pmatrix} \cdot (-0.8957, -0.4447) \cdot \begin{pmatrix} 0.0301 \\ 0.0150125 \end{pmatrix} \\ &= \begin{pmatrix} 0.998 \\ 2.004 \end{pmatrix}. \end{aligned}$$

#2 SVD

$$\begin{aligned} Df(x') &= \begin{bmatrix} 1.996 & 1 \\ 1 & 0.501 \end{bmatrix} \\ &= \begin{bmatrix} -0.8941 & -0.4479 \\ -0.4479 & 0.8941 \end{bmatrix} \cdot \begin{bmatrix} 2.497 & 0 \\ 0 & 0.000021 \end{bmatrix} \cdot \begin{bmatrix} -0.8941 & -0.4479 \\ -0.4479 & 0.8941 \end{bmatrix} \end{aligned}$$

Parameter-Estimation Initialize

$$(a_{1,1}, a_{1,2})^T = v_2' = (-0.4479, 0.8941)^T,$$

then $\mathit{\Delta}_2(f) = (0.2006, 0.09992)^T$ by (**??**) and $|\mathbf{u_2'}^* \mathbf{\Delta_2(f)}| = \mathbf{0.00053 < 0.01}$. For $k = 2$, we have

$$\begin{pmatrix} a_{2,1} \\ a_{2,2} \end{pmatrix} = -\frac{1}{2.497} \begin{pmatrix} -0.8941 \\ -0.4479 \end{pmatrix} \cdot (-0.8941, -0.4479) \cdot \begin{pmatrix} 0.2006 \\ 0.09992 \end{pmatrix}$$
$$= \begin{pmatrix} -0.0803 \\ -0.0402 \end{pmatrix},$$

then $\mathit{\Delta}_3(f) = (0.0719, -0.009)^T$ by (**??**) and $|\mathbf{u_2'}^* \mathbf{\Delta_3(f)}| = \mathbf{0.04024 > 0.01}$. The multiplicity is recovered as $\mu = 3$.

#2 Refinement

$$x'' = \begin{pmatrix} 0.998 \\ 2.004 \end{pmatrix} - \frac{1}{3} \begin{pmatrix} -0.4479 \\ -0.8941 \end{pmatrix} \cdot \frac{0.00053}{0.04024}$$
$$= \begin{pmatrix} 1.000007 \\ 2.000106 \end{pmatrix}.$$

Let $x_0 = x = (1.01, 2.01)$, $x_1 = x'' = (1.000007, 2.000106)$, and $x_2 = N_2(f, N_1(f, x_1))$ and $x_3 = N_2(f, N_1(f, x_2))$ are the refined approximations, then the distance $\|x_k - \xi\|$ is reduced from 10^{-2} to 10^{-16} for $k = 0, 1, 2, 3$:

$$0.01414 \rightarrow 0.000106 \rightarrow 6.8462 \times 10^{-9} \rightarrow 4.4409 \times 10^{-16}.$$

4.5 Numerical Stability

In this subsection, we show the numerical stability of the algorithms with respect to the initially given approximate solution, the separation bound and the tolerance.

Example 2 The polynomial system

$$f = \{X^2 + Y^3, X + 10^{-k} \cdot Y\}$$

has a simple double zero $\xi = (0, 0)$ and a nearby simple zero $\eta = (10^{-3k}, -10^{-2k})$.

The following experiments are done in Maple 2018 with the Maple environment variable being set by the statement "Digits:=14".

- The first experiment is conducted by setting $k = 1$, i.e., $\eta = (0.001, -0.01)$ and varying x from $(0.0001, 0.0001)$ to $(0.001, 0.001)$ and $(0.01, 0.01)$. We notice from Table **??** that the accuracy of the approximate solution x affects the convergence rate of our algorithm. When x is sufficiently close to $\xi = (0, 0)$ and far from another zero $\eta = (0.001, -0.01)$, the first row of Table **??** indicates that x converges to ξ quadratically as our theoretical analysis proved. When x is chosen between the double zero $\xi = (0, 0)$ and the simple zero $\eta = (0.001, -0.01)$, then the convergent rate drops significantly.

  Note that the symbol $\rightarrow$ means the error is measured by the Euclidean distance between x'' and the simple double zero ξ.

– The second experiment is conducted by setting $x = (0.0001, 0.0001)$ and varying $k = 1, 2, 3$. Table **??** shows that when the double zero $\xi = (0, 0)$ is well separated from another zero $\eta = (0.001, -0.01)$, and the approximate solution $x = (0.0001, 0.0001)$ is close to ξ, then our algorithm converges quadratically to the double zero ξ. However, when $\eta = (10^{-6}, -10^{-4})$ or $\eta = (10^{-9}, -10^{-6})$, then for the given tolerance $\tau = 0.0001$, ξ and η can be treated as a cluster of zeros of f, which behaves like a simple triple zero of a slightly perturbed system numerically. Therefore, x converges to the centroid $(2 \cdot \xi + 1 \cdot \eta)/3$ of ξ and η quadratically.

  Note that the symbol $\rightsquigarrow$ means the error is measured by the Euclidean distance between x'' and the centroid $(2 \cdot \xi + 1 \cdot \eta)/3$.

– The third experiment is conducted by setting $k = 2$, $\eta = (10^{-6}, -10^{-4})$, $x = (0.0001, 0.0001)$, and varying the value of τ from 10^{-3} to 10^{-4}. Results are recorded in Table **??**. The convergence and the multiplicity are affected by values of τ. When $\tau = 10^{-3}$, the first row indicates that x converges to the centroid $(2 \cdot \xi + 1 \cdot \eta)/3$ quadratically, i.e., x is identified as close to a cluster of three zeros. When $\tau = 10^{-4}$, x is identified as close to the double root ξ, the second row indicates that x converges to the simple double zero ξ, however the convergence rate is linear rather than quadratic.

Table 2 Fix $k = 1$, vary $x = (0.0001, 0.0001), (0.001, 0.001), (0.01, 0.01)$

$\|x - \xi\|$	$\|x - \eta\|$	τ	μ	error per iteration
1.41e-04	1.01e-02	0.001	2	1.41e-04→1.17e-06→2.03e-10→3.28e-16
1.41e-03	1.10e-02	0.001	2	1.41e-03→9.45e-05→1.30e-06→2.50e-10
1.41e-02	1.68e-02	0.001	2	1.41e-02→3.27e-03→8.06e-04→7.81e-05

Table 3 Fix $x = (0.0001, 0.0001)$, vary $k = 1, 2, 3$

k	$\|\xi - \eta\|$	τ	μ	error per iteration
1	1.00e-02	0.001	2	1.41e-04→1.17e-06→2.03e-10→3.28e-16
2	1.00e-04	0.001	3	1.66e-04⇝2.04e-12⇝1.45e-17⇝1.45e-17
3	1.00e-06	0.001	3	1.42e-04⇝2.00e-12⇝3.55e-21⇝3.55e-21

Table 4 Fix $k = 2$ and $x = (0.0001, 0.0001)$, vary $\tau = 0.001, 0.0001$

τ	μ	error per iteration
0.001	3	1.66e-04⇝2.04e-12⇝1.45e-17⇝1.45e-17
0.0001	2	1.41e-04→3.70e-05→9.74e-06→1.10e-06

Three experiments reveal that the numerical stability of our algorithm is not only affected by the accuracy of the identified approximation zero x but also by the distribution of the other zeros of f. The quantitative analysis of f and x for guaranteeing quadratic convergence is left for future works, which is actually the generalization of Smale's α-theory to simple multiple zeros.

5 Implementation and Experiment

In this section, we show that the proposed two-step Newton's method not only enables a more elegant proof of the quadratic convergence, but also results in a more efficient implementation compared with MultipleRootRefinerBreadthOne [?] and ApproximateMultiplicityStructure [?].

5.1 Implemental Comparison

Comparing Algorithm 3 and Algorithm 4 to MultipleRootRefinerBreadthOne [?] and ApproximateMultiplicityStructure [?], they both use two SVDs in their implementation, namely,

$$\text{Step 1: } Df(x) = U \cdot \operatorname{diag}(\sigma_1, \ldots, \sigma_n) \cdot V^*,$$

where $U = (u_1, \ldots, u_n)$, $V = (v_1, \ldots, v_n)$ are unitary and $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$,

$$\text{Step 3: } Df(x') = U' \cdot \operatorname{diag}(\sigma'_1, \ldots, \sigma'_n) \cdot {V'}^*,$$

where $U' = (u'_1, \ldots, u'_n)$, $V' = (v'_1, \ldots, v'_n)$ are unitary and $\sigma'_1 \geq \cdots \geq \sigma'_n \geq 0$.

However, how they exploit two outputs in those subsequent steps: #1 Refinement (Step 2), #2 Refinement (Step 4) and Parameter-Estimation (Step 2) are different.

#1 Refinement MultipleRootRefinerBreadthOne takes σ_n from the SVD in Step 1 as the regularization parameter, then solves a damped least-squares problem for completing #1 Refinement, which totally costs $O(n^3)$ floating-point multiplications. Respectively, Algorithm 3 takes the sum of the first $n-1$ products of $v_i u_i^* f(x)/\sigma_i$, which uses $O(n^2)$ floating-point multiplications.

Table 5 Implemental Comparison on #1 Refinement

MultipleRootRefinerBreadthOne [?]	SimpleMultipleRootRefiner
$x' = x + y$ where y is the optimal solution of $\min\ \|Df(x) \cdot y + f(x)\|^2 + \sigma_n \|y\|^2$	$x' = x + y$ where $y = -\sum_{i=1}^{n-1} v_i u_i^* f(x)/\sigma_i$
$O(n^3)$	$O(n^2)$

#2 Refinement MultipleRootRefinerBreadthOne solves a least-squares problem for completing #2 Refinement, which costs about $O(n^3)$ floating-point multiplications. Respectively, Algorithm 3 takes only one-time floating-point multiplication because ${u'_n}^* \Delta_{\mu-1}(f)$ and ${u'_n}^* \Delta_\mu(f)$ are two by-products of Algorithm 4.

Parameter-Estimation ApproximateMultiplicityStructure recursively solves $\mu-1$ least-squares problems for estimating the values of $a_{k,i}$, which totally costs $O(\mu n^3 - n^3)$ floating-point multiplications. Respectively, Algorithm 4 takes the sum of the first $n-1$ products of $v'_i {u'_i}^* \Delta_k(f)/\sigma'_i$ recursively, which costs $O(\mu n^2 - n^2)$ floating-point multiplications.

Table 6 Implemental Comparison on #2 Refinement

MultipleRootRefinerBreadthOne [?]	SimpleMultipleRootRefiner
$x'' = x' + \delta \cdot v'_n/\mu$ where (δ, y) is the optimal solution of min $\Vert \Delta_\mu(f) \cdot \delta + Df(x) \cdot y + \Delta_{\mu-1}(f)\Vert^2$ subject to ${v'_n}^* \cdot y = 0$	$x'' = x' + \delta \cdot v'_n/\mu$ where $\delta = -{u'_n}^* \Delta_{\mu-1}(f)/{u'_n}^* \Delta_\mu(f)$
$O(n^3)$	$O(1)$

Table 7 Implemental Comparison on Parameter-Estimation

ApproximateMultiplicityStructure [?]	LS-Free ApproximateMultiplicityStructure
$\mathbf{a}_k$ is the optimal solution of min $\Vert Df(x) \cdot \mathbf{a}_k + \Delta_k(f)\Vert^2$ subject to ${v'_n}^* \cdot \mathbf{a}_k = 0$	$\mathbf{a}_k = -\sum_{i=1}^{n-1} v'_i {u'_i}^* \Delta_k(f)/\sigma'_i$
$O(\mu n^3 - n^3)$	$O(\mu n^2 - n^2)$

5.2 Experimental Comparison

In order to examine the performance of the proposed two-step Newton's method comparing to the previous approach in [**?**,**?**], we compare both implementations on two categories of examples: the benchmark examples in literature and the large-size examples.

All experiments are done in Maple 2018 on a MacBook Pro with 2.3 GHz 8-Core Intel Core i9 (processor) and 16 GB 2400 MHz DDR4 (memory) running Catalina. The Maple environment variable is set by the statement "Digits:=14". All timings are measured as elapsed time in seconds. The code of SimpleMultipleRootRefiner and MultipleRootRefinerBreadthOne and the worksheet of all experimental results are available at `http://www.mmrc.iss.ac.cn/~lzhi/Research/hybrid/RefineSimpleMultipleZeros/`

Benchmark Examples in Literature Some benchmark examples in the literature are tested and their experimental results are listed in Table **??**. All polynomial systems are well-constrained (# variables = # equations = n) associated with a simple multiple zero of μ. τ denotes the specified tolerance in the experiment for correctly recovering μ. The last two columns illustrate the averaging elapsed times for one iteration of MultipleRootRefinerBreadthOne and SimpleMultipleRootRefiner respectively. In Table **??**, eight examples are listed in order of n and μ, where the fourth example is modified from [**?**] (the original system is of multiplicity 4). Obviously, our proposed method is 2x-3x faster.

Examples in Table **??** are of small-size ($n < 4$) such that it is hard to confirm the efficiency of the proposed method. Some large-size examples are further examined.

Large-Size Examples We demonstrate the efficiency of the proposed method by testing the following example, which possesses a simple multiple zero at the origin of multiplicity $\mu = k$. The experimental comparison on four examples ($n =$

Table 8 Experimental Comparison on Benchmark Examples in Literature

Systems	n	μ	τ	previous	proposed
Decker3 [?]	2	2	0.1	0.0117	0.0063
Decker1 [?]	2	3	0.1	0.0160	0.0067
Decker2 [?]	2	4	0.1	0.0213	0.0087
Rump [?]	2	5	0.1	0.0327	0.0117
Ojika3 [?]	3	2	0.01	0.0117	0.0063
Ojika4 [?]	3	3	0.1	0.0163	0.0077
Giusti [?]	3	4	0.1	0.0207	0.0097
Dayton [?]	3	5	0.1	0.0237	0.0113

100/1000, $\mu = 2/3$) are listed in Table **??**. The last two columns illustrate the averaging runtime for one iteration, where the averaging runtime for three different solvers in Table **??**, Table **??** and Table **??** are listed in brackets.

Example 3 [?] Consider $f = \{X_1^2 + X_1 - X_2, \ldots, X_{n-1}^2 + X_{n-1} - X_n, X_n^k\}$ associated with $\xi = (0, \ldots, 0)$ of multiplicity $\mu = k$.

Table 9 Experimental Comparison on Large-Size Examples

n	μ	τ	previous (LeastSquares)	proposed (Closed-Form)
100	2	1.0e-05	0.8793 (0.5523)	0.1540 (0.0110)
100	3	1.0e-05	1.7827 (0.6460)	0.8597 (0.0130)
1000	2	1.0e-06	31.8237 (19.5770)	9.3320 (0.1923)
1000	3	1.0e-06	60.8303 (25.3040)	31.9803 (0.2997)

In Table **??**, it is clearly observed from the last column that the new proposed method is 2x-5x faster than the previous method in [?,?]. If we pay further attention to the three different solvers in Table **??**, Table **??** and Table **??**, the speed-up proportion rises up to 50x-100x faster.

5.3 Other Experimental Results

Examples below are given to demonstrate that our proposed methods are applicable to analytic systems and polynomial systems with clusters of simple zeros as well.

Analytic-Singularity Examples

Example 4 [?] Consider $f = \{X^2 sin(Y),\ Y - Z^2,\ Z + sin(X^4)\}$ associated with $\xi = (0, 0, 0)$ and an approximation $x = (0.001, 0.001, 0.001)$.

The error in each iteration is measured as the Euclidian distance between ξ and the current approximation. The quadratic convergence is clearly observed.

Table 10 Experimental Results on Analytic-Singularity Examples

n	μ	τ	error per iteration	time (sec.)
3	10	0.1	1.73e-03→1.00e-06→4.34e-22	0.03367

Zero-Cluster Examples

Example 5 [?] Consider $f = \{14\ X + 33\ Y - 3\sqrt{5}\ (X^2 + 4\ XY + 4\ Y^2 + 2) + \sqrt{7} + X^3 + 6\ X^2Y + 12\ XY^2 + 8\ Y^3, 41\ X - 18\ Y - \sqrt{5} + 8\ X^3 - 12\ X^2Y + 6\ XY^2 - Y^3 + 3\sqrt{7}\ (4\ XY - 4\ X^2 - Y^2 - 2)\}$ associated with $\xi = (2\sqrt{7}/5 + \sqrt{5}/5, -\sqrt{7}/5 + 2\sqrt{5}/5)$ and an approximation $x = (1.506, 0.366)$.

Table 11 Experimental Results on Zero-Cluster Examples

n	μ	τ	error per iteration	time (sec.)
2	5	0.5	8.71e-04→1.55e-09→1.48e-13	0.01100

The irrational coefficients of f are rounded to 14 digits such that the numerically-rounded system $\tilde{f}$ has a cluster of five simple zeros centering at ξ. The error in each iteration is measured as the Euclidian distance between the rounded ξ and the current approximation. The quadratic convergence is clearly observed.

Acknowledgement

The authors would like to thank referees for giving constructive comments which helped improving the quality of the exposition.

Conflict of Interest

The authors declare that they have no conflict of interest.

References

1. T. A. Akoglu, J. D. Hauenstein, and A. Szanto, *Certifying solutions to overdetermined and singular polynomial systems over q*, Journal of Symbolic Computation, 84 (2018), pp. 147 – 171.
2. L. Blum, F. Cucker, M. Shub, and S. Smale, *Complexity and Real Computation*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1998.
3. B. Dayton, *Numerical local rings and local solutions of nonlinear systems*, in ProcSNC07, J. Verschelde and S. M. Watt, eds., New York, N. Y., 2007, ACM Press, pp. 79–86.
4. B. Dayton, T. Li, and Z. Zeng, *Multiple zeros of nonlinear systems*, Mathematics of Computation, 80 (2011), pp. 2143–2168.
5. B. Dayton and Z. Zeng, *Computing the multiplicity structure in solving polynomial systems*, in Proceedings of the 2005 International Symposium on Symbolic and Algebraic Computation, M. Kauers, ed., New York, NY, USA, 2005, ACM, pp. 116–123.

6. D. W. Decker, H. B. Keller, and C. T. Kelley, *Convergence rates for Newton's method at singular points*, SIAM Journal on Numerical Analysis, 20 (1983), pp. 296–314.
7. D. W. Decker, H. B. Keller, and C. T. Kelley, *Broyden's method for a class of problems having singular jacobian at the root*, SIAM Journal on Numerical Analysis, 22 (1985), pp. 566–574.
8. D. W. Decker and C. T. Kelley, *Newton's method at singular points I*, SIAM Journal on Numerical Analysis, 17 (1980), pp. 66–70.
9. D. W. Decker and C. T. Kelley, *Newton's method at singular points II*, SIAM Journal on Numerical Analysis, 17 (1980), pp. 465–471.
10. D. W. Decker and C. T. Kelley, *Convergence acceleration for Newton's method at singular points*, SIAM Journal on Numerical Analysis, 19 (1982), pp. 219–229.
11. J. P. Dedieu and M. Shub, *On simple double zeros and badly conditioned zeros of analytic functions of n variables*, Mathematics of Computation, 70 (2001), pp. 319–327.
12. M. Giusti, G. Lecerf, B. Salvy, and J.-C. Yakoubsohn, *On location and approximation of clusters of zeros of analytic functions*, Foundations of Computational Mathematics, 5 (2005), pp. 257–311.
13. M. Giusti, G. Lecerf, B. Salvy, and J.-C. Yakoubsohn, *On location and approximation of clusters of zeros: case of embedding dimension one*, Foundations of Computational Mathematics, 7 (2007), pp. 1–58.
14. M. Giusti and J.-C. Yakoubsohn., *Multiplicity hunting and approximating multiple roots of polynomial systems*, in Contemporay Mathematics 604, Recent Advances in Real Complexity and Computation, American Mathematical Society, 2013, pp. 104–128.
15. M. Giusti and J.-C. Yakoubsohn, *Numerical approximation of multiple isolated roots of analytic systems*. Manuscript available at `https://arxiv.org/abs/1707.06301`, 2017.
16. A. Griewank, *Analysis and modification of Newton's method at singularities*, thesis, Australian National University, 1980.
17. A. Griewank, *On solving nonlinear equations with simple singularities or nearly singular solutions*, SIAM Review, 27 (1985), pp. 537–563.
18. A. Griewank and M. R. Osborne, *Newton's method for singular problems when the dimension of the null space is* > 1, SIAM Journal on Numerical Analysis, 18 (1981), pp. 145–149.
19. Z. Hao, W. Jiang, N. Li, and L. Zhi, *Computing simple multiple zeros of polynomial systems*. Manuscript available at `https://www.arxiv.org/pdf/1703.03981.pdf`, 2017.
20. Z. Hao, W. Jiang, N. Li, and L. Zhi, *Computing simple multiple zeros of polynomial systems*, Mathematics of computation, 89 (2020), pp. 879–909.
21. J. D. Hauenstein, B. Mourrain, and A. Szanto, *Certifying isolated singular points and their multiplicity structure*, in Proceedings of the 2015 ACM on International Symposium on Symbolic and Algebraic Computation, ISSAC '15, New York, NY, USA, 2015, ACM, pp. 213–220.
22. J. D. Hauenstein, B. Mourrain, and A. Szanto, *On deflation and multiplicity structure*, Journal of Symbolic Computation, 83 (2017), pp. 228 – 253, Special issue on the conference {ISSAC} 2015: Symbolic computation and computer algebra.
23. J. D. Hauenstein and C. W. Wampler, *Isosingular sets and deflation*, Foundations of Computational Mathematics, 13 (2013), pp. 371–403.
24. G. Lecerf, *Quadratic Newton iteration for systems with multiplicity*, Foundations of Computational Mathematics, 2 (2002), pp. 247–293.
25. A. Leykin, J. Verschelde, and A. Zhao, *Newton's method with deflation for isolated singularities of polynomial systems*, Theoretical Computer Science, 359 (2006), pp. 111–122.
26. A. Leykin, J. Verschelde, and A. Zhao, *Higher-order deflation for polynomial systems with isolated singular solutions*, in Algorithms in Algebraic Geometry, A. Dickenstein, F.-O. Schreyer, and A. Sommese, eds., vol. 146 of The IMA Volumes in Mathematics and its Applications, Springer New York, 2008, pp. 79–97.
27. N. Li and L. Zhi, *Compute the multiplicity structure of an isolated singular solution: case of breadth one*, Journal of Symbolic Computation, 47 (2012), pp. 700–710.
28. N. Li and L. Zhi, *Computing isolated singular solutions of polynomial systems: case of breadth one*, SIAM Journal on Numerical Analysis, 50 (2012), pp. 354–372.
29. N. Li and L. Zhi, *Verified error bounds for isolated singular solutions of polynomial systems: case of breadth one*, Theoretical Computer Science, 479 (2013), pp. 163–173.

30. 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, A. Leykin, ed., ISSAC '11, New York, NY, USA, 2011, ACM, pp. 249–256.
31. A. Mantzaflaris, B. Mourrain, and A. Szanto, *Punctual Hilbert scheme and certified approximate singularities*, in Proceedings of the 45th International Symposium on Symbolic and Algebraic Computation, ISSAC '20, New York, NY, USA, 2020, Association for Computing Machinery, pp. 336–343.
32. A. P. Morgan, A. J. Sommese, and C. W. Wampler, *Computing singular solutions to polynomial systems*, Advances in Applied Mathematics, 13 (1992), pp. 305–327.
33. T. Ojika, *Modified deflation algorithm for the solution of singular problems. i. a system of nonlinear algebraic equations*, Journal of Mathematical Analysis and Applications, 123 (1987), pp. 199 – 221.
34. T. Ojika, *A numerical method for branch points of a system of nonlinear algebraic equations*, Applied Numerical Mathematics, 4 (1988), pp. 419 – 430.
35. T. Ojika, S. Watanabe, and T. Mitsui, *Deflation algorithm for the multiple roots of a system of nonlinear equations*, Journal of Mathematical Analysis and Applications, 96 (1983), pp. 463 – 479.
36. L. Rall, *Convergence of the Newton process to multiple solutions*, Numerische Mathematik, 9 (1966), pp. 23–37.
37. G. W. Reddien, *On Newton's method for singular problems*, SIAM Journal on Numerical Analysis, 15 (1978), pp. 993–996.
38. G. W. Reddien, *Newton's method and high order singularities*, Computers & Mathematics with Applications, 5 (1979), pp. 79 – 86.
39. S. M. Rump and S. Graillat, *Verified error bounds for multiple roots of systems of nonlinear equations*, Numerical Algorithms, 54 (2010), pp. 359–377.
40. Y.-Q. Shen and T. J. Ypma, *Newton's method for singular nonlinear equations using approximate left and right nullspaces of the Jacobian*, Applied Numerical Mathematics, 54 (2005), pp. 256 – 265, 6th IMACS International Symposium on Iterative Methods in Scientific Computing.
41. Y.-Q. Shen and T. J. Ypma, *Solving rank-deficient separable nonlinear equations*, Appl. Numer. Math., 57 (2007), pp. 609–615.
42. M. Shub and S. Smale, *Computational complexity: On the geometry of polynomials and a theory of cost: I*, Annales Scientifiques De L École Normale Supérieure, 18 (1985), pp. 107–142.
43. M. Shub and S. Smale, *Computational complexity: On the geometry of polynomials and a theory of cost: II*, SIAM Journal on Computing, 15 (1986), pp. 145–161.
44. M. Shub and S. Smale, *Complexity of bezout's theorem IV: Probability of success; extensions*, SIAM Journal on Numerical Analysis, 33 (1996), pp. 128–148.
45. S. Smale, *The fundamental theorem of algebra and complexity theory*, Bulletin of the American Mathematical Society, 4 (1981), pp. 1–36.
46. S. Smale, *Newton's method estimates from data at one point*, in The Merging of Disciplines: New Directions in Pure, Applied and Computational Mathematics, Springer New York, 1986, pp. 185–196.
47. H. Stetter, *Numerical Polynomial Algebra*, Society for Industrial and Applied Mathematics, 2004.
48. H. Weyl, *Das asymptotische verteilungsgesetz der eigenwerte linearer partieller differentialgleichungen (mit einer anwendung auf die theorie der hohlraumstrahlung)*, Mathematische Annalen, 74 (1912), pp. 441–479.
49. X. Wu and L. Zhi, *Computing the multiplicity structure from geometric involutive form*, in Proceedings of the Twenty-first International Symposium on Symbolic and Algebraic Computation, D. Jeffrey, ed., ISSAC '08, New York, NY, USA, 2008, ACM, pp. 325–332.
50. X. Wu and L. Zhi, *Determining singular solutions of polynomial systems via symbolic-numeric reduction to geometric involutive forms*, Journal of Symbolic Computation, 47 (2012), pp. 227–238.
51. J.-C. Yakoubsohn, *Finding a cluster of zeros of univariate polynomials*, Journal of Complexity, 16 (2000), pp. 603–638.
52. J.-C. Yakoubsohn, *Simultaneous computation of all zero clusters of an univariate polynomial*, in Foundations of Computational Mathematics, World Sci. Publishing, 2002, pp. 433–455.
53. N. Yamamoto, *Regularization of solutions of nonlinear equations with singular jacobian matrices*, Journal of Information Processing, 7 (1984), pp. 16–21.

Appendix

We prove below conditions (**??**), (**??**), (**??**) and (**??**).

Proof of (??). It is easy to check that

$$\mathbf{d}_{\xi}^{\alpha}\left((X-\xi)^{\beta}\right)=\begin{cases}1,\ \alpha=\beta,\\ 0,\ \text{otherwise.}\end{cases}$$

Since the order of every differential monomial in Δ_k is bounded above by k, we derive

$$\Delta_k\left((X-\xi)^{\beta}\right)=0,\ \text{if } |\beta|>k.$$

Proof of (??). Since $(X-\xi)^{\beta}f_i(X)\in I_f$ for $i=1,\dots,n$, we have

$$\begin{aligned}0&=\Lambda_k\left((X-\xi)^{\beta}f(X)\right)\\ &=\Delta_k\left((X-\xi)^{\beta}f(X)\right)+a_{k,1}d_1\left((X-\xi)^{\beta}f(X)\right)+\cdots+a_{k,n}d_n\left((X-\xi)^{\beta}f(X)\right).\end{aligned}$$

Since $f(\xi)=0$, we have $d_j\left((X-\xi)^{\beta}f(X)\right)=0$ for $j=1,\dots,n$. Therefore, we derive

$$\Delta_k\left((X-\xi)^{\beta}f(X)\right)=0,\quad \text{if } |\beta|>0.$$

Proof of (??). Let $\Phi[\mathbf{a}_k]=\sum_{i=1}^{n}a_{k,i}\Psi_i$, where $\Psi_i:\mathfrak{D}_{\xi}\to\mathfrak{D}_{\xi}$ be the morphism that satisfies $\Psi_i(d_1^{\alpha_1}\cdots d_n^{\alpha_n})=d_i^{\alpha_i+1}\cdots d_n^{\alpha_n}$ if $\alpha_1=\cdots=\alpha_{i-1}=0$ and 0 otherwise for $i=1,\dots,n$, then (**??**) and (**??**) can be rewritten into

$$\begin{aligned}\Delta_k&=\sum_{j=1}^{k-1}\Phi[\mathbf{a}_j](\Lambda_{k-j}),\\ \Lambda_k&=\Delta_k+\Phi[\mathbf{a}_k](1).\end{aligned}$$

Consequently, Δ_k and Λ_k can be written into a sum of homogenous terms formulated by $\prod_{j=1}^{k}\Phi[\mathbf{a}_j]^{n_j}(1)$, where $n_j\geq 0$ and $\sum_{j=1}^{k}n_j\leq k$. We can check that

$$\prod_{j=1}^{k}\Phi[\mathbf{a}_j]^{n_j}(1)\left([v^*(X-\xi)]^l\right)=\begin{cases}\prod_{j=1}^{k}(v^*\mathbf{a}_j)^{n_j},\ \sum_{j=1}^{k}n_j=l,\\ \qquad 0,\ \text{otherwise.}\end{cases}$$

Since $\Phi[\mathbf{a}_1]^k$ is the only pure term of $\Phi[\mathbf{a}_1]$ in Δ_k (other terms are mixed with at least one $\Phi[\mathbf{a}_j]$ for $j=2,\dots,k-1$), and ${v_n}^*\mathbf{a}_1=1$, ${v_n}^*\mathbf{a}_j=0$ according to (**??**), we derive

$$\Delta_k\left(\left[{v_n}^*(X-\xi)\right]^l\right)=\begin{cases}1,\ \text{if } k=l,\\ 0,\ \text{otherwise.}\end{cases}$$

Proof of (??). Since $\|{v'_n}^*(x'-\xi)\|\leq\|{v'_n}^*\|\|x'-\xi\|=O(\epsilon)$, we get

$$\begin{aligned}\left[{v'_n}^*(X-x')\right]^k&=\left[{v'_n}^*(X-\xi)+{v'_n}^*(\xi-x')\right]^k\\ &=\left[{v'_n}^*(X-\xi)\right]^k+k\left[{v'_n}^*(X-\xi)\right]^{k-1}\left[{v'_n}^*(\xi-x')\right]+O(\epsilon^2) &(47)\\ &=\left[{v'_n}^*(X-\xi)\right]^k+O(\epsilon). &(48)\end{aligned}$$

Similar to the proof of (**??**), $\hat{\Delta}_{\mu-1}$ and $\hat{\Lambda}_{\mu-1}$ can be written as a sum of homogenous terms formulated by $\prod_{j=1}^{\mu-1} \Phi[\hat{\mathbf{a}}_j]^{n_j}(1)$, where $n_j \geq 0$ and $\sum_{j=1}^{\mu-1} n_j \leq \mu - 1$. According to (**??**) and (**??**), we know that $|{v'_n}^* \hat{\mathbf{a}}_1| = 1 - O(\epsilon)$ and $|{v'_n}^* \hat{\mathbf{a}}_j| = O(\epsilon)$, so we have

$$\prod_{j=1}^{\mu-1} \Phi[\hat{\mathbf{a}}_j]^{n_j}(1)\left([{v'_n}^*(X-\xi)]^k\right) = \begin{cases} ({v'_n}^* \hat{\mathbf{a}}_1)^k, \sum_{j=1}^{\mu-1} n_j = k \text{ and } n_1 = k, \\ O(\epsilon), \sum_{j=1}^{\mu-1} n_j = k \text{ and } n_1 < k, \\ 0, \text{otherwise}. \end{cases} \tag{49}$$

Since $\Phi[\hat{\mathbf{a}}_1]^{\mu-1}$ is the only pure term of $\Phi[\hat{\mathbf{a}}_1]$ in $\hat{\Lambda}_{\mu-1}$ (other terms are mixed with at least one $\Phi[\hat{\mathbf{a}}_j]$ for $j = 2, \ldots, \mu-1$), combining (**??**) and (**??**), we derive

$$\hat{\Lambda}_{\mu-1}\left(\left[{v'_n}^*(X-x')\right]^k\right) = \begin{cases} O(\epsilon), & k \leq \mu-2, \\ ({v'_n}^* \hat{\mathbf{a}}_1)^{\mu-1} + O(\epsilon), & k = \mu-1. \end{cases}$$

For $k = \mu$, combining (**??**) and (**??**), we derive

$$\begin{aligned} &\hat{\Lambda}_{\mu-1}\left(\left[{v'_n}^*(X-x')\right]^{\mu}\right) \\ =&\hat{\Lambda}_{\mu-1}\left(\left[{v'_n}^*(X-\xi)\right]^{\mu}\right) + \hat{\Lambda}_{\mu-1}\left(\mu\left[{v'_n}^*(X-\xi)\right]^{\mu-1}\left[{v'_n}^*(\xi-x')\right]\right) + O(\epsilon^2) \\ =&0 + \mu {v'_n}^*(\xi-x')({v'_n}^* \hat{\mathbf{a}}_1)^{\mu-1} + O(\epsilon^2). \end{aligned}$$