EXISTENCE VERIFICATION FOR HIGHER DEGREE SINGULAR ZEROS OF COMPLEX NONLINEAR SYSTEMS*

R. BAKER KEARFOTT† AND JIANWEI DIAN‡

Abstract. It is known that, in general, no computational techniques can verify the existence of a singular solution of the nonlinear system of n equations in n variables within a given region $\boldsymbol{x}$ of n-space. However, computational verification that a given number of true solutions exist within a region in complex space containing $\boldsymbol{x}$ is possible. That can be done by computation of the topological degree. In a previous paper, we presented theory and algorithms for the simplest case, when the rank-defect of the Jacobi matrix at the solution is one and the topological index is 2. Here, we will generalize that result to arbitrary topological index $d \geq 2$: We present theory, algorithms, and experimental results. We also present a heuristic for determining the degree, obtaining a value that we can subsequently verify with our algorithms.

Key words. complex nonlinear systems, interval computations, verified computations, singularities, topological degree

AMS subject classifications. 65G10, 65H10

1. Introduction.

Our fundamental problem is

$$\boxed{\begin{array}{l}\text{Given } F:\boldsymbol{x}\to\mathbb{R}^n \text{ and } \boldsymbol{x}\in\mathbb{IR}^n,\ \textit{rigorously} \text{ verify:}\\ \qquad\bullet\ \text{there exists a unique } x^*\in\boldsymbol{x} \text{ such that } F(x^*)=0, \text{ where}\\ \qquad\bullet\ \boldsymbol{x}=\left\{(x_1,x_2,\ldots,x_n)^T\in\mathbb{R}^n \,\middle|\, \underline{x}_i\leq x_i\leq\overline{x}_i, 1\leq i\leq n\right\},\end{array}} \tag{1.1}$$

where the $\underline{x}_i$ and $\overline{x}_i$ represent constant bounds on the problem variables $\boldsymbol{x}_i$. When the Jacobi matrix $F'(x^*)$ well-conditioned and not too quickly varying, interval computations have no trouble proving that there is a unique solution within small boxes with x^* reasonably near the center; see [4, 9, 11]. When $F'(x^*)$ is ill-conditioned or singular, in general, no computational techniques can verify the existence of a solution within a given region $\boldsymbol{x}$ of $\mathbb{R}^n$. However, in the singular case, computational but rigorous verification that a given number of true solutions exist within a region in complex space containing $\boldsymbol{x}$ is possible, as we indicated in [10]. In [10], we studied the simplest case, when the rank-defect of the Jacobi matrix at the solution is one, and we developed and experimentally validated algorithms for the case when the topological index is 2. There, we proved the special case of Theorem 3.1 when $d = 2$ under the same assumptions as those in §2, we developed specialized versions of the algorithms in §4, and we presented varying-dimensional experimental results.

We were surprised and pleased that the results in [10] could be generalized so easily. In particular, we developed an alternate simple, general proof for Theorem 3.1. Furthermore, the algorithms in §4, although not taking advantage of special efficiencies in the degree-2 case, are similar in structure and have the same computational complexity as the algorithms in [10].

1.1. Notation.

We assume familiarity with the fundamentals of interval arithmetic; see [1, 4, 9, 11, 13] for introductory material.

*This work was partially supported by National Science Foundation grant DMS-9701540.

†Department of Mathematics, University of Louisiana at Lafayette, Lafayette, LA 70504, USA (`rbk@louisiana.edu`).

‡Software Design Engineer, Hewlett–Packard Company, 3000 Waterview Parkway, Richardson, TX 75080, USA (`jianwei@rsn.hp.com`).

Throughout, scalars and vectors will be denoted by lower case, while matrices will be denoted by upper case. Intervals, interval vectors (also called "boxes") and interval matrices will be denoted by boldface. For instance, $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ denotes an interval vector, $A = (a_{i,j})$ denotes a point matrix, and $\boldsymbol{A} = (\boldsymbol{a}_{i,j})$ denotes an interval matrix. The midpoint of an interval or interval vector $\boldsymbol{x}$ will be denoted by $\mathrm{m}(\boldsymbol{x})$. Real n-space will be denoted by $\mathbb{R}^n$, while complex n-space will be denoted by $\mathbb{C}^n$.

Suppose $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ is an n-dimensional real box, where $\boldsymbol{x}_k = [\underline{x}_k, \overline{x}_k]$. The non-oriented boundary of $\boldsymbol{x}$, denoted by $\boldsymbol{\partial x}$, consists of $2n$ $(n-1)$-dimensional real boxes

$$\boldsymbol{x}_{\underline{k}} \equiv (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{k-1}, \underline{x}_k, \boldsymbol{x}_{k+1}, \ldots, \boldsymbol{x}_n) \quad \text{and} \quad \boldsymbol{x}_{\overline{k}} \equiv (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{k-1}, \overline{x}_k, \boldsymbol{x}_{k+1}, \ldots, \boldsymbol{x}_n),$$

where $k = 1, \ldots, n$. If $\boldsymbol{x}$ is positively oriented, then the derived orientation of $\boldsymbol{x}_{\underline{k}}$ is $(-1)^k$ and the derived orientation of $\boldsymbol{x}_{\overline{k}}$ is $(-1)^{k+1}$; see [10].

1.2. Formulas from Degree Theory. In [10], we reviewed the topological degree in the context of this paper. Also see [2, 3, 6, 7, 12, 14]. Here, we repeat several properties used in the proofs in subsequent sections.

THEOREM 1.1. ([12, p. 150]) *Suppose that the Jacobian matrix $F'(x)$ is nonsingular at each zero of F. Then, the degree $\mathrm{d}(F, \mathbf{D}, 0)$ is equal to the number of zeros of F at which the determinant of the Jacobian matrix $F'(x)$ is positive minus the number of zeros of F at which the determinant of the Jacobian matrix $F'(x)$ is negative.*

Theorem 1.1 gives some intuition of what the degree is, under the conditions in the theorem. It's a kind of counting of zeros of F in $\mathbf{D}$.

THEOREM 1.2. ([12, p. 150]) *Let F, $G : \overline{\mathbf{D}} \subset \mathbb{R}^n \to \mathbb{R}^n$ be two continuous functions. If $F(x) = G(x)$ for $x \in \boldsymbol{\partial}\mathbf{D}$, then $\mathrm{d}(F, \mathbf{D}, 0) = \mathrm{d}(G, \mathbf{D}, 0)$.*

Theorem 1.2 states one of the most important properties of degree: the degree depends only on the function values on the boundary.

THEOREM 1.3. ([12, p. 152]) *Let $\alpha = \min\{\|F(x)\|_2 | x \in \boldsymbol{\partial}\mathbf{D}\}$. If*

$$\sup\{\|F(x) - G(x)\|_2 | x \in \overline{\mathbf{D}}\} < \frac{1}{7}\alpha,$$

then

$$\mathrm{d}(F, \mathbf{D}, 0) = \mathrm{d}(G, \mathbf{D}, 0).$$

Theorem 1.3 tells us that small perturbations of F don't change the degree.

THEOREM 1.4. ([12, p. 157]) *Let F, $G : \overline{\mathbf{D}} \subset \mathbb{R}^n \to \mathbb{R}^n$ be two continuous functions. If*

$$0 \notin \{tF(x) + (1-t)G(x) | x \in \boldsymbol{\partial}\mathbf{D} \text{ and } t \in [0,1]\},$$

then

$$\mathrm{d}(F, \mathbf{D}, 0) = \mathrm{d}(G, \mathbf{D}, 0).$$

Theorem 1.4 is the famous Poincaré-Bohl Theorem. It's a particular case of the homotopy invariant property of the topological degree.

Suppose $F : \overline{\mathbf{D}} \subset \mathbb{C}^n \to \mathbb{C}^n$ is analytic, and view the real and imaginary components of F and its argument $z \in \mathbb{C}^n$ as real components in $\mathbb{R}^{2n}$. Let $z = x + iy$ and $F(z) = u(x, y) + iv(x, y)$, where $x = (x_1, \ldots, x_n)$, $y = (y_1, \ldots, y_n)$, $u(x, y) = (u_1(x, y), \ldots, u_n(x, y))$ and $v(x, y) = (v_1(x, y), \ldots, v_n(x, y))$. We define $\tilde{\mathbf{D}}$ by

$$\tilde{\mathbf{D}} \equiv \{(x_1, y_1, \ldots, x_n, y_n) | (x_1 + iy_1, \ldots, x_n + iy_n) \in \mathbf{D}\}$$

and $\tilde{F} : \overline{\tilde{\mathbf{D}}} \subset \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ by $\tilde{F} = (u_1, v_1, \ldots, u_n, v_n)$. Then, we have the following property of topological degree $\mathrm{d}(\tilde{F}, \tilde{\mathbf{D}}, 0)$, and relationships between $\mathrm{d}(\tilde{F}, \tilde{\mathbf{D}}, 0)$ and the solutions of the system $F(z) = 0$ in $\mathbf{D}$.

THEOREM 1.5. ([10]) *Suppose $F : \overline{\mathbf{D}} \subset \mathbb{C}^n \to \mathbb{C}^n$ is analytic, with $F(z) \neq 0$ for any $z \in \boldsymbol{\partial}\mathbf{D}$, and suppose $\tilde{\mathbf{D}}$ and $\tilde{F} : \overline{\tilde{\mathbf{D}}} \to \mathbb{R}^{2n}$ are defined as above. Then*

1. $\mathrm{d}(\tilde{F}, \tilde{\mathbf{D}}, 0) \geq 0$.
2. $\mathrm{d}(\tilde{F}, \tilde{\mathbf{D}}, 0) > 0$ *if and only if there is a solution $z^* \in \mathbf{D}$, $F(z^*) = 0$.*
3. $\mathrm{d}(\tilde{F}, \tilde{\mathbf{D}}, 0)$ *is equal to the number of solutions $z^* \in \mathbf{D}$, $F(z^*) = 0$, counting multiplicities.*
4. *If the Jacobi matrix $F'(z^*)$ is non-singular at every $z^* \in \mathbf{D}$ with $F(z^*) = 0$, then $\mathrm{d}(\tilde{F}, \tilde{\mathbf{D}}, 0)$ is equal to the number of solutions $z^* \in \mathbf{D}$, $F(z^*) = 0$.*

1.3. A Basic Degree Computation Formula. If we let

$$F_{\neg k}(\boldsymbol{x}) \equiv \big(f_1(\boldsymbol{x}), \ldots, f_{k-1}(\boldsymbol{x}), f_{k+1}(\boldsymbol{x}), \ldots, f_n(\boldsymbol{x})\big)$$

and select $s \in \{-1, 1\}$, then $\mathrm{d}(F, \boldsymbol{x}, 0)$ is equal to the number of zeros of $F_{\neg k}$ on $\partial\boldsymbol{x}$ with positive orientation at which $\mathrm{sgn}(f_k) = s$, minus the number of zeros of $F_{\neg k}$ on $\partial\boldsymbol{x}$ with negative orientation at which $\mathrm{sgn}(f_k) = s$. The orientation of each zero can be computed by computing the sign of the determinant of the Jacobian of $F_{\neg k}$ and by taking into account the orientation of the face of $\boldsymbol{x}$ on which the zero lies.

Next, we present a degree computation formula that is similar to a formula used in [10]; see Theorem 2.5 of [10]. We can get the formula in the following theorem by noticing formulas (4.12) and (4.14) in [14] and by taking the orientations of the faces of $\boldsymbol{x}$ into account. We will use this formula to derive the computational procedures in §4.

THEOREM 1.6. *Suppose $F \neq 0$ on $\boldsymbol{\partial x}$, and suppose there is p, $1 \leq p \leq n$, such that:*

1. *$F_{\neg p} \equiv (f_1, \ldots, f_{p-1}, f_{p+1}, \ldots, f_n) \neq 0$ on $\boldsymbol{\partial x}_{\underline{k}}$ or $\boldsymbol{\partial x}_{\overline{k}}$, $k = 1, \ldots, n$; and*
2. *the Jacobi matrices of $F_{\neg p}$ are non-singular at all solutions of $F_{\neg p} = 0$ on $\boldsymbol{\partial x}$.*

Then

$$\mathrm{d}(F, \boldsymbol{x}, 0) = (-1)^{p-1} s \Bigg\{ \sum_{k=1}^{n} (-1)^k \sum_{\substack{x \in \boldsymbol{x}_{\underline{k}} \\ F_{\neg p}(x) = 0 \\ \mathrm{sgn}(f_p(x)) = s}} \mathrm{sgn} \left| \frac{\partial F_{\neg p}}{\partial x_1 x_2 \ldots x_{k-1} x_{k+1} \ldots x_n}(x) \right| + \sum_{k=1}^{n} (-1)^{k+1} \sum_{\substack{x \in \boldsymbol{x}_{\overline{k}} \\ F_{\neg p}(x) = 0 \\ \mathrm{sgn}(f_p(x)) = s}} \mathrm{sgn} \left| \frac{\partial F_{\neg p}}{\partial x_1 x_2 \ldots x_{k-1} x_{k+1} \ldots x_n}(x) \right| \Bigg\},$$

where $s = +1$ or -1.

2. Assumptions and Choice of Box. In this section, we present the basic assumptions. We also introduce how we choose the coordinate bounds $\boldsymbol{x}_i = [\underline{x}_i, \overline{x}_i]$ to satisfy the assumptions and enable more efficient algorithms. When the rank of $F'(x^*)$ is $n-p$ for some $p > 0$, an appropriate preconditioner can be used to reduce $\boldsymbol{F'}(\boldsymbol{x})$ to approximately the pattern shown in Figure 2.1. (See [9] and [10] for details on preconditioning.)

$$
Y\boldsymbol{F'}(\boldsymbol{x}) = \begin{pmatrix}
1 & 0 & \dots & 0 & \overbrace{* \dots *}^{p} \\
0 & 1 & 0\dots & 0 & * \dots * \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & \dots & 0 & 1 & * \dots * \\
0 & \dots & 0 & 0 & 0 \dots 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \dots & 0 & 0 & 0 \dots 0
\end{pmatrix}.
$$

FIG. 2.1. *A preconditioned singular system of rank $n-p$, where "$*$" represents a non-zero element.*

In the analysis to follow, we assume that the system has already been preconditioned, so that it is, to within second-order terms with respect to $\mathrm{w}(\boldsymbol{x})$, of the form in Figure 2.1. Here as in [10], we concentrate on the case p=1.

2.1. The Basic Assumptions. As in the special case $d = 2$ of [10], we assume

1. $F : \overline{\mathbf{D}} \subset \mathbb{R}^n \to \mathbb{R}^n$ can be extended to an analytic function in $\mathbb{C}^n$.
2. $\boldsymbol{x} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_n) = ([\underline{x}_1, \overline{x}_1], \dots, [\underline{x}_n, \overline{x}_n])$ is a small box constructed to be centered at an approximate solution $\check{x}$, i.e. $\mathrm{m}(\boldsymbol{x}) = (\check{x}_1, \dots, \check{x}_n)$.
3. $\check{x}$ is near a point x^* with $F(x^*) = 0$, such that $\|\check{x} - x^*\|$ is much smaller than the width of the box $\boldsymbol{x}$, and width of the box $\boldsymbol{x}$ is small enough so that mean value interval extensions lead, after preconditioning, to a system like Figure 2.1, with small intervals replacing the zeros.
4. F has been preconditioned as in Figure 2.1, and $F'(x^*)$ has null space of dimension 1.

Denote

$$
\begin{aligned}
\alpha_k &\equiv \frac{\partial f_k}{\partial x_n}(\check{x}), \qquad 1 \le k \le n-1, \\
\alpha_n &\equiv -1, \\
\Delta_1 &\equiv \left| \frac{\partial F}{\partial x_1 \dots \partial x_n}(\check{x}) \right| \\
\Delta_d &\equiv \sum_{k_1=1}^{n} \dots \sum_{k_d=1}^{n} \frac{\partial^d f_n}{\partial x_{k_1} \dots \partial x_{k_d}}(\check{x}) \alpha_{k_1} \dots \alpha_{k_d}, \qquad 2 \le d.
\end{aligned}
$$

The following representation of $f(x)$ near $\check{x}$ is appropriate under these assumptions.

$$
f_k(x) = (x_k - \check{x}_k) + \alpha_k (x_n - \check{x}_n) + \mathcal{O}\left(\|x - \check{x}\|^2\right) \quad \text{for } 1 \le k \le n-1. \tag{2.1}
$$

$$
f_n(x) = \frac{1}{2!} \sum_{k_1=1}^{n} \sum_{k_2=1}^{n} \frac{\partial^2 f_n}{\partial x_{k_1} \partial x_{k_2}}(\check{x})(x_{k_1} - \check{x}_{k_1})(x_{k_2} - \check{x}_{k_2}) + \dots \tag{2.2}
$$

$$+\frac{1}{d!}\sum_{k_1=1}^{n}\dots\sum_{k_d=1}^{n}\frac{\partial^d f_n}{\partial x_{k_1}\dots\partial x_{k_d}}(\check{x})(x_{k_1}-\check{x}_{k_1})\dots(x_{k_d}-\check{x}_{k_d})$$
$$+\mathcal{O}\left(\|x-\check{x}\|^{d+1}\right),$$

or

$$f_k(x)\approx(x_k-\check{x}_k)+\alpha_k(x_n-\check{x}_n)\quad\text{for } 1\le k\le n-1. \tag{2.3}$$

$$f_n(x)\approx\frac{1}{2!}\sum_{k_1=1}^{n}\sum_{k_2=1}^{n}\frac{\partial^2 f_n}{\partial x_{k_1}\partial x_{k_2}}(\check{x})(x_{k_1}-\check{x}_{k_1})(x_{k_2}-\check{x}_{k_2})+\dots \tag{2.4}$$
$$+\frac{1}{d!}\sum_{k_1=1}^{n}\dots\sum_{k_d=1}^{n}\frac{\partial^d f_n}{\partial x_{k_1}\dots\partial x_{k_d}}(\check{x})(x_{k_1}-\check{x}_{k_1})\dots(x_{k_d}-\check{x}_{k_d}).$$

For $F:\mathbb{R}^n\to\mathbb{R}^n$, extend F to complex space: $x+iy$, with y in a small box $\boldsymbol{y}=\big(\boldsymbol{y}_1,\dots,\boldsymbol{y}_n\big)=\big([\underline{y}_1,\overline{y}_1],\dots,[\underline{y}_n,\overline{y}_n]\big)$, where $\boldsymbol{y}$ is centered at $(0,\dots,0)$. Define $\tilde{\boldsymbol{x}}\equiv(\boldsymbol{x}_1,\boldsymbol{y}_1,\dots,\boldsymbol{x}_n,\boldsymbol{y}_n)=([\underline{x}_1,\overline{x}_1],[\underline{y}_1,\overline{y}_1],\dots,[\underline{x}_n,\overline{x}_n],[\underline{y}_n,\overline{y}_n])$, $u_k(x,y)\equiv\Re(f_k(x+iy))$ and $v_k(x,y)\equiv\Im(f_k(x+iy))$. With this, define

$$\tilde{F}(x,y)\equiv(u_1(x,y),v_1(x,y),\dots,u_n(x,y),v_n(x,y)):\mathbb{R}^{2n}\to\mathbb{R}^{2n}.$$

Also define

$$\tilde{F}_{\neg u_n}(x,y)\equiv\big(u_1(x,y),v_1(x,y),\dots,u_{n-1}(x,y),v_{n-1}(x,y),v_n(x,y)\big).$$

Then, based on (2.1) and (2.2), for $1\le k\le(n-1)$,

$$\left.\begin{array}{rcl} u_k(x,y) & = & (x_k-\check{x}_k)+\alpha_k(x_n-\check{x}_n)\\ & & +\mathcal{O}\left(\|(x-\check{x},y)\|^2\right),\\ v_k(x,y) & = & y_k+\alpha_k y_n+\mathcal{O}\left(\|(x-\check{x},y)\|^2\right), \end{array}\right\} \tag{2.5}$$

or

$$\left.\begin{array}{rcl} u_k(x,y) & \approx & (x_k-\check{x}_k)+\alpha_k(x_n-\check{x}_n),\\ v_k(x,y) & \approx & y_k+\alpha_k y_n. \end{array}\right\} \tag{2.6}$$

2.2. Choosing the Coordinate Bounds. We use a similar scheme to that of §5 of [10]. In particular, having defined $\boldsymbol{x}_{\underline{k}}$ and $\boldsymbol{x}_{\overline{k}}$ in §1.1, we define $\boldsymbol{y}_{\underline{k}}$ and $\boldsymbol{y}_{\overline{k}}$ similarly:

$$\boldsymbol{y}_{\underline{k}}\equiv(\boldsymbol{x}_1,\boldsymbol{y}_1,\dots,\boldsymbol{x}_{k-1},\boldsymbol{y}_{k-1},\boldsymbol{x}_k,\underline{y}_k,\boldsymbol{x}_{k+1},\boldsymbol{y}_{k+1},\dots,\boldsymbol{x}_n,\boldsymbol{y}_n)\quad\text{and}$$
$$\boldsymbol{y}_{\overline{k}}\equiv(\boldsymbol{x}_1,\boldsymbol{y}_1,\dots,\boldsymbol{x}_{k-1},\boldsymbol{y}_{k-1},\boldsymbol{x}_k,\overline{y}_k,\boldsymbol{x}_{k+1},\boldsymbol{y}_{k+1},\dots,\boldsymbol{x}_n,\boldsymbol{y}_n).$$

To compute the degree $\mathrm{d}(\tilde{F},\tilde{\boldsymbol{x}},0)$, we consider $\tilde{F}_{\neg u_n}$ on the boundary of $\tilde{\boldsymbol{x}}$. This boundary consists of the $4n$ faces $\boldsymbol{x}_{\underline{1}}$, $\boldsymbol{x}_{\overline{1}}$, $\boldsymbol{y}_{\underline{1}}$, $\boldsymbol{y}_{\overline{1}}$, …, $\boldsymbol{x}_{\underline{n}}$, $\boldsymbol{x}_{\overline{n}}$, $\boldsymbol{y}_{\underline{n}}$, $\boldsymbol{y}_{\overline{n}}$. We set $\boldsymbol{x}_n$ and $\boldsymbol{y}_n$ in such a way that

$$\mathrm{w}(\boldsymbol{x}_n)\le\frac{1}{2}\min_{1\le k\le n-1}\left\{\frac{\mathrm{w}(\boldsymbol{x}_k)}{|\alpha_k|}\right\}\quad\text{and}\quad\mathrm{w}(\boldsymbol{y}_n)\le\frac{1}{2}\min_{1\le k\le n-1}\left\{\frac{\mathrm{w}(\boldsymbol{y}_k)}{|\alpha_k|}\right\}. \tag{2.7}$$

Constructing the box widths this way makes it is unlikely that $u_k(x,y)=0$ on either $\boldsymbol{x}_{\underline{k}}$ or $\boldsymbol{x}_{\overline{k}}$ and unlikely that $v_k(x,y)=0$ on either $\boldsymbol{y}_{\underline{k}}$ or $\boldsymbol{y}_{\overline{k}}$, where $k=1,\dots,n-1$.

This, in turn, allows us to replace searches on $4n-4$ of the $4n$ faces of $\boldsymbol{\partial\tilde{x}}$ by simple interval evaluations, reducing the total computational cost dramatically. See [10] for details.

A difference between the scheme used here and that of [10] is the way the ratio $\mathrm{w}(\boldsymbol{y}_n)/\mathrm{w}(\boldsymbol{x}_n)$ is chosen. In [10], $\mathrm{w}(\boldsymbol{y}_n)$ was chosen large relative to $\boldsymbol{x}_n$, to arrange no solutions of $u_n = 0$ on $\boldsymbol{y}_{\underline{n}}$ and $\boldsymbol{y}_{\overline{n}}$. When the degree is odd, that is not possible, and we have found the strategy represented by formula (4.3) below, implying $\mathrm{w}(\boldsymbol{y}_n)$ small relative to $\mathrm{w}(\boldsymbol{x}_n)$ as in Figure 4.1 below, to be more convenient.

3. A Theorem. In [10] we proved that, under the assumptions in §2, if (2.3) and (2.4) are exact for $d = 2$, and if $\Delta_1 = 0$ but $\Delta_2 \neq 0$, then $\mathrm{d}(\tilde{F}, \boldsymbol{\tilde{x}}, 0) = 2$. Here, we generalize that result to $\Delta_1 = \ldots = \Delta_{d-1} = 0$, $\Delta_d \neq 0$.

THEOREM 3.1. *Suppose*

1. *all the assumptions in §2 are true;*
2. *(2.3) and (2.4) are exact; and*
3. $\Delta_1 = \ldots = \Delta_{d-1} = 0$, $\Delta_d \neq 0$, *where* $2 \leq d$.

Then $\mathrm{d}(\tilde{F}, \boldsymbol{\tilde{x}}, 0) = d$.

In contrast to the proof in [10], we use a homotopy argument to prove Theorem 3.1.

Proof. Let $z = (z_1, \ldots, z_n) = (x_1 + iy_1, \ldots, x_n + iy_n)$. Then

$$F(z) = (f_1(z), \ldots, f_{n-1}(z), f_n(z)),$$

where

$$\begin{aligned} f_k(z) &= (z_k - \check{z}_k) + \frac{\partial f_k}{\partial x_n}(\check{x})(z_n - \check{z}_n) \\ &\qquad \text{for } 1 \leq k \leq n-1, \\ f_n(z) &= \frac{1}{2!}\sum_{k_1=1}^{n}\sum_{k_2=1}^{n} \frac{\partial^2 f_n}{\partial x_{k_1}\partial x_{k_2}}(\check{x})(z_{k_1} - \check{z}_{k_1})(z_{k_2} - \check{z}_{k_2}) + \ldots \\ &\quad + \frac{1}{d!}\sum_{k_1=1}^{n}\cdots\sum_{k_d=1}^{n} \frac{\partial^2 f_n}{\partial x_{k_1}\ldots\partial x_{k_d}}(\check{x})(z_{k_1} - \check{z}_{k_1})\ldots(z_{k_d} - \check{z}_{k_d}). \end{aligned} \tag{3.1}$$

We construct $G : \mathbb{C}^n \to \mathbb{C}^n$ by

$$G(z) = (g_1(z), \ldots, g_{n-1}(z), g_n(z)),$$

where

$$\begin{aligned} g_k(z) &= (z_k - \check{z}_k) \quad \text{for } 1 \leq k \leq n-1, \\ g_n(z) &= \frac{(-1)^d \Delta_d}{d!}(z_n - \check{z}_n)^d. \end{aligned} \tag{3.2}$$

Let $p_k(x, y) \equiv \Re(g_k(x + iy))$ and $q_k(x, y) \equiv \Im(g_k(x + iy))$. With this, define $\tilde{G} : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ by

$$\tilde{G}(x, y) \equiv (p_1(x, y), q_1(x, y), \ldots, p_n(x, y), q_n(x, y)).$$

We will first prove $\mathrm{d}(\tilde{F}, \boldsymbol{\tilde{x}}, 0) = \mathrm{d}(\tilde{G}, \boldsymbol{\tilde{x}}, 0)$. Define

$$\begin{aligned} \tilde{H}((x, y), t) &\equiv t\tilde{F}(x, y) + (1 - t)\tilde{G}(x, y) \\ \text{and} \qquad H(z, t) &\equiv tF(z) + (1 - t)G(z). \end{aligned}$$

We will prove that $\tilde{H}((x,y),t) \neq 0$ when $(x,y) \in \partial\tilde{\boldsymbol{x}}$ and $t \in [0,1]$. It's clear that $\tilde{H}((x,y),t) = 0$ is equivalent to $H(z,t) = 0$, so we consider $H(z,t)$.

$$
\begin{aligned}
H(z,t) &= \Big(H_1(z,t), \ldots, H_n(z,t)\Big) \\
&= \Big(tf_1(z) + (1-t)g_1(z), \ldots, tf_{n-1}(z) + (1-t)g_{n-1}(z), \\
&\quad tf_n(z) + (1-t)g_n(z)\Big) \\
&= \Big(t((z_1 - \check{z}_1) + \alpha_1(z_n - \check{z}_n)) + (1-t)(z_1 - \check{z}_1), \ldots, \\
&\quad t((z_{n-1} - \check{z}_{n-1}) + \alpha_{n-1}(z_n - \check{z}_n)) + (1-t)(z_{n-1} - \check{z}_{n-1}), \\
&\quad tf_n(z) + (1-t)\frac{(-1)^d \Delta_d}{d!}(z_n - \check{z}_n)^d\Big) \\
&= \Big((z_1 - \check{z}_1) + t\alpha_1(z_n - \check{z}_n), \ldots, (z_{n-1} - \check{z}_{n-1}) + t\alpha_{n-1}(z_n - \check{z}_n), \\
&\quad tf_n(z) + (1-t)\frac{(-1)^d \Delta_d}{d!}(z_n - \check{z}_n)^d\Big)
\end{aligned}
$$

Thus, $H(z,t) = 0$ implies $z_k = \check{z}_k - t\alpha_k(z_n - \check{z}_n)$ for $k = 1, \ldots, n-1$. Plugging $z_k - \check{z}_k = -t\alpha_k(z_n - \check{z}_n)$ for each such k into (3.1), we get

$$
\begin{aligned}
f_n(z) &= \frac{1}{2!}\sum_{k_1=1}^{n}\sum_{k_2=1}^{n} \frac{\partial^2 f_n}{\partial x_{k_1}\partial x_{k_2}}(\check{x})(-1)^2 t^2 \alpha_{k_1}\alpha_{k_2}(z_n - \check{z}_n)^2 + \ldots + \\
&\quad \frac{1}{d!}\sum_{k_1=1}^{n}\cdots\sum_{k_d=1}^{n} \frac{\partial^2 f_n}{\partial x_{k_1}\ldots\partial x_{k_d}}(\check{x})(-1)^d t^d \alpha_{k_1}\ldots\alpha_{k_d}(z_n - \check{z}_n)^d \\
&= \frac{(-1)^2 t^2}{2!}(z_n - \check{z}_n)^2 \sum_{k_1=1}^{n}\sum_{k_2=1}^{n} \frac{\partial^2 f_n}{\partial x_{k_1}\partial x_{k_2}}(\check{x})\alpha_{k_1}\alpha_{k_2} + \ldots \\
&\quad + \frac{(-1)^d t^d}{d!}(z_n - \check{z}_n)^d \sum_{k_1=1}^{n}\cdots\sum_{k_d=1}^{n} \frac{\partial^2 f_n}{\partial x_{k_1}\ldots\partial x_{k_d}}(\check{x})\alpha_{k_1}\ldots\alpha_{k_d} \\
&= \frac{(-1)^2 t^2 \Delta_2}{2!}(z_n - \check{z}_n)^2 + \ldots + \frac{(-1)^{d-1} t^{d-1}\Delta_{d-1}}{(d-1)!}(z_n - \check{z}_n)^{d-1} \\
&\quad + \frac{(-1)^d t^d \Delta_d}{d!}(z_n - \check{z}_n)^d \\
&= \frac{(-1)^d t^d \Delta_d}{d!}(z_n - \check{z}_n)^d.
\end{aligned}
$$

Thus, the last component of $H(z,t)$ is:

$$
\begin{aligned}
&tf_n(z) + (1-t)\frac{(-1)^d \Delta_d}{d!}(z_n - \check{z}_n)^d \\
&= \frac{(-1)^d t^{d+1}\Delta_d}{d!}(z_n - \check{z}_n)^d + (1-t)\frac{(-1)^d \Delta_d}{d!}(z_n - \check{z}_n)^d \\
&= \frac{(-1)^d (1 - t + t^{d+1})\Delta_d}{d!}(z_n - \check{z}_n)^d.
\end{aligned}
$$

When $t = 0$ and $t = 1$, $1 - t + t^{d+1} = 1$. When $t \in (0,1)$, $1 - t + t^{d+1} > t^{d+1} > 0$. Thus, $1 - t + t^{d+1} \neq 0$ for $t \in [0,1]$. Then, $H(z,t) = 0$ implies $(z_n - \check{z}_n)^d = 0$, and consequently, $z_n - \check{z}_n = 0$ or $z_n = \check{z}_n$.

Now we know $H(z,t)$ has a unique zero at $(\tilde{z}_1,\dots,\tilde{z}_{n-1},\tilde{z}_n) = (\check{z}_1 - t\alpha_1(z_n - \check{z}_n),\dots,\check{z}_{n-1} - t\alpha_{n-1}(z_n - \check{z}_n),\check{z}_n)$. Accordingly, $\tilde{H}((x,y),t)$ has a unique zero at $(\tilde{x}_1,\tilde{y}_1,\dots,\tilde{x}_{n-1},\tilde{y}_{n-1},\tilde{x}_n,\tilde{y}_n) = (\check{x}_1 - t\alpha_1(x_n - \check{x}_n),(\check{y}_1 - t\alpha_1(y_n - \check{y}_n),\dots,\check{x}_{n-1} - t\alpha_{n-1}(x_n - \check{x}_n),\check{y}_{n-1} - t\alpha_{n-1}(y_n - \check{y}_n),\check{x}_n,\check{y}_n)$. Based on the way we have constructed $\boldsymbol{x}_n$ and $\boldsymbol{y}_n$, we have

$$\begin{aligned} &|\tilde{x}_k - \check{x}_k| = |t\alpha_k(x_n - \check{x}_n)| \le |\alpha_k(x_n - \check{x}_n)| \le |\alpha_k|\frac{\mathrm{w}(\boldsymbol{x}_n)}{2} < \frac{\mathrm{w}(\boldsymbol{x}_k)}{2} \\ \text{and}\quad &|\tilde{y}_k - \check{y}_k| = |t\alpha_k(y_n - \check{y}_n)| \le |\alpha_k(y_n - \check{y}_n)| \le |\alpha_k|\frac{\mathrm{w}(\boldsymbol{y}_n)}{2} < \frac{\mathrm{w}(\boldsymbol{y}_k)}{2}, \end{aligned}$$

where $k = 1,\dots,n-1$. Thus, $\tilde{x}_k \notin \boldsymbol{\partial x}_k$ and $\tilde{y}_k \notin \boldsymbol{\partial y}_k$ for $k = 1,\dots,n-1$. Obviously, $\tilde{x}_n = \check{x}_n \notin \boldsymbol{\partial x}_n$ and $\tilde{y}_n = \check{y}_n \notin \boldsymbol{\partial y}_n$. This implies $\tilde{H}((x,y),t) \neq 0$ for $(x,y) \in \boldsymbol{\partial \tilde{x}}$ and $t \in [0,1]$. Then, by Theorem 1.4,

$$\mathrm{d}(\tilde{F},\tilde{\boldsymbol{x}},0) = \mathrm{d}(\tilde{G},\tilde{\boldsymbol{x}},0).$$

Next, we prove $\mathrm{d}(\tilde{G},\tilde{\boldsymbol{x}},0) = d$. Perturb $G(z)$ by an arbitrary small ϵ to define

$$G_\epsilon(z) = (g_{1\epsilon}(z),\dots,g_{(n-1)\epsilon}(z),g_{n\epsilon}(z)),$$

where

$$\begin{aligned} g_{k\epsilon}(z) &= g_k(z) = (z_k - \check{z}_k) \quad \text{for } 1 \le k \le n-1, \\ g_{n\epsilon}(z) &= g_n(z) + \epsilon = \frac{(-1)^d \Delta_d}{d!}(z_n - \check{z}_n)^d + \epsilon. \end{aligned} \tag{3.3}$$

Let $p_{k\epsilon}(x,y) \equiv \Re(g_{k\epsilon}(x+iy))$ and $q_{k\epsilon}(x,y) \equiv \Im(g_{k\epsilon}(x+iy))$. With this, define

$$\tilde{G}_\epsilon(x,y) \equiv (p_{1\epsilon}(x,y),q_{1\epsilon}(x,y),\dots,p_{n\epsilon}(x,y),q_{n\epsilon}(x,y)).$$

It is obvious that $p_{k\epsilon}(x,y) = x_k - \check{x}_k$ and $q_{k\epsilon}(x,y) = y_k - \check{y}_k$ for $k = 1,\dots,n-1$. Assume ϵ is small enough. Then, $G_\epsilon(z)$, and thus $\tilde{G}_\epsilon(x,y)$, have d zeros $\tilde{z} = (\tilde{z}_1,\dots,\tilde{z}_{n-1},\tilde{z}_n)$, or $\tilde{x} = (\tilde{x}_1,\tilde{y}_1,\dots,\tilde{x}_{n-1},\tilde{y}_{n-1},\tilde{x}_n,\tilde{x}_n)$ in $\tilde{\boldsymbol{x}}$, with $\tilde{z}_k - \check{z}_k = 0$, or $\tilde{x}_k - \check{x}_k = 0$ and $\tilde{y}_k - \check{y}_k = 0$ for $k = 1,\dots,n-1$, and $(\tilde{z}_n - \check{z}_n)^d = \frac{d!\epsilon}{(-1)^{d+1}\Delta_d} \neq 0$. $\frac{\partial g_{n\epsilon}}{\partial z_n}(\tilde{z}) = \frac{(-1)^d\Delta_d}{(d-1)!}(\tilde{z}_n - \check{z}_n)^{d-1} \neq 0$.

$$\begin{aligned} \left|\frac{\partial \tilde{G}_\epsilon}{\partial x_1 \partial y_1 \dots \partial x_n \partial y_n}(\tilde{x})\right| &= \begin{vmatrix} 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \\ 0 & 0 & \dots & 1 & 0 & 0 \\ 0 & 0 & \dots & 0 & \frac{\partial p_{n\epsilon}}{\partial x_n} & \frac{\partial p_{n\epsilon}}{\partial y_n} \\ 0 & 0 & \dots & 0 & \frac{\partial q_{n\epsilon}}{\partial x_n} & \frac{\partial q_{n\epsilon}}{\partial y_n} \end{vmatrix} \\ &= \begin{vmatrix} \frac{\partial p_{n\epsilon}}{\partial x_n} & \frac{\partial p_{n\epsilon}}{\partial y_n} \\ \frac{\partial q_{n\epsilon}}{\partial x_n} & \frac{\partial q_{n\epsilon}}{\partial y_n} \end{vmatrix} = \begin{vmatrix} \frac{\partial p_{n\epsilon}}{\partial x_n} & \frac{\partial p_{n\epsilon}}{\partial y_n} \\ -\frac{\partial p_{n\epsilon}}{\partial y_n} & \frac{\partial p_{n\epsilon}}{\partial x_n} \end{vmatrix} \\ &= \left(\frac{\partial p_{n\epsilon}}{\partial x_n}\right)^2 + \left(\frac{\partial p_{n\epsilon}}{\partial y_n}\right)^2 = \left|\frac{\partial g_{n\epsilon}}{\partial z_n}(\tilde{z})\right|^2 > 0. \end{aligned} \tag{3.4}$$

Thus, by Theorem 1.1, $\mathrm{d}(\tilde{G}_\epsilon,\tilde{\boldsymbol{x}},0) = d$, and then $\mathrm{d}(\tilde{G},\tilde{\boldsymbol{x}},0) = d$ by Theorem 1.3. Finally,

$$\mathrm{d}(\tilde{F},\tilde{\boldsymbol{x}},0) = \mathrm{d}(\tilde{G},\tilde{\boldsymbol{x}},0) = d.$$

□

Unless the components of F are exactly linear and degree d polynomials, the equalities (2.3) and (2.4) in §2 are not exact, but are only approximately true to second order. However, if second-order approximations are accurate, we can expect the degree to be equal to d. The disadvantage of the proof of Theorem 3.1 is that it does not lead to a practical computational technique. If we try to verify $H(z,t) \neq 0$ or $\tilde{H}((x,y),t) \neq 0$ when $(x,y) \in \partial\tilde{\boldsymbol{x}}$ and $t \in [0,1]$, then it would require an inordinate amount of work for a verification process that would normally require only a single step of an interval Newton method in the nonsingular case. First, we would need to compute Δ_d, which involves all partial derivatives of order 1 and order d. This is expensive when both n and d are large. Second, we would need to know where the solutions of $u_n(x) = 0$ and $v_n(x) = 0$ are on $\boldsymbol{x}_{\underline{n}}$, $\boldsymbol{x}_{\overline{n}}$, $\boldsymbol{y}_{\underline{n}}$ and $\boldsymbol{y}_{\overline{n}}$ when $z_k = \check{z}_k - t\alpha_k(z_n - \check{z}_n)$, and the search process for such solutions is expensive.

We could try to verify $H(z,t) \neq 0$ when $(x,y) \in \partial\tilde{\boldsymbol{x}}$ and $t \in [0,1]$ in another way: verify $H(z,t) = 0$ has a unique solution in the interior of $\tilde{\boldsymbol{x}}$ when $t \in [0,1]$. However, we will run into the singular situation again if we do that.

In fact, there is an alternative algorithm to compute the degree. That will be the subject of next section.

4. Algorithm. The algorithm we present here is similar to the algorithm in [10]. Based on Theorem 1.6 in §1.2, the following theorem underlies our algorithm.

Theorem 4.1. *Suppose*

1. $u_k \neq 0$ *on* $\boldsymbol{x}_{\underline{k}}$ *and* $\boldsymbol{x}_{\overline{k}}$, *and* $v_k \neq 0$ *on* $\boldsymbol{y}_{\underline{k}}$ *and* $\boldsymbol{y}_{\overline{k}}$, $k = 1, \ldots, n-1$;
2. $\tilde{F}_{\neg u_n} = 0$ *has solutions, if there are any, on* $\boldsymbol{x}_{\underline{n}}$ *and* $\boldsymbol{x}_{\overline{n}}$ *with* y_n *in the interior of* $\boldsymbol{y}_n$, *and* $\tilde{F}_{\neg u_n} = 0$ *has solutions, if there are any, on* $\boldsymbol{y}_{\underline{n}}$ *and* $\boldsymbol{y}_{\overline{n}}$ *with* x_n *in the interior of* $\boldsymbol{x}_n$;
3. $u_n \neq 0$ *at the solutions of* $\tilde{F}_{\neg u_n} = 0$ *in condition 2; and*
4. *the Jacobi matrices of* $\tilde{F}_{\neg u_n}$ *are non-singular at the solutions of* $\tilde{F}_{\neg u_n} = 0$ *in condition 2.*

Then, for a fixed $s \in \{-1, 1\}$,

$$
\begin{aligned}
\mathrm{d}(\tilde{F}, \tilde{\boldsymbol{x}}, 0) = &-s \sum_{\substack{x_n = \underline{x}_n \\ \tilde{F}_{\neg u_n}(x,y)=0 \\ \mathrm{sgn}(u_n(x,y))=s}} \mathrm{sgn}\left|\frac{\partial \tilde{F}_{\neg u_n}}{\partial x_1 y_1 \ldots x_{n-1} y_{n-1} y_n}(x,y)\right| \\
&+s \sum_{\substack{x_n = \overline{x}_n \\ \tilde{F}_{\neg u_n}(x,y)=0 \\ \mathrm{sgn}(u_n(x,y))=s}} \mathrm{sgn}\left|\frac{\partial \tilde{F}_{\neg u_n}}{\partial x_1 y_1 \ldots x_{n-1} y_{n-1} y_n}(x,y)\right| \\
&+s \sum_{\substack{y_n = \underline{y}_n \\ \tilde{F}_{\neg u_n}(x,y)=0 \\ \mathrm{sgn}(u_n(x,y))=s}} \mathrm{sgn}\left|\frac{\partial \tilde{F}_{\neg u_n}}{\partial x_1 y_1 \ldots x_{n-1} y_{n-1} x_n}(x,y)\right| \\
&-s \sum_{\substack{y_n = \overline{y}_n \\ \tilde{F}_{\neg u_n}(x,y)=0 \\ \mathrm{sgn}(u_n(x,y))=s}} \mathrm{sgn}\left|\frac{\partial \tilde{F}_{\neg u_n}}{\partial x_1 y_1 \ldots x_{n-1} y_{n-1} x_n}(x,y)\right|.
\end{aligned}
$$

Proof. Condition 1 implies $\tilde{F} \neq 0$ on $\boldsymbol{x}_{\underline{k}}$, $\boldsymbol{x}_{\overline{k}}$, $\boldsymbol{y}_{\underline{k}}$ and $\boldsymbol{y}_{\overline{k}}$, $k = 1, \ldots, n-1$, and conditions 2 and 3 imply $\tilde{F} \neq 0$ on $\boldsymbol{x}_{\underline{n}}$, $\boldsymbol{x}_{\overline{n}}$, $\boldsymbol{y}_{\underline{n}}$ and $\boldsymbol{y}_{\overline{n}}$. Thus, $\tilde{F} \neq 0$ on $\boldsymbol{\partial}\tilde{\boldsymbol{x}}$. Now, condition 1 implies $\tilde{F}_{\neg u_n} \neq 0$ on $\boldsymbol{\partial x}_{\underline{k}}$, $\boldsymbol{\partial x}_{\overline{k}}$, $\boldsymbol{\partial y}_{\underline{k}}$ and $\boldsymbol{\partial y}_{\overline{k}}$, $k = 1, \ldots, n-1$. $\boldsymbol{\partial x}_{\underline{n}}$ consists of $2(n-1)$ $(2n-2)$-dimensional boxes, each of which is either embedded in some $\boldsymbol{x}_{\underline{k}}$, $\boldsymbol{x}_{\overline{k}}$, $\boldsymbol{y}_{\underline{k}}$ or $\boldsymbol{y}_{\overline{k}}$, $1 \leq k \leq n-1$ or is embedded in $\boldsymbol{\partial y}_{\underline{n}}$ or $\boldsymbol{\partial y}_{\overline{n}}$. Thus, by 1 and 2, $\tilde{F}_{\neg u_n} \neq 0$ on $\boldsymbol{\partial x}_{\underline{n}}$. Similarly, $\tilde{F}_{\neg u_n} \neq 0$ on $\boldsymbol{\partial x}_{\overline{n}}$, $\boldsymbol{\partial y}_{\underline{n}}$ and $\boldsymbol{\partial y}_{\overline{n}}$. Thus, condition 1 in Theorem 1.6 is satisfied. Finally, with condition 4, all the conditions of Theorem 1.6 are satisfied. The formula is thus obtained. □

By constructing the box $\tilde{\boldsymbol{x}}$ according to (2.7), we can verify $u_k \neq 0$ on $\boldsymbol{x}_{\underline{k}}$ and $\boldsymbol{x}_{\overline{k}}$, and $v_k \neq 0$ on $\boldsymbol{y}_{\underline{k}}$ and $\boldsymbol{y}_{\overline{k}}$, $k = 1, \ldots, n-1$, since $u_k(x,y) \approx (x_k - \check{x}_k) + \alpha_k(x_n - \check{x}_n) \neq 0$ on $\boldsymbol{x}_{\underline{k}}$ and $\boldsymbol{x}_{\overline{k}}$, and $v_k(x,y) \approx y_k + \alpha_k y_n \neq 0$ on $\boldsymbol{y}_{\underline{k}}$ and $\boldsymbol{y}_{\overline{k}}$. This only needs $4n-4$ interval evaluations. Then, we only need to search the four faces $\boldsymbol{x}_{\underline{n}}, \boldsymbol{x}_{\overline{n}}, \boldsymbol{y}_{\underline{n}}$ and $\boldsymbol{y}_{\overline{n}}$ for solutions of $\tilde{F}_{\neg u_n}(x,y) = 0$, regardless of how large n is, thus dramatically reducing the total computational cost. The four faces $\boldsymbol{x}_{\underline{n}}$, $\boldsymbol{x}_{\overline{n}}$, $\boldsymbol{y}_{\underline{n}}$ and $\boldsymbol{y}_{\overline{n}}$ remaining to be searched are $(2n-1)$-dimensional boxes. However, exploitation of (2.5) will reduce the search of solutions of $\tilde{F}_{\neg u_n}(x,y) = 0$ on the $(2n-1)$-dimensional boxes to actually a one dimensional search. We use $\boldsymbol{x}_{\underline{n}}$ as an example to explain this.

On $\boldsymbol{x}_{\underline{n}}$, $x_n = \underline{x}_n$. We know from (2.5) that if x_n is known precisely, formally solving $\boldsymbol{u}_k(\boldsymbol{x},\boldsymbol{y}) = 0$ for x_k gives sharper bounds $\tilde{\boldsymbol{x}}_k$ with $\mathrm{w}(\tilde{\boldsymbol{x}}_k) = \mathcal{O}\left(\|(\boldsymbol{x} - \check{x}, \boldsymbol{y})\|^2\right)$, $1 \leq k \leq n-1$. Then, we can divide $\boldsymbol{y}_n$ into smaller subintervals. For a small subinterval $\boldsymbol{y}_n^0$ of $\boldsymbol{y}_n$, we can formally solve $\boldsymbol{v}_k(\boldsymbol{x},\boldsymbol{y}) = 0$ for y_k to get sharper bounds $\tilde{\boldsymbol{y}}_k$ with $\mathrm{w}(\tilde{\boldsymbol{y}}_k) = \mathcal{O}\left(\max(\|(\boldsymbol{x} - \check{x}, \boldsymbol{y})\|^2, \|\boldsymbol{y}_n^0\|)\right)$, $1 \leq k \leq n-1$. Thus, we have reduced the search to searching the one dimensional interval $\boldsymbol{y}_n$, much less costly than searching a $(2n-1)$-dimensional box when n is large. Furthermore, if we know approximately where the solutions of $\tilde{F}_{\neg u_n}(x,y) = 0$ are, we can even reduce the cost of the one dimensional search. To this end, we will next analyze the solutions of $\tilde{F}_{\neg u_n}(x,y) = 0$ on the four faces $\boldsymbol{x}_{\underline{n}}$, $\boldsymbol{x}_{\overline{n}}$, $\boldsymbol{y}_{\underline{n}}$ and $\boldsymbol{y}_{\overline{n}}$.

For convenience of analysis, assume (2.3) and (2.4) are exact. In other words, for $1 \leq k \leq (n-1)$,

$$\left.\begin{array}{rcl} u_k(x,y) & = & (x_k - \check{x}_k) + \alpha_k(x_n - \check{x}_n), \\ v_k(x,y) & = & y_k + \alpha_k y_n. \end{array}\right\} \tag{4.1}$$

and

$$\begin{aligned} f_n(z) = \frac{1}{2!} \sum_{k_1=1}^{n} \sum_{k_2=1}^{n} \frac{\partial^2 f_n}{\partial x_{k_1} \partial x_{k_2}}(\check{x})(z_{k_1} - \check{z}_{k_1})(z_{k_2} - \check{z}_{k_2}) + \ldots + \\ \frac{1}{d!} \sum_{k_1=1}^{n} \cdots \sum_{k_d=1}^{n} \frac{\partial^d f_n}{\partial x_{k_1} \ldots \partial x_{k_d}}(\check{x})(z_{k_1} - \check{z}_{k_1}) \ldots (z_{k_d} - \check{z}_{k_d}). \end{aligned}$$

From this, $\tilde{F}_{\neg u_n}(x,y) = 0$ implies $u_k(x,y) = 0$ and $v_k(x,y) = 0$, i.e. $f_k(z) = 0$, whence $z_k = \check{z}_k - \alpha_k(z_n - \check{z}_n)$ for $k = 1, \ldots, n-1$. In terms of the real and imaginary coordinates,

$$\left.\begin{array}{rcl} x_k & = & \check{x}_k - \alpha_k(x_n - \check{x}_n), \\ y_k & = & -\alpha_k y_n, \end{array}\right\} \tag{4.2}$$

for $k = 1, \ldots, n-1$. Plugging $z_k - \check{z}_k = -\alpha_k(z_n - \check{z}_n)$, $k = 1, \ldots, n-1$, into $f_n(z)$ as in the proof of Theorem 3.1 then gives

$$f_n(z) = \frac{(-1)^d \Delta_d}{d!}(z_n - \check{z}_n)^d.$$

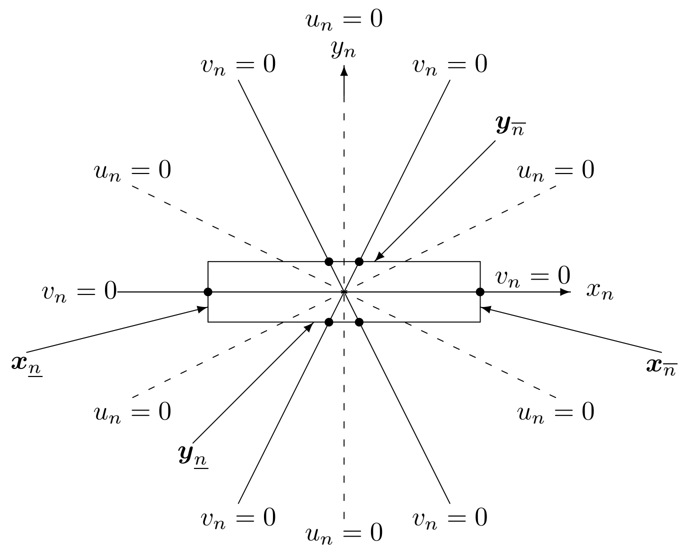

FIG. 4.1. *When d is odd. Here, $d = 3$. $v_n = 0$ on solid lines and $u_n = 0$ on dashed lines. The thick dots are the solutions of $\tilde{F}_{\neg u_n}(x, y) = 0$ on $\boldsymbol{\partial}\tilde{\boldsymbol{x}}$.*

Thus, $u_n(x, y) = \Re(f_n(z)) = \frac{(-1)^d \Delta_d}{d!} \Re((z_n - \check{z}_n)^d)$ and $v_n(x, y) = \Im(f_n(z)) = \frac{(-1)^d \Delta_d}{d!} \Im((z_n - \check{z}_n)^d)$. Setting $z_n - \check{z}_n = r(\cos(\theta) + i \sin(\theta))$, we obtain $u_n(x, y) = \frac{(-1)^d \Delta_d}{d!} r \cos(d\theta)$ and $v_n(x, y) = \frac{(-1)^d \Delta_d}{d!} r \sin(d\theta)$, so $u_n(x, y) = 0$ is equivalent to $\cos(d\theta) = 0$ and $v_n(x, y) = 0$ is equivalent to $\sin(d\theta) = 0$. If we choose $\boldsymbol{x}_n$ and $\boldsymbol{y}_n$ such that

$$\frac{\mathrm{w}(\boldsymbol{y}_n)}{\mathrm{w}(\boldsymbol{x}_n)} = \tan\left(\frac{\pi}{4d}\right), \qquad \text{that is,} \qquad \mathrm{w}(\boldsymbol{y}_n) = \tan\left(\frac{\pi}{4d}\right) \mathrm{w}(\boldsymbol{x}_n), \tag{4.3}$$

then all solutions of $v_n(x, y) = 0$, and consequently all solutions of $\tilde{F}_{\neg u_n}(x, y) = 0$ are arranged in a known pattern on $\boldsymbol{x}_{\underline{n}}$, $\boldsymbol{x}_{\overline{n}}$, $\boldsymbol{y}_{\underline{n}}$, and $\boldsymbol{y}_{\overline{n}}$. In particular, on $\boldsymbol{x}_{\underline{n}}$, $\tilde{x}_n = \underline{x}_n$. $v_n(x, y) = 0$ has a unique solution $\tilde{y}_n = 0$. Plugging these into (4.2), we get the unique solution of $\tilde{F}_{\neg u_n}(x, y) = 0$ with

$$(\tilde{x}, \tilde{y}) = (\check{x}_1 - \alpha_1(\underline{x}_n - \check{x}_n), 0, \ldots, \check{x}_{n-1} - \alpha_{n-1}(\underline{x}_n - \check{x}_n), 0, \underline{x}_n, 0)\,.$$

Similarly, $\tilde{F}_{\neg u_n}(x, y) = 0$ has a unique solution on $\boldsymbol{x}_{\overline{n}}$ with

$$(\tilde{x}, \tilde{y}) = (\check{x}_1 - \alpha_1(\overline{x}_n - \check{x}_n), 0, \ldots, \check{x}_{n-1} - \alpha_{n-1}(\overline{x}_n - \check{x}_n), 0, \overline{x}_n, 0).$$

On $\boldsymbol{y}_{\underline{n}}$, $\tilde{y}_n = \underline{y}_n$. $v_n(x, y) = 0$ has $d - 1$ solutions with

$$\tilde{x}_n = \frac{\mathrm{w}(\boldsymbol{y}_n)}{\tan\left(\frac{m\pi}{d}\right)}, \quad m = d-1, d-2, \ldots, 1. \tag{4.4}$$

Plugging these into (4.2) gives the $d - 1$ solutions $(\tilde{x}, \tilde{y})$ of $\tilde{F}_{\neg u_n}(x, y) = 0$ with

$$(\tilde{x}, \tilde{y}) = \Bigg(\check{x}_1 - \alpha_1\left(\frac{\mathrm{w}(\boldsymbol{y}_n)}{\tan\left(\frac{m\pi}{d}\right)} - \check{x}_n\right), \alpha_1 \underline{y}_n, \ldots, \check{x}_{n-1} - \alpha_{n-1}\left(\frac{\mathrm{w}(\boldsymbol{y}_n)}{\tan\left(\frac{m\pi}{d}\right)} - \check{x}_n\right),$$

$$-\alpha_{n-1}\underline{y}_n, \check{x}_n - \left(\frac{\mathrm{w}(\boldsymbol{y}_n)}{\tan\left(\frac{m\pi}{d}\right)} - \check{x}_n\right), \underline{y}_n\Bigg).$$

Similarly, $\tilde{F}_{\neg u_n}(x, y) = 0$ has $d-1$ solutions on $\boldsymbol{y}_{\overline{n}}$ with

$$(\tilde{x}, \tilde{y}) = \Bigg(\check{x}_1 - \alpha_1\left(\frac{\mathrm{w}(\boldsymbol{y}_n)}{\tan\left(\frac{m\pi}{d}\right)} - \check{x}_n\right), \ldots, \alpha_{n-1}\left(\frac{\mathrm{w}(\boldsymbol{y}_n)}{\tan\left(\frac{m\pi}{d}\right)} - \check{x}_n\right),$$

$$-\alpha_{n-1}\overline{y}_n, \check{x}_n - \left(\frac{\mathrm{w}(\boldsymbol{y}_n)}{\tan\left(\frac{m\pi}{d}\right)} - \check{x}_n\right), \overline{y}_n\Bigg).$$

For example, Figure 4.1 gives the solutions of $v_n(x, y) = 0$ on the four faces $\boldsymbol{x}_{\underline{n}}$, $\boldsymbol{x}_{\overline{n}}$, $\boldsymbol{y}_{\underline{n}}$ and $\boldsymbol{y}_{\overline{n}}$ when $d = 3$.

To use the above analysis to find approximations to the solutions of $\tilde{F}_{\neg u_n} = 0$ on the faces we search, we need to know d; we present a heuristic for d in the next section.

Now, we present our algorithm. The algorithm consists of three phases:

1. the box-construction phase where we set $\tilde{\boldsymbol{x}}$,
2. the elimination phase where we use interval evaluations to verify that $u_k \neq 0$ on $\boldsymbol{x}_{\underline{k}}$ and $\boldsymbol{x}_{\overline{k}}$, and $v_k \neq 0$ on $\boldsymbol{y}_{\underline{k}}$ and $\boldsymbol{y}_{\overline{k}}$, where $1 \leq k \leq n-1$, and thus eliminate those $4n-4$ faces, and
3. the search phase, where we
   (a) search $\boldsymbol{x}_{\underline{n}}$, $\boldsymbol{x}_{\overline{n}}$, $\boldsymbol{y}_{\underline{n}}$ and $\boldsymbol{y}_{\overline{n}}$ to locate the solutions of $\tilde{F}_{\neg u_n}(x, y) = 0$,
   (b) compute the signs of u_n and determinants of the Jacobi matrices of $\tilde{F}_{\neg u_n}$ at those solutions,
   (c) compute the degree contributions of each of the four faces $\boldsymbol{x}_{\underline{n}}$, $\boldsymbol{x}_{\overline{n}}$, $\boldsymbol{y}_{\underline{n}}$ and $\boldsymbol{y}_{\overline{n}}$ according to Theorem 4.1, and
   (d) finally sum up to get the degree.

ALGORITHM 1

Box-setting Phase

1. Compute the preconditioner of the original system, using Gaussian elimination with full pivoting.
2. Set the widths of $\boldsymbol{x}_k$ and $\boldsymbol{y}_k$ (see explanation below), for $1 \leq k \leq n-1$.
3. Set the width of $\boldsymbol{x}_n$ as in (2.7).
4. Set the width of $\boldsymbol{y}_n$ to be the minimum of that obtained from conditions (2.7) and (4.3).

Elimination Phase

Do for $1 \leq k \leq n-1$

1. Do for $\boldsymbol{x}_{\underline{k}}$ and $\boldsymbol{x}_{\overline{k}}$
   (a) Compute the mean-value extension of $\boldsymbol{u}_k$ over that face.
   (b) If $0 \in \boldsymbol{u}_k$, then stop and signal failure.
2. Do for $\boldsymbol{y}_{\underline{k}}$ and $\boldsymbol{y}_{\overline{k}}$
   (a) Compute the mean-value extension of $\boldsymbol{v}_k$ over that face.
   (b) If $0 \in \boldsymbol{v}_k$, then stop and signal failure.

Search Phase

1. Set the value of $s \in \{+1, -1\}$.
   (a) Initialize s to be $+1$. Initialize *search_lower* and *search_upper* to be *false*. (See the second note below.)
   (b) Do for $\boldsymbol{x}_{\underline{n}}$ and $\boldsymbol{x}_{\overline{n}}$

i. Use mean-value extensions for $\boldsymbol{u}_k(\boldsymbol{x},\boldsymbol{y}) = 0$ to solve for x_k to get sharper bounds $\tilde{\boldsymbol{x}}_k$ with width $\mathcal{O}\left(\|(\boldsymbol{x}-\check{x},\boldsymbol{y})\|^2\right)$, $1 \le k \le n-1$, and thus to get a subface $\boldsymbol{x}_{\underline{n}}^0$ (or $\boldsymbol{x}_{\overline{n}}^0$) of $\boldsymbol{x}_{\underline{n}}$ (or $\boldsymbol{x}_{\overline{n}}$.).

ii. If $\tilde{\boldsymbol{x}}_k \cap \boldsymbol{x}_k = \emptyset$, then cycle.

iii. Compute the mean-value extension $\boldsymbol{u}_n$ over $\boldsymbol{x}_{\underline{n}}^0$ (or $\boldsymbol{x}_{\overline{n}}^0$).

iv. If $\boldsymbol{u}_n$ contains 0, then set *search_lower* (or *search_upper*) to be *true* and cycle.

v. If $\boldsymbol{u}_n$ does not contain 0, then set $s = -\mathrm{sgn}(\boldsymbol{u}_n)$.

(c) If $\boldsymbol{u}_n$ does not contain 0 on both $\boldsymbol{x}_{\underline{n}}$ and $\boldsymbol{x}_{\overline{n}}$, then set s to be the opposite sign to the sign of $\boldsymbol{u}_n$ on $\boldsymbol{x}_{\overline{n}}$, and if $\boldsymbol{u}_n$ has different signs on $\boldsymbol{x}_{\underline{n}}$ and $\boldsymbol{x}_{\overline{n}}$, then set *search_lower* to be *true*.

2. For $\boldsymbol{x}_{\underline{n}}$ (or $\boldsymbol{x}_{\overline{n}}$), if *search_lower* (or *search_upper*) is *true*, then take $\boldsymbol{x}_{\underline{n}}$ and 0 as inputs and apply Algorithm 2 to compute the degree contribution of $\boldsymbol{x}_{\underline{n}}$ (or $\boldsymbol{x}_{\overline{n}}$).
3. For $\boldsymbol{y}_{\underline{n}}$ (or $\boldsymbol{y}_{\overline{n}}$)

(a) Use (4.4) to compute the $\tilde{x}_n^m$, $m = d-1, d-2, \ldots, 1$, $\tilde{x}_n^{d-1} < \tilde{x}_n^{d-2} < \ldots < \tilde{x}_n^1$, corresponding to the $d-1$ approximate solutions of $\tilde{F}_{\neg u_n} = 0$ on $\boldsymbol{y}_{\underline{n}}$.

(b) Divide $\boldsymbol{x}_n$ into $d-1$ parts $\boldsymbol{x}_n^m$, $m = 1, \ldots, d-1$ as follows:

$$\boldsymbol{x}_n^1 = [\underline{x}_n, (\tilde{x}_n^1 + \tilde{x}_n^2)/2], \qquad \boldsymbol{x}_n^m = [(\tilde{x}_n^{m-1} + \tilde{x}_n^m)/2, (\tilde{x}_n^m + \tilde{x}_n^{m+1})/2]$$

for $m = 2, \ldots, d-2$, and $\boldsymbol{x}_n^{d-1} = [(\tilde{x}_n^{d-2} + \tilde{x}_n^{d-1})/2, \overline{x}_n]$.

(c) Do for $m = 1, \ldots, d-1$.

i. Set a subface $\boldsymbol{y}_{\underline{n}}^m$ of $\boldsymbol{y}_{\underline{n}}$ (or $\boldsymbol{y}_{\overline{n}}^m$ of $\boldsymbol{y}_{\overline{n}}$) by replacing $\boldsymbol{x}_n$ by $\boldsymbol{x}_n^m$.

ii. Apply Algorithm 3 with $\boldsymbol{y}_{\underline{n}}^m$ and $\tilde{x}_n^m$ as inputs, to compute the degree contribution of $\boldsymbol{y}_{\underline{n}}^m$ (or $\boldsymbol{y}_{\overline{n}}^m$.)

(d) Add the degree contributions in the last step to get the degree contribution of $\boldsymbol{y}_{\underline{n}}$ (or $\boldsymbol{y}_{\overline{n}}$.)

4. Add the degree contributions of $\boldsymbol{x}_{\underline{n}}$, $\boldsymbol{x}_{\overline{n}}$, $\boldsymbol{y}_{\underline{n}}$ and $\boldsymbol{y}_{\overline{n}}$ to get the overall degree.

END OF ALGORITHM 1

Notes for Algorithm 1

1. In Step 3 of the box-setting phase, the width $\mathrm{w}(\boldsymbol{x}_n)$ of $\boldsymbol{x}_n$ depends on the accuracy of the approximate solution $\check{x}$ of the system $F(x) = 0$: $\mathrm{w}(\boldsymbol{x}_n)$ should be much larger than $|\check{x}_k - x_k^*|$, but also should be sufficiently small for a quadratic model to be accurate over the box.
2. We may set s to minimize the amount of work required to evaluate the sum in Theorem 4.1. In particular, if we know $\mathrm{sgn} u_n = \sigma$ on a large number of faces, then setting $s = -\sigma$ will eliminate the need to search those faces.

ALGORITHM 2

Inputs: $\boldsymbol{x}_{\underline{n}}$ and $\check{y}_n$ (or $\boldsymbol{x}_{\overline{n}}$ and $\check{y}_n$)

1. (a) Use mean-value extensions for $\boldsymbol{u}_k(\boldsymbol{x},\boldsymbol{y}) = 0$ to solve for x_k to get sharper bounds $\tilde{\boldsymbol{x}}_k$ with width $\mathcal{O}\left(\|(\boldsymbol{x}-\check{x},\boldsymbol{y})\|^2\right)$, $1 \le k \le n-1$.

   (b) If $\tilde{\boldsymbol{x}}_k \cap \boldsymbol{x}_k = \emptyset$, then return the degree contribution of that face as 0.

   (c) Update $\boldsymbol{x}_k$.
2. (a) Compute the mean-value extension $\boldsymbol{u}_n$ over that face.

   (b) If $s \times \mathrm{sgn}(\boldsymbol{u}_n) < 0$, then return the degree contribution of that face as 0.
3. Construct a small subinterval $\boldsymbol{y}_n^0$ of $\boldsymbol{y}_n$ centered at $\check{y}_n$.

4. Step 4 to step 9 are identical to step 1(d) to step 1(i), respectively, of the search phase in the algorithm in [10].
10. Apply Theorem 4.1 to compute the degree contribution of $\boldsymbol{x}_{\underline{n}}$ or $\boldsymbol{x}_{\overline{n}}$.

END OF ALGORITHM 2

ALGORITHM 3

Inputs: $\boldsymbol{y}_{\underline{n}}$ and $\check{x}_n$ (or $\boldsymbol{y}_{\overline{n}}$ and $\check{x}_n$.)

1. (a) Use mean-value extensions for $\boldsymbol{v}_k(\boldsymbol{x}, \boldsymbol{y}) = 0$ to solve for y_k to get sharper bounds $\tilde{\boldsymbol{y}}_k$ with width $\mathcal{O}\left(\|(\boldsymbol{x} - \check{x}, \boldsymbol{y})\|^2\right)$, $1 \le k \le n-1$.
   (b) If $\tilde{\boldsymbol{y}}_k \cap \boldsymbol{y}_k = \emptyset$, then return the degree contribution of that face as 0.
   (c) Update $\boldsymbol{y}_k$.
2. (a) Compute the mean-value extension $\boldsymbol{u}_n$ over that face.
   (b) If $s \times \operatorname{sgn}(\boldsymbol{u}_n) < 0$, then return the degree contribution of that face as 0.
3. Construct a small subinterval $\boldsymbol{x}_n^0$ of $\boldsymbol{x}_n$ which is centered at $\check{x}_n$.
4. Step 4 to step 9 are identical to step 2(d) to step 2(i), respectively, of the search phase in the algorithm in [10].
10. Same as step 10 of Algorithm 2.

END OF ALGORITHM 3

Notes for Algorithm 2 and Algorithm 3

1. Algorithms 2 and 3 are identical to steps 1 and 2 of the search phase of the algorithm in [10], except, in Algorithm 2, $\check{y}_n$ can be any interior point of $\boldsymbol{y}_n$, while $\check{y}_n$ is assumed to equal zero in step 1 of the search phase in the algorithm in [10]. Similarly, in Algorithm 3, $\check{x}_n$ can be any interior point of $\boldsymbol{x}_n$, whereas $\check{x}_n$ is assumed to equal the center of $\boldsymbol{x}_n$ in step 2 of the search phase in the algorithm in [10].
2. In the overall algorithm, Algorithm 1, the actual inputs are $\boldsymbol{y}_{\underline{n}}^m$ and $\check{x}_n^m$ when Algorithm 3 is applied. However, for notational simplicity, we use $\boldsymbol{y}_{\underline{n}}$ and $\check{x}_n$ as inputs in the presentation of Algorithm 3.

The computational complexity of Algorithms 1, 2, and 3 is $\mathcal{O}\left(n^3\right)$. (See [10] for detailed analysis.) Thus, the computational complexity of the overall algorithm, Algorithm 1, is $\mathcal{O}\left(n^3\right)$. This is the best possible order, since computing preconditioners of the original system and the system $\tilde{F}_{\neg u_n}$ is necessary and computing each preconditioner is of order $\mathcal{O}\left(n^3\right)$.

5. A Heuristic for the Degree. The algorithms in §4 require a value for d to locate the approximate positions of solutions of $\tilde{F}_{\neg u_n} = 0$ on the faces we search. Here, we present a practical heuristic for the value of d.

Suppose (2.3) and (2.4) are exact. Then, if we set $x_k - \check{x}_k = -\alpha_k(x_n - \check{x}_n)$, $k = 1, \ldots, n$ and plug those equalities into f_n, we obtain a univariate function

$$
\begin{aligned}
g(x_n - \check{x}_n) &= f_n(x_1, \ldots, x_n) \\
&= \frac{1}{2!} \sum_{k_1=1}^{n} \sum_{k_2=1}^{n} \frac{\partial^2 f_n}{\partial x_{k_1} \partial x_{k_2}}(\check{x})(x_{k_1} - \check{x}_{k_1})(x_{k_2} - \check{x}_{k_2}) + \ldots + \\
&\quad \frac{1}{d!} \sum_{k_1=1}^{n} \cdots \sum_{k_d=1}^{n} \frac{\partial^2 f_n}{\partial x_{k_1} \ldots \partial x_{k_d}}(\check{x})(x_{k_1} - \check{x}_{k_1}) \ldots (x_{k_d} - \check{x}_{k_d}) \\
&= \frac{1}{2!} \sum_{k_1=1}^{n} \sum_{k_2=1}^{n} \frac{\partial^2 f_n}{\partial x_{k_1} \partial x_{k_2}}(\check{x})(-1)^2 \alpha_{k_1} \alpha_{k_2} (x_n - \check{x}_n)^2 + \ldots +
\end{aligned}
$$

$$
\begin{aligned}
&\frac{1}{d!}\sum_{k_1=1}^{n}\cdots\sum_{k_d=1}^{n}\frac{\partial^2 f_n}{\partial x_{k_1}\ldots\partial x_{k_d}}(\check{x})(-1)^d\alpha_{k_1}\ldots\alpha_{k_d}(x_n-\check{x}_n)^d\\
&=\frac{(-1)^2}{2!}(x_n-\check{x}_n)^2\frac{\partial^2 f_n}{\partial x_{k_1}\partial x_{k_2}}(\check{x})\alpha_{k_1}\alpha_{k_2}+\ldots\\
&\quad+\frac{(-1)^d}{d!}(x_n-\check{x}_n)^d\frac{\partial^2 f_n}{\partial x_{k_1}\ldots\partial x_{k_d}}(\check{x})\alpha_{k_1}\ldots\alpha_{k_d}\\
&=\frac{(-1)^2\Delta_2}{2!}(x_n-\check{x}_n)^2+\ldots+\frac{(-1)^{d-1}\Delta_{d-1}}{(d-1)!}(x_n-\check{x}_n)^{d-1}\\
&\quad+\frac{(-1)^d\Delta_d}{d!}(x_n-\check{x}_n)^d\\
&=\frac{(-1)^d\Delta_d}{d!}(x_n-\check{x}_n)^d=\frac{\Delta_d}{d!}(\check{x}_n-x_n)^d.
\end{aligned}
$$

Setting

$$
K(r,x_n-\check{x}_n)\equiv\frac{g(x_n-\check{x}_n)}{(x_n-\check{x}_n)^r}=\frac{\Delta_d}{d!}(\check{x}_n-x_n)^{d-r},
$$

it is clear that $K(d,x_n-\check{x}_n)=\Delta_d/d!$ is independent of x_n, while $K(r,x_n-\check{x}_n)$ depends on x_n for any other r value. We have the following ratios.

$$
\frac{K(d,\delta(x_n-\check{x}_n))}{K(d,x_n-\check{x}_n)}=\frac{\frac{\Delta_d}{d!}}{\frac{\Delta_d}{d!}}=1,\quad\text{while}
$$

$$
R(r)=\frac{K(r,\delta(x_n-\check{x}_n))}{K(r,x_n-\check{x}_n)}=\frac{\frac{\Delta_d}{d!}(\delta(\check{x}_n-x_n))^{d-r}}{\frac{\Delta_d}{d!}(\check{x}_n-x_n)^{d-r}}=\delta^{d-r}
$$

for any other r value. The first ratio $R(d)$ always equals 1, but $R(r)$, $r\neq d$, depends on the δ value. We can choose δ to distinguish d from other r values. For example, if we choose $\delta=100$, then $R(r)$ is not smaller than 100 when r is smaller than d and is not larger than 0.01 when r is larger than d. Both values are sufficiently different from 1. We can also vary the δ value to check our detection of d. Thus, $R(r)$ is a good heuristic to determine the value of d.

The above discussion is based on the assumptions in §2. However, unless the first $n-1$ components of F are exactly linear and the last component is a degree-d polynomial of n variables, those assumptions are only approximately true. There are some finer issues to consider. The above analysis is valid only when the equality $g(x_n-\check{x}_n)=\frac{\Delta_d}{d!}(\check{x}_n-x_n)^d$ is accurately approximated. That implies $(\check{x}_n-x_n)^d$ should dominate the value of $g(x_n-\check{x}_n)$. Actually,

$$
g(x_n-\check{x}_n)=\sum_{k=1}^{d-1}c_k\Delta_k(x_n-\check{x}_n)^k+c_d\Delta_d(x_n-\check{x}_n)^d+\sum_{k=d+1}^{\infty}c_k\Delta_k(x_n-\check{x}_n)^k,
$$

where, approximately, $\Delta_1=\ldots=\Delta_{d-1}=0$, $\Delta_d\neq 0$. Thus, $x_n-\check{x}_n$ and $\delta(x_n-\check{x}_n)$ should not be too small, since $\sum_{k=1}^{d-1}c_k\Delta_k(x_n-\check{x}_n)^k$ could dominate otherwise. They should not be too big either, since $\sum_{k=d+1}^{\infty}c_k\Delta_k(x_n-\check{x}_n)^k$ could dominate otherwise. If $\Delta_k\approx 0, k=1,\ldots,d-1$ are quite accurate, then we can choose $x_n-\check{x}_n$ very small, so both $\sum_{k=1}^{d-1}c_k\Delta_k(x_n-\check{x}_n)^k$ and $\sum_{k=d+1}^{\infty}c_k\Delta_k(x_n-\check{x}_n)^k$ can be ignored in the detection of d.

The choice of $x_n - \check{x}_n$ is independent of the settings of $\boldsymbol{x}_k$, $k = 1, \ldots, n$, since we only want to know what d is at that stage.

An alternative choice for detecting d is to compute the values of Δ_k, $k = 1, 2, \ldots$ by interval evaluations until we get some Δ_{k_0} that is sufficiently different from 0. Then, we can decide $d = k_0$. The obvious disadvantage of this method is that it's too expensive for just detecting the value of d, since computation of Δ_k involves computations of all k-th order derivatives. Furthermore, even if we actually evaluate Δ_k, $k = 1, 2, \ldots$, spending much time in the process, we still can not detect the value of d if the magnitudes of Δ_k, $k = 1, \ldots, d-1, d$, are not sufficiently different either due to the problem itself or due to the range overestimation in interval computations.

6. Numerical Results. In this section, we present numerical results for the algorithm in §4.

6.1. Test Problems. EXAMPLE 1. *(The same as Example 3 from [10], motivated from considerations in [5] Set $f(x) = h(x,t) = (1-t)(Ax - x^2) - tx$, where $A \in \mathbb{R}^{n\times n}$ is the matrix corresponding to central difference discretization of the boundary value problem $-u'' = 0$, $u(0) = u(1) = 0$ and $x^2 = (x_1^2, \ldots, x_n^2)^T$. t was chosen to be equal to $t_1 = \lambda_1/(1+\lambda_1)$, where λ_1 is the largest eigenvalue of A.*

In Example 1, if we change the exponent of x from 2 to 3, then we get another problem.

EXAMPLE 2. *This example is identical to Example 1, except that we set $f(x) = h(x,t) = (1-t)(Ax - x^3) - tx$.*

The test set consists of Example 1 and Example 2 with n = 5, 10, 20, 40, 80, 160. For each test problem, we used $(0, 0, \ldots, 0)$, the exact solution to $F(x) = 0$, as the approximate solution to the problem $F(x) = 0$. We set the widths $\mathrm{w}(\boldsymbol{x}_k)$ and $\mathrm{w}(\boldsymbol{y}_k)$ to 10^{-2} for $1 \le k \le n-1$, then the algorithm automatically computed $\mathrm{w}(\boldsymbol{x}_n)$ and $\mathrm{w}(\boldsymbol{y}_n)$. For all the problems, the algorithm succeeded.

6.2. Test Environment. The algorithm in §4 was programmed in the Fortran 90 environment developed and described in [8, 9]. Similarly, all the test functions were programmed using the same Fortran 90 system, which generated internal symbolic representations of the functions to execution of the numerical tests. In the actual tests, generic routines then interpreted the internal representations to obtain both floating point and interval values.

The Sun Fortran 95 compiler version 6.0 was used on a Sparc Ultra-1 model 140 with optimization level 0. Execution times were measured with the Port library routine `ETIME`. All times are given in CPU seconds.

6.3. Test Results. We present the numerical results in Table 6.1.

The column labels of Table 6.1 are as follows.

Problem: names of the problems identified in §6.1.
n: number of independent variables.
Heuristic Degree: the heuristic value of the degree computed by the heuristic described in §5.
Success: whether the algorithm was successful.
Verified Degree: topological degree verified by the algorithm.
CPU Time: CPU time in seconds of the algorithm.
Time Ratio: The ratio of two successive CPU times for a particular problem.

We can see from Table 6.1 that the algorithm was successful on all the problems in the test set. We also see from the CPU time ratios that the algorithm is approximately of order $\mathcal{O}(n^3)$ in practice. However, when the degree is higher, the system $F(x)$ is

Table 6.1
Numerical Results

Problem	n	Heuristic Degree	Success	Verified Degree	CPU Time	Time Ratio
Example 1	5	2	Yes	2	1.13	
Example 1	10	2	Yes	2	5.99	5.30
Example 1	20	2	Yes	2	38.40	6.41
Example 1	40	2	Yes	2	273.61	7.13
Example 1	80	2	Yes	2	2198.14	8.03
Example 1	160	2	Yes	2	13033.22	5.93
Example 2	5	3	Yes	3	39.27	
Example 2	10	3	Yes	3	10.31	0.26
Example 2	20	3	Yes	3	74.32	7.21
Example 2	40	3	Yes	3	481.23	6.48
Example 2	80	3	Yes	3	3805.06	7.91
Example 2	160	3	Yes	3	33944.20	8.92

flatter at the singular solution. We can expect that the condition number of the Jacobi matrix of the system $\tilde{F}_{\neg u_n}$ will be larger, and thus, make the interval Newton method method to verify the unique solutions of $\tilde{F}_{\neg u_n}$ in Step 5 of Algorithm 2 and Step 5 of Algorithm 3 less efficient: More iterations should be expected when the condition number is larger. The experimental results are consistent with our expectations.

REFERENCES

[1] G. Alefeld and J. Herzberger. *Introduction to Interval Computations.* Academic Press, New York, 1983.

[2] P. S. Aleksandrov and H. Hopf. *Topologie.* Springer, Berlin, 1935.

[3] J. Cronin. *Fixed Points and Topological Degree in Nonlinear Analysis.* American Mathematical Society, Providence, RI, 1964.

[4] E. R. Hansen. *Global Optimization Using Interval Analysis.* Marcel Dekker, Inc., New York, 1992.

[5] H. Jürgens, H.-O. Peitgen, and D. Saupe. Topological perturbations in the numerical nonlinear eigenvalue and bifurcation problems. In S. M. Robinson, editor, *Analysis and Computation of Fixed Points*, pages 139–181, New York, 1980. Academic Press.

[6] R. B. Kearfott. *Computing the Degree of Maps and a Generalized Method of Bisection.* PhD thesis, University of Utah, 1977.

[7] R. B. Kearfott. An efficient degree-computation method for a generalized method of bisection. *Numer. Math.*, 32:109–127, 1979.

[8] R. B. Kearfott. A Fortran 90 environment for research and prototyping of enclosure algorithms for nonlinear equations and global optimization. *ACM Trans. Math. Software*, 21:63–78, 1995.

[9] R. B. Kearfott. *Rigorous Global Search: Continuous Problems.* Kluwer, Dordrecht, 1996.

[10] R. B. Kearfott, J. Dian, and A. Neumaier. Existence verification for singular zeros of complex nonlinear systems. *SIAM. J. Numer. Anal.*, 38:360–379, 2000.

[11] A. Neumaier. *Interval Methods for Systems of Equations.* Cambridge University Press, Cambridge, England, 1990.

[12] J. M. Ortega and W. C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables.* Academic Press, New York, 1970.

[13] H. Ratschek and J. Rokne. *New Computer Methods for Global Optimization.* Wiley, New York, 1988.

[14] F. Stenger. Computing the topological degree of a mapping in $\mathbb{R}^n$. *Numer. Math.*, 25:23–38, 1975.