EXISTENCE VERIFICATION FOR SINGULAR ZEROS OF NONLINEAR SYSTEMS*

R. BAKER KEARFOTT†, JIANWEI DIAN‡, AND A. NEUMAIER§

Abstract. Computational fixed point theorems can be used to automatically verify existence and uniqueness of a solution to a nonlinear system of equations $F(x) = 0$, $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ within a given region $\boldsymbol{x}$ of n-space. But such computations succeed only when the Jacobi matrix $F'(x)$ is nonsingular everywhere in $\boldsymbol{x}$. However, in many practical problems, the Jacobi matrix is singular, or nearly so, at the solution x^*, $F(x^*) = 0$. In such cases, arbitrarily small perturbations of the problem result in problems $\tilde{F}(x) = 0$ either with *no* solutions in $\boldsymbol{x}$ or with *more than one* solution in $\boldsymbol{x}$; thus no general computational technique can prove existence and uniqueness. This leads to a fundamental philosophical problem: "What is meant by existence and uniqueness in such cases?"

Here, an interpretation of verification is given in the singular context: proof that a given number of true solutions exist within a region in complex space containing $\boldsymbol{x}$.

Proof that a given number of true solutions exist within a given region of complex space is possible by computation of the topological degree, but such computations heretofore have required a global search on the $(n-1)$-dimensional boundary of an n-dimensional region. Here, it is observed that preconditioning leads to a system of equations whose topological degree can be computed with a much lower-dimensional search. Formulas are given for this computation, and the special case of rank-defect one is studied, both theoretically and empirically.

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

AMS subject classifications. 65G10, 65H10

1. Introduction.

Given an approximate solution $\check{x}$ to a nonlinear system of equations $F(x) = 0$, $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$, it is useful in various contexts to construct bounds around $\check{x}$ in which it is proven that there exists a unique solution x^*, $F(x^*) = 0$. For continuously differentiable F for which the Jacobian $\det(F'(x^*)) \neq 0$ and for which that Jacobian is well conditioned, interval computations have no trouble proving that there is a unique solution within small boxes with x^* reasonably near the center; see [5, 9, 10]. However, if $F'(x^*)$ is ill-conditioned or singular, such computations necessarily must fail. In this singular situation, what is meant by "unique solution" can be defined in several ways. A particular interpretation may be most appropriate in a particular practical context. Also, given a particular interpretation, various computational procedures can be developed to verify existence and uniqueness. In this paper, one interpretation and its corresponding computational procedure are considered.

1.1. Notation.

We assume familiarity with the fundamentals of interval arithmetic; see [9, 10] for an introduction in the present context. (The works [2, 5, 11] also contain introductory material.)

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

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

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

‡Department of Mathematics, University of Louisiana at Lafayette, Lafayette, LA 70504, USA (dian@usl.edu).

§Institut für Mathematik, Universität Wien, Strudhofgasse 4, A-1050 Wien, Austria (neum@cma.univie.ac.at).

an interval vector, $A = (a_{i,j})$ denotes a point matrix, and $\boldsymbol{A} = (\boldsymbol{a}_{i,j})$ denotes an interval matrix. Real n-space will be denoted by $\mathbb{R}^n$, while the set of n-dimensional interval vectors, i.e. n-dimensional boxes, will be denoted by $\mathbb{IR}^n$. Similarly, complex n-space will be denoted by $\mathbb{C}^n$, while the space of n-dimensional complex interval vectors will be denoted by $\mathbb{IC}^n$. The midpoint of an interval or interval vector $\boldsymbol{x}$ will be denoted by $\mathrm{m}(\boldsymbol{x})$. The non-oriented boundary of a box $\boldsymbol{x}$ will be denoted by $\boldsymbol{\partial x}$ while its oriented boundary will be denoted by $b(\boldsymbol{x})$. (See §2.)

1.2. Traditional Computational Existence and Uniqueness. Computational existence and uniqueness verification rests on interval versions of Newton's method. Typically, such computations can be described as evaluation of a related interval operator $\boldsymbol{G}(\boldsymbol{x})$; $\boldsymbol{G}(\boldsymbol{x}) \subseteq \boldsymbol{x}$ then implies existence and uniqueness of $F(x) = 0$ within $\boldsymbol{x}$. To describe these, we review

DEFINITION 1.1. ([10, p. 174], etc.) *Let $F : \mathbb{R}^n \to \mathbb{R}^m$. The matrix $\boldsymbol{A}$ is said to be a* Lipschitz matrix *for F over $\boldsymbol{x}$ provided, for every $x \in \boldsymbol{x}$ and $y \in \boldsymbol{x}$, $F(x) - F(y) = A(x-y)$ for some $A \in \boldsymbol{A}$.*

Most interval Newton methods for $F : \mathbb{R}^n \to \mathbb{R}^n$, abstractly, are of the general form

$$\tilde{\boldsymbol{x}} = \boldsymbol{N}(F; \boldsymbol{x}, \check{x}) = \check{x} + \boldsymbol{v}, \tag{1.1}$$

where $\boldsymbol{v}$ is computed to contain the solution set to the interval linear system

$$\boldsymbol{A}v = -F(\check{x}), \tag{1.2}$$

and where, for initial uniqueness verification, $\boldsymbol{A}$ is generally a Lipschitz matrix[1] for F over the box (interval vector) $\boldsymbol{x}$ and $\check{x} \in \boldsymbol{x}$ is a guess point. We sometimes write $\boldsymbol{F'}(\boldsymbol{x})$ in place of $\boldsymbol{A}$, since the matrix can be an interval extension of the Jacobi matrix of F. Uniqueness verification traditionally depends on regularity of the matrix $\boldsymbol{A}$. We have

LEMMA 1.2. *([9, 10]) Suppose $\tilde{\boldsymbol{x}} = \check{x} + \boldsymbol{v}$ is the image under the interval Newton method (formula (1.1)), where $\boldsymbol{v}$ is computed by any method that bounds the solution set to the interval linear system (1.2), and $\tilde{\boldsymbol{x}} \subseteq \boldsymbol{x}$. Then $\boldsymbol{A}$ is regular.*

The method of bounding the solution set of equation (1.2) to be considered here is the interval Gauss–Seidel method, defined by:

DEFINITION 1.3. *The preconditioned interval Gauss–Seidel image $\mathbf{GS}(F; \boldsymbol{x}, \check{x})$ of a box $\boldsymbol{x}$ is defined as $\mathbf{GS}(F; \boldsymbol{x}, \check{x}) \equiv (\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_n)$, where $\tilde{\boldsymbol{x}}_i$ is defined sequentially for $i = 1$ to n by:*

$$\tilde{\boldsymbol{x}}_i \equiv \boldsymbol{x}_i \cap \left(\check{x}_i - \boldsymbol{N}_i / (Y_i \boldsymbol{A}_i)\right),$$

where

$$\boldsymbol{N}_i = Y_i F(\check{x}) + \sum_{j=1}^{i-1} Y_i \boldsymbol{A}_j (\tilde{\boldsymbol{x}}_j - \check{x}_j) + \sum_{j=i+1}^{n} Y_i \boldsymbol{A}_j (\boldsymbol{x}_j - \check{x}_j),$$

and where $\check{x} = (\check{x}_1, \ldots, \check{x}_n)^T$ is an initial guess point, $Y\boldsymbol{A} \in \mathbb{IR}^{n\times n}$ and $YF(\check{x})$ are the matrix and right-hand-side vector for the preconditioned interval system $Y\boldsymbol{A}(x - \check{x}) = -YF(\check{x})$, $Y \in \mathbb{R}^{n\times n}$ is a point preconditioning matrix, Y_i denotes the i-th row of Y, and $\boldsymbol{A}_j$ denotes the j-th column of $\boldsymbol{A}$.

[1]However, see [9, 12] for techniques for using slope matrices.

Lemma 1.2 applies when $\boldsymbol{N}(F;\boldsymbol{x},\check{x}) = \mathbf{GS}(F;\boldsymbol{x},\check{x})$, provided we specify that $\mathbf{GS}(F;\boldsymbol{x},\check{x})$ be in the interior $\mathrm{int}(\boldsymbol{x})$ of $\boldsymbol{x}$. (We must specify the interior because of the intersection step in Definition 1.3.) In particular, we have

THEOREM 1.4. *([9, 10]) Suppose $F : \boldsymbol{x} \subset \mathbb{R}^n \to \mathbb{R}^n$ and $\boldsymbol{A}$ is a Lipschitz matrix such as an interval extension $\boldsymbol{F'}(\boldsymbol{x})$ of the Jacobi matrix. If $\tilde{\boldsymbol{x}}$ is the image under an interval Newton method as in formula (1.1) and $\tilde{\boldsymbol{x}} \subset \mathrm{int}(\boldsymbol{x})$, then there is a unique $x^* \in \boldsymbol{x}$ with $F(x^*) = 0$.* Various authors have proven Theorem 1.4; see [9, 10]. In particular, Miranda's theorem can be used to easily prove Theorem 1.4 for $\boldsymbol{N}(F;\boldsymbol{x},\check{x}) = \mathbf{GS}(F;\boldsymbol{x},\check{x})$; see [9, p. 60].

EXAMPLE 1. *Take*

$$f_1(x_1,x_2) = x_1^2 - x_2,$$
$$f_2(x_1,x_2) = x_1 - x_2^2,$$

and $\boldsymbol{x} = \left(\boldsymbol{x}_1,\boldsymbol{x}_2\right)^T = \left([-0.1,0.1],[-0.1,0.3]\right)^T$. *There is a unique root $x^* = (0,0)^T$ of $F = (f_1,f_2)^T$ within $\boldsymbol{x}$.* In Example 1, if $\check{x} = (0,.1)^T$, then $F(\check{x}) = (-0.1,-0.01)^T$, and an interval extension of the Jacobi matrix is

$$\boldsymbol{F'}(\boldsymbol{x}) = \begin{pmatrix} 2\boldsymbol{x}_1 & -1 \\ 1 & -2\boldsymbol{x}_2 \end{pmatrix} = \begin{pmatrix} [-.2,.2] & -1 \\ 1 & [-0.6,0.2] \end{pmatrix}.$$

If the preconditioning matrix Y is taken to be the inverse of the midpoint matrix of $\boldsymbol{F'}(\boldsymbol{x})$, then

$$Y = \left\{\mathrm{m}\left(\boldsymbol{F'}(\boldsymbol{X})\right)\right\}^{-1} = \begin{pmatrix} -0.2 & 1 \\ -1 & 0 \end{pmatrix}.$$

We obtain, rounded out to four digits,

$$\mathbf{GS}(F;\boldsymbol{x},\check{x}) = ([-0.07292,0.09375],[-0.01875,0.01875])^T \subset \mathrm{int}(\boldsymbol{x}).$$

Therefore, this computation proves that there is a unique solution to $F(x) = 0$ within $\boldsymbol{x}$.

Inclusion in the interval Gauss–Seidel method is made possible because the inverse midpoint preconditioner reduces the interval Jacobi matrix to approximately a diagonal matrix.

1.3. Singularities: Philosophical Considerations. Theorem 1.4 is applicable only when the matrix $\boldsymbol{A}$ is regular, i.e. when $\boldsymbol{A}$ does not contain singular matrices. But consider

EXAMPLE 2. *Take*

$$f_1(x_1,x_2) = x_1^2 - x_2,$$
$$f_2(x_1,x_2) = x_1^2 + x_2,$$

and $\boldsymbol{x} = \left(\boldsymbol{x}_1,\boldsymbol{x}_2\right)^T = \left([-0.001,0.001],[-0.001,0.001]\right)^T$. Even though there is a unique root $x^* = (0,0)^T$ of $F = (f_1,f_2)^T$ within $\boldsymbol{x}$ when F is as in Example 2, the interval Gauss–Seidel method cannot prove this, since the Jacobi matrix $F'(x^*)$ is singular. In fact, the interval Jacobi matrix is computed to be

$$\boldsymbol{F'}(\boldsymbol{x}) = \begin{pmatrix} 2\boldsymbol{x}_1 & -1 \\ 2\boldsymbol{x}_1 & 1 \end{pmatrix} = \begin{pmatrix} \text{[-0.002,0.002]} & -1 \\ \text{[-0.002,0.002]} & 1 \end{pmatrix},$$

and the midpoint

$$\mathrm{m}(\boldsymbol{F}'(\boldsymbol{x})) = \begin{pmatrix} 0 & -1 \\ 0 & 1 \end{pmatrix},$$

matrix the inverse of which is often used as the preconditioner matrix Y, is not invertible[2].

Symbolic methods can be used to show that Example 2 has a unique solution in a small region containing $x_1 = 0$, $x_2 = 0$. However, arbitrarily small perturbations of the problem result in either no solutions or two solutions. Consider

EXAMPLE 3. *Take*

$$\begin{aligned} f_1(x_1, x_2) &= x_1^2 - x_2, \\ f_2(x_1, x_2) &= x_1^2 + x_2 + \epsilon, \end{aligned}$$

and $\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2)^T = ([-0.001, 0.001], [-0.001, 0.001])^T$. *Here,* $|\epsilon|$ *is very small.*

The system in Example 3 has two solutions for $\epsilon < 0$ and no solutions for $\epsilon > 0$. But roundout in computer arithmetic and, perhaps, uncertainties in the system itself due to modelling or measurement uncertainties, make it impossible to distinguish systems such as in Example 3 for different ϵ, especially when computer arithmetic is used as part of the verification process. In such instances, a new interpretation needs to be given to existence / uniqueness. One expects the appropriate interpretation to be application-dependent in general, but the following is possible: **verify that the system has an exact number of solutions within a larger space containing the original space**. This paper will be based on this interpretation.

To be specific, we can extend an n-dimensional box in $\mathbb{R}^n$ to an n-dimensional box in $\mathbb{C}^n$ by adding a small imaginary part to each variable. If the system can be extended to an analytic function in complex n-space (or if it can be extended to a function that can be approximated by an analytic function), then the *topological degree* gives the number of solutions, counting multiplicities, within the small region in complex space. (See §2 for an explanation of multiplicity.) When the imaginary parts are small, we can view the complex roots as approximating a multiple real root. For example, the degree of the system in Example 3 within an extended box in complex space can be computed to be 2, regardless of whether ϵ is negative, positive or zero. (See the numerical results in §8.) The topological degree corresponds roughly to algebraic degree in one dimension; for example, the degree of z^n in a small region in $\mathbb{C}^1$ containing 0 is n.

1.4. Organization of This Paper. A review of properties of the topological degree, to be used later, appears in §2. The issue of preconditioning appears in §3. Construction of the box in the complex space appears in §4.

Several algorithms have previously been proposed for computing the topological degree [1, 7, 14], but these require computational effort equivalent to finding all solutions to $4n$ (2n-1)-dimensional nonlinear systems within a given box, or worse. In §5, a reduction is proposed that allows computation of the topological degree with a search in a space of dimension equal to the rank defect of the Jacobian matrix. A theorem is proven that further simplifies the search.

[2]Alternate preconditioners can nonetheless be computed; see [9]. However, it can be shown that uniqueness cannot be proven in this case; see [9, 10].

In §6, the actual algorithm is presented and its computational complexity is given. Test problems and the test environment are described in §7. Numerical results appear in §8. Future directions appear in §9.

2. Review of Some Elements of Degree Theory. The topological degree or Brouwer degree, well-known within algebraic topology and nonlinear functional analysis, is both a generalization of the concept of a sign change of a one-dimensional continuous function and of the winding number for analytic functions. It can be used to generalize the concept of multiplicity of a root. The fundamentals will not be reviewed here, but we refer to [3, 4, 7]. We only present the material we need.

Here, we explain what we mean by "multiplicity". Actually, there is a more general concept *index* (see [4, Chapter I]) for an isolated zero. The topological degree is equal to the sum of the indices of zeros in the domain. The index is always positive in our context. So, we use multiplicity as an alternative term for index since it's more suggestive.

Suppose that $F : \mathbf{D} \subset \mathbb{C}^n \to \mathbb{C}^n$ is analytic. Then the real and imaginary components of F and its argument $z \in \mathbb{C}^n$ may be viewed 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 can 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 $\tilde{F} : \tilde{\mathbf{D}} \subset \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ 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 2.1. *Suppose $F : \mathbf{D} \subset \mathbb{C}^n \to \mathbb{C}^n$ is analytic, with $F(z) \neq 0$ for any $z \in \boldsymbol{\partial}\mathbf{D}$, and suppose $\tilde{\mathbf{D}}$ and $\tilde{F} : \tilde{\mathbf{D}} \to \mathbb{R}^{2n}$ 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 $z^* \in \mathbf{D}$, $F(z^*) = 0$.*

To obtain Theorem 2.1, we need to notice that $F(z^*) = 0$ for $z^* \in \mathbf{D}$ is equivalent to $\tilde{F}(x^*, y^*) = 0$ for $(x^*, y^*) \in \tilde{\mathbf{D}}$, where $z^* = x^* + iy^*$. In Theorem 2.1, the first conclusion is easily obtained by knowing $|\mathrm{d}(F_R, \mathbf{D}_R, 0)| \leq \mathrm{d}(\tilde{F}, \tilde{\mathbf{D}}, 0)$, where F_R is function $F(z)$ with z in a corresponding n-dimensional real domain $\mathbf{D}_R \in \mathbb{R}^n$. (See [4, p. 49].) The third conclusion is actually Theorem (5.2) in Chapter I of [4], considering the definition of topological degree. The fourth conclusion can be obtained by the third conclusion, Theorem (7.2) and Lemma (9.3-2) in Chapter I of [4]. As for the second conclusion, it's easy to see that $\mathrm{d}(\tilde{F}, \tilde{\mathbf{D}}, 0) > 0$ implies there is a solution $(x^*, y^*) \in \tilde{\mathbf{D}}$, $\tilde{F}(x^*, y^*) = 0$. (See Existence Theorem (6.6) in Chapter I of [4].) The reverse is obtained by noticing the third conclusion and the fact that the multiplicities are positive.

The following three theorems will lead to the degree computation formula in Theorem 5.1 in §5, the formula used in our computational scheme.

THEOREM 2.2. *(See [13, §4.2].) Let $\mathbf{D}$ be an n-dimensional connected, oriented region in $\mathbb{R}^n$ and $F = (f_1, \ldots, f_n)$, where f_k, $k = 1, \ldots, n$, are continuous functions* *defined in* $\mathbf{D}$*. Assume* $F \neq 0$ *on the oriented boundary* $b(\mathbf{D})$ *of* $\mathbf{D}$*,* $b(\mathbf{D})$ *can be subdivided into a finite number of closed, connected* $(n-1)$*-dimensional oriented subsets* $\beta_{n-1}^k, k = 1, \ldots, r$*, and there is* p*,* $1 \le p \le n$*, such that:*

1. $F_{\neg p} \equiv (f_1, \ldots, f_{p-1}, f_{p+1}, \ldots, f_n) \neq 0$ *on the oriented boundary* $b(\beta_{n-1}^k)$ *of* β_{n-1}^k*,* $k = 1, \ldots, r$*; and*
2. f_p *has the same sign at all solutions of* $F_{\neg p} = 0$*, if there are any, on* β_{n-1}^k*,* $1 \le k \le r$*.*

Choose $s \in \{-1, +1\}$ *and let* $K_0(s)$ *denote the subset of the integers* $k \in \{1, \ldots, r\}$ *such that* $F_{\neg p} = 0$ *has solutions on* β_{n-1}^k *and* $\mathrm{sgn}(f_p) = s$ *at each of those solutions. Then*

$$\mathrm{d}(F, \mathbf{D}, 0) = (-1)^{p-1} s \sum_{k \in K_0(s)} \mathrm{d}(F_{\neg p}, \beta_{n-1}^k, 0).$$

The formula in Theorem 2.2 is a combination of formulas (4.15) and (4.16) in [13]. The orientation of $\mathbf{D}$ is positive and the orientations of β_{n-1}^k, whether positive or negative, are induced by the orientation of $\mathbf{D}$. If we assume that the Jacobi matrices of $F_{\neg p}$ are non-singular at all solutions of $F_{\neg p} = 0$ on β_{n-1}^k, then

$$\mathrm{d}(F_{\neg p}, \beta_{n-1}^k, 0) = t(\beta_{n-1}^k) \sum_{\substack{x \in \beta_{n-1}^k \\ F_{\neg p} = 0}} \mathrm{sgn}(JF_{\neg p}(x)),$$

where $t(\beta_{n-1}^k) = +1$ or -1 depending on whether β_{n-1}^k has positive orientation or negative orientation, and $JF_{\neg p}(x)$ is the determinant of the Jacobi matrix of $F_{\neg p}$ at x . (See Theorem (5.2) and (7.2) in Chapter I of [4].) So, under this assumption, we can simplify the formula in Theorem 2.2.

THEOREM 2.3. *Suppose that all the conditions of Theorem 2.2 are satisfied and the Jacobi matrices of* $F_{\neg p}$ *are non-singular at all solutions of* $F_{\neg p} = 0$ *on* β_{n-1}^k *for each* $k \in K_0(s)$*. Then*

$$\mathrm{d}(F, \mathbf{D}, 0) = (-1)^{p-1} s \sum_{k \in K_0(s)} t(\beta_{n-1}^k) \sum_{\substack{x \in \beta_{n-1}^k \\ F_{\neg p}(x) = 0}} \mathrm{sgn}(JF_{\neg p}(x)),$$

where $t(\beta_{n-1}^k) = +1$ *or* -1 *depending on whether* β_{n-1}^k *has positive orientation or negative orientation, and* $JF_{\neg p}(x)$ *is the determinant of the Jacobi matrix of* $F_{\neg p}$ *at* x*.*

In our context, the region $\mathbf{D}$ is an n-dimensional box

$$\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n),$$

where $n \ge 2$ and $\boldsymbol{x}_k = [\underline{x}_k, \overline{x}_k]$. The boundary $\boldsymbol{\partial x}$ of $\boldsymbol{x}$ consists of $2n$ $(n-1)$-dimensional boxes

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

and

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

where $k = 1, \ldots, n$.

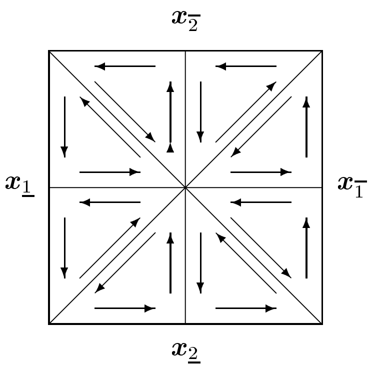

FIG. 2.1. *Positive center decomposition of* $\boldsymbol{x}$ *when* $n = 2$.

Before introducing the next theorem, for completeness in the exposition, we construct the so-called positive or negative *center decomposition* of $\boldsymbol{x}$ (although such decompositions are undoubtedly known among experts). This center decomposition shows us how to specify orientations of the faces of $\boldsymbol{x}$. We follow the convention that an n-simplex $< P_0, P_1, \ldots, P_n >$ in $\mathbb{R}^n$ has positive or negative orientation if the determinant $\Delta_n((1, P_0), (1, P_1), \ldots, (1, P_n))$ is positive or negative. Here, $P_i = (x_{i1}, \ldots, x_{in}), i = 0, 1, \ldots, n$, are $n + 1$ points in $\mathbb{R}^n$ represented by row vectors, and $(1, P_i) = (1, x_{i1}, \ldots, x_{in}), i = 0, 1, \ldots, n$, are $(n+1)$-dimensional vectors that form the $n + 1$ rows of the determinant. See [3, 4, 7] for further details.

We construct the center decompositions of $\boldsymbol{x}$ inductively. Without loss of generality, assume $\boldsymbol{x}_k = [-1, +1], k = 1, \ldots, n$. So, the center of $\boldsymbol{x}$ is $(0, \ldots, 0)$. When $n = 2$ the positive center decomposition is shown in Figure 2.1, in which $P_0 = (0, 0)$. It's easy to check that each of the $2^2 \cdot 2!$ 2-simplexes has positive orientation. So, $\boldsymbol{x}$ has positive orientation. Sometimes we say the orientation is $+1$ or -1 if it's positive or negative. It's clear that the orientations of $\boldsymbol{x}_{\underline{1}}$, $\boldsymbol{x}_{\overline{1}}$, $\boldsymbol{x}_{\underline{2}}$ and $\boldsymbol{x}_{\overline{2}}$ are -1, $+1$, $+1$ and -1. The negative center decomposition of $\boldsymbol{x}$ can be obtained by taking all the simplexes with the negative orientation. Sometimes we say a center decomposition with sign $+1$ or -1 if it's positive or negative, respectively.

Suppose we already have center decompositions of $(n-1)$-dimensional $(n > 2)$ boxes with each decomposition consisting of $2^{n-1} \cdot (n-1)!$ $(n-1)$-simplexes . We construct the positive center decomposition of an n-dimensional box $\boldsymbol{x}$ as follows. We take the center decomposition with sign $(-1)^k$ of the $(n-1)$-dimensional box $\boldsymbol{x}_{\underline{k}}$ and we take the center decomposition with sign $(-1)^{k+1}$ of the $(n-1)$-dimensional box $\boldsymbol{x}_{\overline{k}}$, where $k = 1, \ldots, n$.

We use

$$< Q_0^{(k,-1)}, Q_1^{(k,-1)}, \ldots, Q_{n-1}^{(k,-1)} >$$

or

$$< Q_0^{(k,+1)}, Q_1^{(k,+1)}, \ldots, Q_{n-1}^{(k,+1)} >$$

to denote an $(n-1)$-simplex in the center decomposition of $\boldsymbol{x}_{\underline{k}}$ or $\boldsymbol{x}_{\overline{k}}$, where

$$Q_i = (x_{i1}, \ldots, x_{i(k-1)}, x_{i(k+1)}, \ldots, x_{in})$$

and

$$Q_i^{(k,\pm 1)} = (x_{i1}, \ldots, x_{i(k-1)}, \pm 1, x_{i(k+1)}, \ldots, x_{in}).$$

So,

$$\begin{array}{rl} & \mathrm{sgn}(\Delta_{n-1}((1, Q_0), (1, Q_1), \ldots, (1, Q_{n-1}))) \\ = & (-1)^k \text{ on } \boldsymbol{x}_{\underline{k}} \text{ and } (-1)^{k+1} \text{ on } \boldsymbol{x}_{\overline{k}}. \end{array}$$

For each $(n-1)$-simplex

$$< Q_0^{(k,-1)}, Q_1^{(k,-1)}, \ldots, Q_{n-1}^{(k,-1)}) >$$

or

$$< Q_0^{(k,+1)}, Q_1^{(k,+1)}, \ldots, Q_{n-1}^{(k,+1)}) >,$$

we add the center $P_0 = (0, 0, \ldots, 0)$ of $\boldsymbol{x}$ as the first point to form an n-simplex

$$< P_0, Q_0^{(k,-1)}, Q_1^{(k,-1)}, \ldots, Q_{n-1}^{(k,-1)} >$$

or

$$< P_0, Q_0^{(k,+1)}, Q_1^{(k,+1)}, \ldots, Q_{n-1}^{(k,+1)} > .$$

It's easy to see that there are altogether $2^n \cdot n!$ n-simplexes. Next we verify that each of those n-simplexes has the positive orientation.

$$\begin{array}{rl} & \Delta_n((1, P_0), (1, Q_0^{(k,-1)}), (1, Q_1^{(k,-1)}), \ldots, (1, Q_{n-1}^{(k,-1)})) \\ = & \Delta_{n-1}(Q_0^{(k,-1)}, Q_1^{(k,-1)}, \ldots, Q_{n-1}^{(k,-1)}) \\ = & (-1)^k \Delta_{n-1}((1, Q_0), (1, Q_1), \ldots, (1, Q_{n-1})). \\ & \Delta_n((1, P_0), (1, Q_0^{(k,+1)}), (1, Q_1^{(k,+1)}), \ldots, (1, Q_{n-1}^{(k,+1)})) \\ = & \Delta_{n-1}(Q_0^{(k,+1)}, Q_1^{(k,+1)}, \ldots, Q_{n-1}^{(k,+1)}) \\ = & (-1)^{k-1} \Delta_{n-1}((1, Q_0), (1, Q_1), \ldots, (1, Q_{n-1})). \end{array}$$

So,

$$\begin{array}{rl} & \mathrm{sgn}(\Delta_n((1, P_0), (1, Q_0^{(k,-1)}), (1, Q_1^{(k,-1)}), \ldots, (1, Q_{n-1}^{(k,-1)}))) \\ = & (-1)^k (-1)^k = +1, \text{ and} \\ & \mathrm{sgn}(\Delta_n((1, P_0), (1, Q_0^{(k,+1)}), (1, Q_1^{(k,+1)}), \ldots, (1, Q_{n-1}^{(k,+1)}))) \\ = & (-1)^{k-1} (-1)^{k+1} = +1. \end{array}$$

This means we actually do get a positive center decomposition of $\boldsymbol{x}$. We can get a negative center decomposition by taking all the $2^n \cdot n!$ n-simplexes with negative orientation.

From the above we know that if we decompose $\boldsymbol{x}$ by the positive center decomposition, then $\boldsymbol{x}$ has positive orientation, the induced orientation of $\boldsymbol{x}_{\underline{k}}$ is $(-1)^k$, and the induced orientation of $\boldsymbol{x}_{\overline{k}}$ is $(-1)^{k+1}$. $b(\boldsymbol{x})$ can be divided into $\boldsymbol{x}_{\underline{k}}$ and $\boldsymbol{x}_{\overline{k}}$, $k = 1, \ldots, n$, with the associated orientations. So, $F \neq 0$ on $b(\boldsymbol{x})$ is the same as $F \neq 0$ on $\boldsymbol{\partial x}$. Similarly, $F_{\neg p}(x) = 0$ on $b(\boldsymbol{x}_{\underline{k}})$ or $b(\boldsymbol{x}_{\overline{k}})$ is the same as $F_{\neg p}(x) = 0$ on $\boldsymbol{\partial x}_{\underline{k}}$ or $\boldsymbol{\partial x}_{\overline{k}}$. Let $\underline{K_0(s)}$ denote the subset of the integers $k \in \{1, \ldots, n\}$ such that $F_{\neg p} = 0$ has solutions on $\boldsymbol{x}_{\underline{k}}$ and $\mathrm{sgn}(f_p) = s$ at all the solutions, and $\overline{K_0(s)}$ denote the subset of the integers $k \in \{1, \ldots, n\}$ such that $F_{\neg p} = 0$ has solutions on $\boldsymbol{x}_{\overline{k}}$ and $\mathrm{sgn}(f_p) = s$ at all the solutions, where $s \in \{-1, +1\}$. Then, by Theorem 2.3, we have

THEOREM 2.4. *Suppose $F \neq 0$ on $\boldsymbol{\partial x}$, and there is p, $1 \leq p \leq n$, such that:*

1. *$F_{\neg p} \neq 0$ on $\boldsymbol{\partial x}_{\underline{k}}$ or $\boldsymbol{\partial x}_{\overline{k}}$, $k = 1, \ldots, n$;*

$$
Y\boldsymbol{F'}(\boldsymbol{x}) = \begin{pmatrix}
1 & 0 & \ldots & 0 & \overbrace{*\ldots*}^{p} \\
0 & 1 & 0\ldots & 0 & *\ldots* \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & \ldots & 0 & 1 & *\ldots* \\
0 & \ldots & 0 & 0 & 0\ldots0 \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \ldots & 0 & 0 & 0\ldots0
\end{pmatrix}.
$$

FIG. 3.1. *A singular system of rank $n-p$ preconditioned with an incomplete LU factorization, where "$*$" represents a non-zero element.*

2. *f_p has the same sign at all solutions of $F_{\neg p}=0$, if there are any, on $\boldsymbol{x}_{\underline{k}}$ or $\boldsymbol{x}_{\overline{k}}$, $1 \le k \le n$; and*
3. *the Jacobi matrices of $F_{\neg p}$ are non-singular at all solutions of $F_{\neg p}=0$ on $\boldsymbol{\partial x}$.*

Then

$$
\begin{aligned}
&\mathrm{d}(F,\boldsymbol{x},0) \\
&= (-1)^{p-1}s\Bigg\{\sum_{k\in\underline{K_0(s)}}(-1)^k \sum_{\substack{x\in\boldsymbol{x}_{\underline{k}}\\ F_{\neg p}(x)=0}} \mathrm{sgn}\left|\frac{\partial F_{\neg p}}{\partial x_1 x_2\ldots x_{k-1}x_{k+1}\ldots x_n}(x)\right| \\
&\quad + \sum_{k\in\overline{K_0(s)}}(-1)^{k+1} \sum_{\substack{x\in\boldsymbol{x}_{\overline{k}}\\ F_{\neg p}(x)=0}} \mathrm{sgn}\left|\frac{\partial F_{\neg p}}{\partial x_1 x_2\ldots x_{k-1}x_{k+1}\ldots x_n}(x)\right|\Bigg\}.
\end{aligned}
$$

3. On Preconditioning. The inverse midpoint preconditioner approximately *diagonalizes* the interval Jacobi matrix, when $F'(x^*)$ is non-singular (and well-enough conditioned). This preconditioner can be computed with Gaussian elimination with partial pivoting. We can compute (to within a series of row permutations) an *LU* factorization of the midpoint matrix $\mathrm{m}\big(\boldsymbol{F'}(\boldsymbol{X})\big)$. The factors L and U may then be applied to actually precondition the interval linear system.

When the rank of $F'(x^*)$ is $n-p$ for some $p>0$, then Gaussian elimination with full pivoting can be used to reduce $\boldsymbol{F'}(\boldsymbol{x})$ to approximately the pattern shown in Figure 3.1. Actually, an incomplete factorization based on full pivoting will put the system into a pattern that resembles a permutation of the columns of the pattern in Figure 3.1. However, for notational simplicity, there is no loss here in assuming exactly the form in Figure 3.1.

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 3.1. In the rest of this paper, we concentrate on the case p=1, although the idea can be applied to the general case.

4. The Complex Setting and System Form. In the remainder of this paper, we assume

1. $F:\mathbf{D}\subset\mathbb{R}^n\to\mathbb{R}^n$ can be extended to an analytic function in $\mathbb{C}^n$.

2. $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = ([\underline{x}_1, \overline{x}_1], \ldots, [\underline{x}_n, \overline{x}_n])$ is a small box that will be constructed centered at the approximate solution $\check{x}$, i.e. $m(\boldsymbol{x}) = (\check{x}_1, \ldots, \check{x}_n)$.
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 for a quadratic model to be accurate over the box $\boldsymbol{x}$.
4. F has been preconditioned as in Figure 3.1, and $F'(x^*)$ has null space of dimension 1.

So,

$$f_k(x) = (x_k - \check{x}_k) + \frac{\partial f_k}{\partial x_n}(\check{x})(x_n - \check{x}_n) + \mathcal{O}\left(\|x - \check{x}\|^2\right)$$
$$\text{for } 1 \le k \le n-1.$$
$$f_n(x) = \frac{1}{2}\sum_{k=1}^{n}\sum_{l=1}^{n} \frac{\partial^2 f_n}{\partial x_k \partial x_l}(\check{x})(x_k - \check{x}_k)(x_l - \check{x}_l) + \mathcal{O}\left(\|x - \check{x}\|^3\right)$$

For $F : \mathbb{R}^n \to \mathbb{R}^n$, extend F to complex space: $x + iy$, with y in a small box

$$\boldsymbol{y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n) = ([\underline{y}_1, \overline{y}_1], \ldots, [\underline{y}_n, \overline{y}_n]),$$

where $\boldsymbol{y}$ is centered at $(0, \ldots, 0)$. Define $\boldsymbol{z} \equiv (\boldsymbol{x}_1, \boldsymbol{y}_1, \ldots, \boldsymbol{x}_n, \boldsymbol{y}_n) = ([\underline{x}_1, \overline{x}_1], [\underline{y}_1, \overline{y}_1], \ldots, [\underline{x}_n, \overline{x}_n], [\underline{y}_n, \overline{y}_n])$, $u_k(x, y) \equiv \Re(f_k(x + iy))$ and $v_k(x, y) \equiv \Im(f_k(x + iy))$. With this, define

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

Then, if preconditioning based on complete factorization of the midpoint matrix for $\boldsymbol{F'}(\boldsymbol{x})$ is used, the first-order terms are eliminated in the pattern of Figure 3.1, and, for $1 \le k \le (n-1)$,

$$\left.\begin{array}{rcl} u_k(x, y) & = & (x_k - \check{x}_k) + \frac{\partial f_k}{\partial x_n}(\check{x})(x_n - \check{x}_n) \\ & & + \mathcal{O}\left(\|(x - \check{x}, y)\|^2\right), \\ v_k(x, y) & = & y_k + \frac{\partial f_k}{\partial x_n}(\check{x}) y_n + \mathcal{O}\left(\|(x - \check{x}, y)\|^2\right), \end{array}\right\} \tag{4.1}$$

and

$$\left.\begin{array}{rcl} u_n(x, y) & = & \frac{1}{2}\sum_{k=1}^{n}\sum_{l=1}^{n} \frac{\partial^2 f_n}{\partial x_k \partial x_l}(\check{x})(x_k - \check{x}_k)(x_l - \check{x}_l) \\ & & -\frac{1}{2}\sum_{k=1}^{n}\sum_{l=1}^{n} \frac{\partial^2 f_n}{\partial x_k \partial x_l}(\check{x}) y_k y_l \\ & & + \mathcal{O}\left(\|(x - \check{x}, y)\|^3\right), \\ v_n(x, y) & = & \sum_{k=1}^{n}\sum_{l=1}^{n} \frac{\partial^2 f_n}{\partial x_k \partial x_l}(\check{x})(x_k - \check{x}_k) y_l \\ & & + \mathcal{O}\left(\|(x - \check{x}, y)\|^3\right). \end{array}\right\} \tag{4.2}$$

5. Simplification of a Degree Computation Procedure. To use Theorem 2.4 to compute the topological degree $\mathrm{d}(\tilde{F}, \boldsymbol{z}, 0)$ directly in a verification algorithm would require a global search of the $4n$ $(n-1)$-dimensional faces of the $2n$-dimensional box $\boldsymbol{z}$ for zeros of $\tilde{F}_{\neg p}$. This is 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. However, if the system is preconditioned and in the form described in §3 and §4, the verification can be reduced to $4n - 4$ interval evaluations and four 1-dimensional searches.

To describe the simplification, define

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

and

$$\boldsymbol{x}_{\overline{k}} \equiv (\boldsymbol{x}_1, \boldsymbol{y}_1, \ldots, \boldsymbol{x}_{k-1}, \boldsymbol{y}_{k-1}, \overline{x}_k, \boldsymbol{y}_k, \boldsymbol{x}_{k+1}, \boldsymbol{y}_{k+1}, \ldots, \boldsymbol{x}_n, \boldsymbol{y}_n).$$

Similarly define $\boldsymbol{y}_{\underline{k}}$ and $\boldsymbol{y}_{\overline{k}}$. Also define

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

To compute the degree $\mathrm{d}(\tilde{F}, \boldsymbol{z}, 0)$, we will consider $\tilde{F}_{\neg u_n}$ on the boundary of $\boldsymbol{z}$. The boundary of $\boldsymbol{z}$ consists of the $4n$ faces $\boldsymbol{x}_{\underline{1}}$, $\boldsymbol{x}_{\overline{1}}$, $\boldsymbol{y}_{\underline{1}}$, $\boldsymbol{y}_{\overline{1}}$, …, $\boldsymbol{x}_{\underline{n}}$, $\boldsymbol{x}_{\overline{n}}$, $\boldsymbol{y}_{\underline{n}}$, $\boldsymbol{y}_{\overline{n}}$.

Observe that, for $1 \le k \le (n-1)$,
$\tilde{F}_{\neg u_n}(x,y) = 0$ on $\boldsymbol{x}_{\underline{k}}$

$$\begin{aligned}
&\Longrightarrow & u_k(x,y) \approx (\underline{x}_k - \check{x}_k) + \frac{\partial f_k}{\partial x_n}(\check{x})(x_n - \check{x}_n) = 0\\
&\Longrightarrow & |\underline{x}_k - \check{x}_k| = |\partial f_k/\partial x_n(\check{x})||x_n - \check{x}_n|\\
&\Longrightarrow & \mathrm{w}(\boldsymbol{x}_k) \le |\partial f_k/\partial x_n(\check{x})|\mathrm{w}(\boldsymbol{x}_n), \text{ or } \frac{\mathrm{w}(\boldsymbol{x}_k)}{|\partial f_k/\partial x_n(\check{x})|} \le \mathrm{w}(\boldsymbol{x}_n).
\end{aligned}$$

Similarly,
$\tilde{F}_{\neg u_n}(x,y) = 0$ on $\boldsymbol{x}_{\overline{k}}$

$$\Longrightarrow \mathrm{w}(\boldsymbol{x}_k) \le |\partial f_k/\partial x_n(\check{x})|\mathrm{w}(\boldsymbol{x}_n), \text{ or } \frac{\mathrm{w}(\boldsymbol{x}_k)}{|\partial f_k/\partial x_n(\check{x})|} \le \mathrm{w}(\boldsymbol{x}_n).$$

Also observe that, for $1 \le k \le (n-1)$,
$\tilde{F}_{\neg u_n}(x,y) = 0$ on $\boldsymbol{y}_{\underline{k}}$,

$$\begin{aligned}
&\Longrightarrow & u_k(x,y) \approx \underline{y}_k + \frac{\partial f_k}{\partial x_n}(\check{x})y_n = 0\\
&\Longrightarrow & |\underline{y}_k| = |\partial f_k/\partial x_n(\check{x})||y_n|\\
&\Longrightarrow & \mathrm{w}(\boldsymbol{y}_k) \le |\partial f_k/\partial x_n(\check{x})|\mathrm{w}(\boldsymbol{y}_n), \text{ or } \frac{\mathrm{w}(\boldsymbol{y}_k)}{|\partial f_k/\partial x_n(\check{x})|} \le \mathrm{w}(\boldsymbol{y}_n).
\end{aligned}$$

Similarly,
$\tilde{F}_{\neg u_n}(x,y) = 0$ on $\boldsymbol{y}_{\overline{k}}$

$$\Longrightarrow \mathrm{w}(\boldsymbol{y}_k) \le |\partial f_k/\partial x_n(\check{x})|\mathrm{w}(\boldsymbol{y}_n), \text{ or } \frac{\mathrm{w}(\boldsymbol{y}_k)}{|\partial f_k/\partial x_n(\check{x})|} \le \mathrm{w}(\boldsymbol{y}_n).$$

Thus, if $\boldsymbol{x}_n$ is chosen so that

$$\mathrm{w}(\boldsymbol{x}_n) \le \frac{1}{2} \min_{1\le k\le n-1} \left\{ \frac{\mathrm{w}(\boldsymbol{x}_k)}{|\partial f_k/\partial x_n(\check{x})|} \right\}, \tag{5.1}$$

then it is unlikely that $u_k(x,y) = 0$ on either $\boldsymbol{x}_{\underline{k}}$ or $\boldsymbol{x}_{\overline{k}}$.

Similarly, if $\boldsymbol{y}_n$ is chosen so that

$$\mathrm{w}(\boldsymbol{y}_n) \le \frac{1}{2} \min_{1\le k\le n-1} \left\{ \frac{\mathrm{w}(\boldsymbol{y}_k)}{|\partial f_k/\partial x_n(\check{x})|} \right\}, \tag{5.2}$$

then it is unlikely that $v_k(x,y)=0$ on either $\boldsymbol{y}_{\underline{k}}$ or $\boldsymbol{y}_{\overline{k}}$.

Here, the coefficient $\frac{1}{2}$ is to take into consideration the fact that $u_k(x,y) \approx (x_k - \check{x}_k) + \frac{\partial f_k}{\partial x_n}(\check{x})(x_n - \check{x}_n)$ and $v_k(x,y) \approx y_k + \frac{\partial f_k}{\partial x_n}(\check{x}) y_n$ are only approximate equalities.

When $\partial f_k/\partial x_n(\check{x})$ happens to be 0, we can think that $\frac{\mathrm{w}(\boldsymbol{x}_k)}{|\partial f_k/\partial x_n(\check{x})|}$ and $\frac{\mathrm{w}(\boldsymbol{y}_k)}{|\partial f_k/\partial x_n(\check{x})|}$ are $+\infty$.

By constructing the box $\boldsymbol{z}$ in this way, we can eliminate search of $4n-4$ of the $4n$ faces of the boundary of $\boldsymbol{z}$, since we have arranged to verify $\tilde{F}_{\neg u_n}(x,y) \neq 0$ on each of these faces. Elimination of these $4n-4$ faces 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. This reduces total computational cost dramatically, since searching a face is expensive. The following theorem forms the theoretical basis of our algorithm in §6.1.

THEOREM 5.1. *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 a unique solution 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 a unique solution 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 four 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 four solutions of* $\tilde{F}_{\neg u_n}=0$ *in condition 2.*

Then

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

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 2 and 3 imply $\tilde{F} \neq 0$ on $\boldsymbol{x}_{\underline{n}}$, $\boldsymbol{x}_{\overline{n}}$, $\boldsymbol{y}_{\underline{n}}$ and $\boldsymbol{y}_{\overline{n}}$. So, $\tilde{F} \neq 0$ on $\boldsymbol{\partial z}$.

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 $2(n-1)$ $(n-2)$-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 $\boldsymbol{y}_{\underline{n}}$ or $\boldsymbol{y}_{\overline{n}}$. So, by 2 and 3, $\tilde{F}_{\neg u_n} \neq 0$ 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 $\boldsymbol{\partial y}_{\overline{n}}$. So, condition 1 in Theorem 2.4 is satisfied.

Condition 2 in Theorem 2.4 is automatically satisfied since $F_{\neg p} = 0$ either has no solutions or a unique solution on $\boldsymbol{x}_{\underline{k}}$, $\boldsymbol{x}_{\overline{k}}$, $\boldsymbol{y}_{\underline{k}}$, or $\boldsymbol{y}_{\overline{k}}$, $1 \le k \le n$.

Then, by noticing 4, the conditions of Theorem 2.4 are satisfied. The formula is thus obtained with $s = +1$. $\square$

The conditions of Theorem 5.1 will be satisfied when the system is that as described in §3 and §4, the box $\boldsymbol{z}$ is constructed as in (5.1) and (5.2), and the quadratic model is accurate. (See Theorem 5.2 and its proof of the results when all the approximations are exact.)

In Theorem 5.1, the degree consists of contributions of the four faces we search. We can compute the degree contribution of each of the four faces and then add them to get the degree.

In Theorem 5.1 we choose $s = +1$. We can also choose $s = -1$. That doesn't make any difference in our context if we ignore higher order terms in the values of u_n at the solutions of $\tilde{F}_{\neg u_n} = 0$ on the four faces $\boldsymbol{x}_{\underline{n}}$, $\boldsymbol{x}_{\overline{n}}$, $\boldsymbol{y}_{\underline{n}}$ and $\boldsymbol{y}_{\overline{n}}$. To be specific, the four values of u_n are

$$
\begin{aligned}
u_n &= \frac{1}{2}\Delta(\underline{x}_n - \check{x}_n)^2 + \mathcal{O}\left(\|(x - \check{x}, y)\|^3\right),\\
u_n &= \frac{1}{2}\Delta(\overline{x}_n - \check{x}_n)^2 + \mathcal{O}\left(\|(x - \check{x}, y)\|^3\right),\\
u_n &= -\frac{1}{2}\Delta\underline{y}_n^2 + \mathcal{O}\left(\|(x - \check{x}, y)\|^3\right),\\
u_n &= -\frac{1}{2}\Delta\overline{y}_n^2 + \mathcal{O}\left(\|(x - \check{x}, y)\|^3\right),
\end{aligned}
$$

respectively, where, Δ is defined in (5.3). When we choose $\mathrm{w}(\boldsymbol{y}_k)$ the same (or roughly the same) as $\mathrm{w}(\boldsymbol{x}_k)$, the values of u_n as a function of $\underline{y}_n$ (or $\overline{y}_n$) will be the same (or roughly the same) as the values of u_n as a function of $\underline{x}_n - \check{x}_n$ (or $\overline{x}_n - \check{x}_n$). So, if we ignore higher order terms, the cost of verifying $u_n < 0$ and searching for solutions of $\tilde{F}_{\neg u_n} = 0$ with $u_n > 0$ is approximately the same as the cost of verifying $u_n > 0$ and searching for solutions of $\tilde{F}_{\neg u_n} = 0$ with $u_n < 0$.

Next, we will give a theorem which will further reduce the search cost by telling us how we should search. Before introducing the theorem, define

$$
\begin{aligned}
\alpha_k &\equiv \frac{\partial f_k}{\partial x_n}(\check{x}), \qquad 1 \le k \le n-1,\\
\alpha_n &\equiv -1,\\
\beta_{kl} &\equiv \frac{\partial^2 f_n}{\partial x_k \partial x_l}(\check{x}) \qquad 1 \le k \le n, 1 \le l \le n,\\
\Delta &\equiv \sum_{k=1}^{n}\sum_{l=1}^{n} \beta_{kl}\alpha_k\alpha_l
\end{aligned}
\tag{5.3}
$$

Note $\beta_{kl} = \beta_{lk}$

THEOREM 5.2. *If the approximations of (4.1) and (4.2) are exact, if we construct the box $\boldsymbol{z}$ as in (5.1) and (5.2), and if $\Delta \ne 0$, then* $\mathrm{d}(\tilde{F}, \boldsymbol{z}, 0) = 2$.

Proof. Under the assumptions,

$$
u_k = (x_k - \check{x}_k) + \alpha_k(x_n - \check{x}_n), \qquad 1 \le k \le n-1,
\tag{5.4}
$$

$$v_k = y_k + \alpha_k y_n, \qquad 1 \le k \le n-1, \tag{5.5}$$

$$u_n = \frac{1}{2}\sum_{k=1}^{n}\sum_{l=1}^{n}\beta_{kl}(x_k - \check{x}_k)(x_l - \check{x}_l) - \frac{1}{2}\sum_{k=1}^{n}\sum_{l=1}^{n}\beta_{kl}y_k y_l \tag{5.6}$$

$$v_n = \sum_{k=1}^{n}\sum_{l=1}^{n}\beta_{kl}(x_k - \check{x}_k)y_l \tag{5.7}$$

Due to the construction of the box $\boldsymbol{z}$,

$$u_k = (\underline{x}_k - \check{x}_k) + \alpha_k(x_n - \check{x}_n) \neq 0$$

on $\boldsymbol{x}_{\underline{k}}$ and $\boldsymbol{x}_{\overline{k}}$, and

$$v_k = \underline{y}_k + \alpha_k y_n \neq 0$$

on $\boldsymbol{y}_{\underline{k}}$ and $\boldsymbol{y}_{\overline{k}}$, where $k = 1, \ldots, n-1$.

Next, we locate the solutions of $\tilde{F}_{\neg u_n} = 0$ on $\boldsymbol{x}_{\underline{n}}$, $\boldsymbol{x}_{\overline{n}}$, $\boldsymbol{y}_{\underline{n}}$ and $\boldsymbol{y}_{\overline{n}}$.

1) On $\boldsymbol{x}_{\underline{n}}$;

$$u_k = 0 \Longrightarrow \tilde{x}_k = \check{x}_k - \alpha_k(\underline{x}_n - \check{x}_n), \qquad 1 \le k \le n-1, \tag{5.8}$$

$$v_k = 0 \Longrightarrow \tilde{y}_k = -\alpha_k y_n, \qquad 1 \le k \le n-1, \tag{5.9}$$

Plugging equations (5.8) and (5.9) into equations (5.6) and (5.7), we get

$$\begin{aligned} u_n &= \frac{1}{2}\sum_{k=1}^{n}\sum_{l=1}^{n}\beta_{kl}\alpha_k\alpha_l(\underline{x}_n - \check{x}_n)^2 - \frac{1}{2}\sum_{k=1}^{n}\sum_{l=1}^{n}\beta_{kl}\alpha_k\alpha_l y_n^2 \\ &= \frac{1}{2}\Delta(\underline{x}_n - \check{x}_n)^2 - \frac{1}{2}\Delta y_n^2 \end{aligned} \tag{5.10}$$

$$\begin{aligned} v_n &= \sum_{k=1}^{n}\sum_{l=1}^{n}\beta_{kl}\alpha_k\alpha_l(\underline{x}_n - \check{x}_n)y_n \\ &= \Delta(\underline{x}_n - \check{x}_n)y_n. \end{aligned} \tag{5.11}$$

Then,

$$v_n = 0 \Longrightarrow \tilde{y}_n = 0, \tag{5.12}$$

since $\Delta \neq 0$.

Thus, by equation (5.9)

$$\tilde{y}_n = 0 \Longrightarrow \tilde{y}_k = 0, \qquad 1 \le k \le n-1, \tag{5.13}$$

So, $\tilde{F}_{\neg u_n} = 0$ has a unique solution

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

on $\boldsymbol{x}_{\underline{n}}$.

Plugging (5.12) into (5.10), we get the u_n value at this solution, which is

$$u_n = \frac{1}{2}\Delta(\underline{x}_n - \check{x}_n)^2. \tag{5.14}$$

Next, we compute the determinant of the Jacobi matrix of $\tilde{F}_{\neg u_n}$ at this solution. Define

$$\gamma_k \equiv \sum_{l=1}^{n} \beta_{lk}\alpha_l.$$

Noting equations (5.4), (5.5) and (5.7), we have

$$\begin{aligned}
&\left|\frac{\partial \tilde{F}_{\neg u_n}}{\partial x_1 y_1 \ldots x_{n-1} y_{n-1} y_n}(\tilde{x},\tilde{y})\right| \\
&= \begin{vmatrix}
\frac{\partial u_1}{\partial x_1} & \frac{\partial u_1}{\partial y_1} & \cdots & \frac{\partial u_1}{\partial x_{n-1}} & \frac{\partial u_1}{\partial y_{n-1}} & \frac{\partial u_1}{\partial y_n} \\
\frac{\partial v_1}{\partial x_1} & \frac{\partial v_1}{\partial y_1} & \cdots & \frac{\partial v_1}{\partial x_{n-1}} & \frac{\partial v_1}{\partial y_{n-1}} & \frac{\partial v_1}{\partial y_n} \\
\vdots & \vdots & & \vdots & \vdots & \vdots \\
\frac{\partial u_{n-1}}{\partial x_1} & \frac{\partial u_{n-1}}{\partial y_1} & \cdots & \frac{\partial u_{n-1}}{\partial x_{n-1}} & \frac{\partial u_{n-1}}{\partial y_{n-1}} & \frac{\partial u_{n-1}}{\partial y_n} \\
\frac{\partial v_{n-1}}{\partial x_1} & \frac{\partial v_{n-1}}{\partial y_1} & \cdots & \frac{\partial v_{n-1}}{\partial x_{n-1}} & \frac{\partial v_{n-1}}{\partial y_{n-1}} & \frac{\partial v_{n-1}}{\partial y_n} \\
\frac{\partial v_n}{\partial x_1} & \frac{\partial v_n}{\partial y_1} & \cdots & \frac{\partial v_n}{\partial x_{n-1}} & \frac{\partial v_n}{\partial y_{n-1}} & \frac{\partial v_n}{\partial y_n}
\end{vmatrix} \\
&= \begin{vmatrix}
1 & 0 & \ldots & 0 & 0 & 0 \\
0 & 1 & \ldots & 0 & 0 & \alpha_1 \\
\vdots & \vdots & & \vdots & \vdots & \vdots \\
0 & 0 & \ldots & 1 & 0 & 0 \\
0 & 0 & \ldots & 0 & 1 & \alpha_{n-1} \\
0 & -\gamma_1(\underline{x}_n - \check{x}_n) & \ldots & 0 & -\gamma_{n-1}(\underline{x}_n - \check{x}_n) & -\gamma_n(\underline{x}_n - \check{x}_n)
\end{vmatrix} \\
&= -(\underline{x}_n - \check{x}_n)\begin{vmatrix}
1 & 0 & \ldots & 0 & 0 & 0 \\
0 & 1 & \ldots & 0 & 0 & \alpha_1 \\
\vdots & \vdots & & \vdots & \vdots & \vdots \\
0 & 0 & \ldots & 1 & 0 & 0 \\
0 & 0 & \ldots & 0 & 1 & \alpha_{n-1} \\
0 & \gamma_1 & \ldots & 0 & \gamma_{n-1} & \gamma_n
\end{vmatrix} \\
&= -(\underline{x}_n - \check{x}_n)(-\sum_{k=1}^{n}\alpha_k\gamma_k) \\
&= (\underline{x}_n - \check{x}_n)\sum_{k=1}^{n}\alpha_k\sum_{l=1}^{n}\beta_{lk}\alpha_l \\
&= (\underline{x}_n - \check{x}_n)\sum_{k=1}^{n}\sum_{l=1}^{n}\beta_{lk}\alpha_k\alpha_l \\
&= (\underline{x}_n - \check{x}_n)\sum_{k=1}^{n}\sum_{l=1}^{n}\beta_{kl}\alpha_k\alpha_l \\
&= (\underline{x}_n - \check{x}_n)\Delta.
\end{aligned} \tag{5.15}$$

2) On $\boldsymbol{x}_{\overline{n}}$;
Similarly, $\tilde{F}_{\neg u_n} = 0$ has a unique solution $(\tilde{x}, \tilde{y})$ on $\boldsymbol{x}_{\overline{n}}$.
The u_n value at this solution is

$$u_n = \frac{1}{2}\Delta(\overline{x}_n - \check{x}_n)^2. \tag{5.16}$$

The determinant of the Jacobi matrix of $\tilde{F}_{\neg u_n}$ at this solution is

$$\left| \frac{\partial \tilde{F}_{\neg u_n}}{\partial x_1 y_1 \ldots x_{n-1} y_{n-1} y_n}(\tilde{x}, \tilde{y}) \right| = (\overline{x}_n - \check{x}_n)\Delta \tag{5.17}$$

3) On $\boldsymbol{y}_{\underline{n}}$;

$$u_k = 0 \Longrightarrow \tilde{x}_k = \check{x}_k - \alpha_k(x_n - \check{x}_n), \qquad 1 \le k \le n-1, \tag{5.18}$$

$$v_k = 0 \Longrightarrow \tilde{y}_k = -\alpha_k \underline{y}_n, \qquad 1 \le k \le n-1, \tag{5.19}$$

Plugging equations (5.18) and (5.19) into equations (5.6) and (5.7), we get

$$\begin{aligned} u_n &= \frac{1}{2}\sum_{k=1}^{n}\sum_{l=1}^{n} \beta_{kl}\alpha_k\alpha_l(x_n - \check{x}_n)^2 - \frac{1}{2}\sum_{k=1}^{n}\sum_{l=1}^{n} \beta_{kl}\alpha_k\alpha_l \underline{y}_n^2 \\ &= \frac{1}{2}\Delta(x_n - \check{x}_n)^2 - \frac{1}{2}\Delta \underline{y}_n^2 \end{aligned} \tag{5.20}$$

$$\begin{aligned} v_n &= \sum_{k=1}^{n}\sum_{l=1}^{n} \beta_{kl}\alpha_k\alpha_l(x_n - \check{x}_n)\underline{y}_n \\ &= \Delta(x_n - \check{x}_n)\underline{y}_n. \end{aligned} \tag{5.21}$$

Then,

$$v_n = 0 \Longrightarrow \tilde{x}_n = \check{x}_n, \tag{5.22}$$

since $\Delta \ne 0$.
Thus, by equation (5.18)

$$\tilde{x}_n = \check{x}_n \Longrightarrow \tilde{x}_k = \check{x}_k, \qquad 1 \le k \le n-1, \tag{5.23}$$

So, $\tilde{F}_{\neg u_n} = 0$ has a unique solution

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

on $\boldsymbol{y}_{\underline{n}}$.
Plugging (5.22) into (5.20), we get the u_n value at this solution, which is

$$u_n = -\frac{1}{2}\Delta \underline{y}_n^2. \tag{5.24}$$

Next, we compute the determinant of the Jacobi matrix of $\tilde{F}_{\neg u_n}$ at this solution.

Noting the equations (5.4), (5.5) and (5.7), we have

$$
\begin{aligned}
(5.25)\qquad & \left|\frac{\partial \tilde{F}_{\neg u_n}}{\partial x_1 y_1 \dots x_{n-1} y_{n-1} x_n}(\tilde{x},\tilde{y})\right| \\
&= \begin{vmatrix}
\frac{\partial u_1}{\partial x_1} & \frac{\partial u_1}{\partial y_1} & \dots & \frac{\partial u_1}{\partial x_{n-1}} & \frac{\partial u_1}{\partial y_{n-1}} & \frac{\partial u_1}{\partial x_n} \\
\frac{\partial v_1}{\partial x_1} & \frac{\partial v_1}{\partial y_1} & \dots & \frac{\partial v_1}{\partial x_{n-1}} & \frac{\partial v_1}{\partial y_{n-1}} & \frac{\partial v_1}{\partial x_n} \\
\vdots & \vdots & & \vdots & \vdots & \vdots \\
\frac{\partial u_{n-1}}{\partial x_1} & \frac{\partial u_{n-1}}{\partial y_1} & \dots & \frac{\partial u_{n-1}}{\partial x_{n-1}} & \frac{\partial u_{n-1}}{\partial y_{n-1}} & \frac{\partial u_{n-1}}{\partial x_n} \\
\frac{\partial v_{n-1}}{\partial x_1} & \frac{\partial v_{n-1}}{\partial y_1} & \dots & \frac{\partial v_{n-1}}{\partial x_{n-1}} & \frac{\partial v_{n-1}}{\partial y_{n-1}} & \frac{\partial v_{n-1}}{\partial x_n} \\
\frac{\partial v_n}{\partial x_1} & \frac{\partial v_n}{\partial y_1} & \dots & \frac{\partial v_n}{\partial x_{n-1}} & \frac{\partial v_n}{\partial y_{n-1}} & \frac{\partial v_n}{\partial x_n}
\end{vmatrix} \\
&= \begin{vmatrix}
1 & 0 & \dots & 0 & 0 & \alpha_1 \\
0 & 1 & \dots & 0 & 0 & 0 \\
\vdots & \vdots & & \vdots & \vdots & \vdots \\
0 & 0 & \dots & 1 & 0 & \alpha_{n-1} \\
0 & 0 & \dots & 0 & 1 & 0 \\
-\gamma_1 \underline{y}_n & 0 & \dots & -\gamma_{n-1}\underline{y}_n & 0 & -\gamma_n \underline{y}_n
\end{vmatrix} \\
&= -\underline{y}_n \begin{vmatrix}
1 & 0 & \dots & 0 & 0 & \alpha_1 \\
0 & 1 & \dots & 0 & 0 & 0 \\
\vdots & \vdots & & \vdots & \vdots & \vdots \\
0 & 0 & \dots & 1 & 0 & \alpha_{n-1} \\
0 & 0 & \dots & 0 & 1 & 0 \\
\gamma_1 & 0 & \dots & \gamma_{n-1} & 0 & \gamma_n
\end{vmatrix} \\
&= -\underline{y}_n \Big(-\sum_{k=1}^{n} \alpha_k \gamma_k\Big) \\
&= \underline{y}_n \sum_{k=1}^{n} \alpha_k \sum_{l=1}^{n} \beta_{lk} \alpha_l \\
&= \underline{y}_n \sum_{k=1}^{n} \sum_{l=1}^{n} \beta_{lk} \alpha_k \alpha_l \\
&= \underline{y}_n \sum_{k=1}^{n} \sum_{l=1}^{n} \beta_{kl} \alpha_k \alpha_l \\
&= \underline{y}_n \Delta.
\end{aligned}
$$

4) On $\boldsymbol{y}_{\overline{n}}$;
Similarly, $\tilde{F}_{\neg u_n} = 0$ has a unique solution $(\tilde{x}, \tilde{y})$ on $\boldsymbol{y}_{\overline{n}}$.
The u_n value at this solution is

$$u_n = -\frac{1}{2}\Delta \overline{y}_n^2. \tag{5.26}$$

The determinant of the Jacobi matrix of $\tilde{F}_{\neg u_n}$ at this solution is

$$\left| \frac{\partial \tilde{F}_{\neg u_n}}{\partial x_1 y_1 \ldots x_{n-1} y_{n-1} x_n} (\tilde{x}, \tilde{y}) \right| = \overline{y}_n \Delta \tag{5.27}$$

Finally, we can use the formula in Theorem 5.1 to compute the topological degree $\mathrm{d}(\tilde{F}, \boldsymbol{z}, 0)$.

If $\Delta > 0$, then we know from the equations (5.14), (5.16), (5.24) and (5.26) that u_n is only positive at the solutions of $\tilde{F}_{\neg u_n} = 0$ on $\boldsymbol{x}_{\underline{n}}$ and $\boldsymbol{x}_{\overline{n}}$. We also know the signs of the determinants of the Jacobi matrices at the two solutions from equations (5.15) and (5.17). So,

$$\mathrm{d}(\tilde{F}, \boldsymbol{z}, 0) = -(-1) + (+1) = 2$$

If $\Delta < 0$, then we know from the equations (5.14), (5.16), (5.24) and (5.26) that u_n is only positive at the solutions of $\tilde{F}_{\neg u_n} = 0$ on $\boldsymbol{y}_{\underline{n}}$ and $\boldsymbol{y}_{\overline{n}}$. We also know the signs of the determinants of the Jacobi matrices at the two solutions from equations (5.25) and (5.27). So,

$$\mathrm{d}(\tilde{F}, \boldsymbol{z}, 0) = +(+1) - (-1) = 2$$

□

The proof of Theorem 5.2 tells us approximately where we can expect to find the solutions of $\tilde{F}_{\neg u_n} = 0$ on the four faces we search and the value of the degree we can expect when the approximations (4.1) and (4.2) are accurate.

From (4.1), we know 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 $\mathrm{w}(\tilde{\boldsymbol{x}}_k) = \mathcal{O}\left(\|(\boldsymbol{x} - \check{x}, \boldsymbol{y})\|^2\right)$, $1 \le k \le n-1$. Similarly, if y_n is known precisely, formally solving $\boldsymbol{v}_k(\boldsymbol{x}, \boldsymbol{y}) = 0$ for y_k gives sharper bounds $\tilde{\boldsymbol{y}}_k$ with $\mathrm{w}(\tilde{\boldsymbol{y}}_k) = \mathcal{O}\left(\|(\boldsymbol{x} - \check{x}, \boldsymbol{y})\|^2\right)$, $1 \le k \le n-1$. So, when we search $\boldsymbol{x}_{\underline{n}}$ (or $\boldsymbol{x}_{\overline{n}}$) for solutions of $\tilde{F}_{\neg u_n} = 0$, we can first get sharper bounds for $x_k, 1 \le k \le n-1$, since x_n is known precisely. Then, for a small subinterval $\boldsymbol{y}_n^0$ of $\boldsymbol{y}_n$, we can solve $\boldsymbol{v}_k(\boldsymbol{x}, \boldsymbol{y}) = 0$ for y_k to get sharper bounds $\tilde{\boldsymbol{y}}_k$ with $\mathrm{w}(\tilde{\boldsymbol{y}}_k) = \mathcal{O}\left(\max(\|(\boldsymbol{x} - \check{x}, \boldsymbol{y})\|^2, \|\boldsymbol{y}_n^0\|)\right)$, $1 \le k \le n-1$. Thus we get a small subface of $\boldsymbol{x}_{\underline{n}}$ (or $\boldsymbol{x}_{\overline{n}}$) over which we can either use an interval Newton method to verify the existence and uniqueness of the zero of $\tilde{F}_{\neg u_n}$ or use mean-value extensions to verify that $\tilde{F}_{\neg u_n}$ has no zeros, depending on whether $\boldsymbol{y}_n^0$ is in the middle of $\boldsymbol{y}_n$ or not. So, we end up with searching over a 1-dimensional interval $\boldsymbol{y}_n$. This further reduces the search cost. We can search $\boldsymbol{y}_{\underline{n}}$ or $\boldsymbol{y}_{\overline{n}}$ in a similar way.

The analysis in this section leads to a practical algorithm in the next section.

6. The Algorithm and Its Computational Complexity. In this section, we will present the actual algorithm and give its complexity.

6.1. Algorithm. The algorithm consists of three phases. In the box-setting phase, we set the box $\boldsymbol{z}$. In the elimination phase, we verify that $u_k \ne 0$ on $\boldsymbol{x}_{\underline{k}}$ and $\boldsymbol{x}_{\overline{k}}$, and $v_k \ne 0$ on $\boldsymbol{y}_{\underline{k}}$ and $\boldsymbol{y}_{\overline{k}}$, where $1 \le k \le n-1$. In the search phase, we verify the unique solution of $\tilde{F}_{\neg u_n} = 0$ on $\boldsymbol{x}_{\underline{n}}$ and $\boldsymbol{x}_{\overline{n}}$ with y_n in the interior of $\boldsymbol{y}_n$, and on $\boldsymbol{y}_{\underline{n}}$ and $\boldsymbol{y}_{\overline{n}}$ with x_n in the interior of $\boldsymbol{x}_n$, compute the signs of u_n and the signs of the Jacobi matrices of $\tilde{F}_{\neg u_n}$ at the four solutions of $\tilde{F}_{\neg u_n} = 0$, compute the degree contributions of the 4 faces $\boldsymbol{x}_{\underline{n}}$, $\boldsymbol{x}_{\overline{n}}$, $\boldsymbol{y}_{\underline{n}}$ and $\boldsymbol{y}_{\overline{n}}$ according to the formula in Theorem 5.1 and finally add the contributions to get the degree.

ALGORITHM

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 $1 \le k \le n-1$.
3. Set the widths of $\boldsymbol{x}_n$ and $\boldsymbol{y}_n$ as in (5.1) and (5.2).

Elimination Phase

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

Search Phase

1. Do for $\boldsymbol{x}_{\underline{n}}$ and $\boldsymbol{x}_{\overline{n}}$
   (a) 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 $\mathcal{O}\left(\|(\boldsymbol{x}-\check{x},\boldsymbol{y})\|^2\right)$, $1 \le k \le n-1$.
      ii. If $\tilde{\boldsymbol{x}}_k \cap \boldsymbol{x}_k = \emptyset$, then return the degree contribution of that face as 0.
      iii. Update $\boldsymbol{x}_k$.
   (b) i. Compute the mean-value extension $\boldsymbol{u}_n$ over that face.
      ii. If $\boldsymbol{u}_n < 0$, then return the degree contribution of that face as 0.
   (c) Construct a small subinterval $\boldsymbol{y}_n^0$ of $\boldsymbol{y}_n$ which is centered at 0.
   (d) i. 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 $\mathcal{O}\left(\max(\|(\boldsymbol{x}-\check{x},\boldsymbol{y})\|^2, \|\boldsymbol{y}_n^0\|)\right)$, $1 \le k \le n-1$, 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 $\tilde{\boldsymbol{y}}_k \cap \boldsymbol{y}_k = \emptyset$, then stop and signal failure.
   (e) i. Set up an interval Newton method for $\tilde{F}_{\neg u_n}$ to verify existence and uniqueness of a zero in the subface $\boldsymbol{x}_{\underline{n}}^0$ (or $\boldsymbol{x}_{\overline{n}}^0$.)
      ii. If the zero can not be verified, then stop and signal failure.
   (f) Inflate $\boldsymbol{y}_n^0$ as much as possible provided the existence and uniqueness of the zero of $\tilde{F}_{\neg u_n}$ can be verified over the corresponding subface, and thus get a subinterval $\boldsymbol{y}_{\underline{n}}^1$ of $\boldsymbol{y}_n$.
   (g) In this step, we verify $\tilde{F}_{\neg u_n} = 0$ has no solutions when $y_n \in \boldsymbol{y}_n \setminus \boldsymbol{y}_n^1$. $\boldsymbol{y}_n \setminus \boldsymbol{y}_n^1$ has two separate parts; we denote the lower part by $\boldsymbol{y}_n^l$ and the upper part by $\boldsymbol{y}_n^u$. We only present the processing of the lower part. The upper part can be processed in a similar way.
      i. Do
         A. Use mean-value extensions for $\boldsymbol{v}_k(\boldsymbol{x},\boldsymbol{y}) = 0$ to solve for y_k to get sharper bounds for y_k, $1 \le k \le n-1$, and thus to get a subface of $\boldsymbol{x}_{\underline{n}}$ (or $\boldsymbol{x}_{\overline{n}}$.)
         B. Compute the mean-value extensions $\tilde{\boldsymbol{F}}_{\neg u_n}$ over the subface obtained in the last step.
         C. If $0 \in \tilde{\boldsymbol{F}}_{\neg u_n}$, then bisect $\boldsymbol{y}_n^l$, update the lower part as a new $\boldsymbol{y}_n^l$

and cycle.
If $0 \notin \tilde{\boldsymbol{F}}_{\neg u_n}$, then exit the loop.

ii. Do

A. If $\underline{y}_n^1 \le \overline{y}_n^l$, then exit the loop.

B. $\boldsymbol{y}_n^l \longleftarrow [\overline{y}_n^l, \overline{y}_n^l + \mathrm{w}(\boldsymbol{y}_n^l)]$.

C. Use mean-value extensions for $\boldsymbol{v}_k(\boldsymbol{x},\boldsymbol{y}) = 0$ to solve for y_k to get sharper bounds for y_k, $1 \le k \le n-1$, and thus to get a subface of $\boldsymbol{x}_{\underline{n}}$ (or $\boldsymbol{x}_{\overline{n}}$.)

D. Compute the mean-value extensions $\tilde{\boldsymbol{F}}_{\neg u_n}$ over the subface obtained in the last step.

E. If $0 \notin \tilde{\boldsymbol{F}}_{\neg u_n}$, then cycle.
If $0 \in \tilde{\boldsymbol{F}}_{\neg u_n}$, then $\boldsymbol{y}_n^l \longleftarrow [\underline{y}_n^l, \mathrm{mid}(\boldsymbol{y}_n^l)]$ and cycle.

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

ii. If $\boldsymbol{u}_n < 0$, then return the degree contribution of that face as 0.

(i) i. Compute $\left|\frac{\partial \tilde{F}_{\neg u_n}}{\partial x_1 y_1 \ldots x_{n-1} y_{n-1} y_n}(\boldsymbol{x}_{\underline{n}}^0)\right|$ (or $\left|\frac{\partial \tilde{F}_{\neg u_n}}{\partial x_1 y_1 \ldots x_{n-1} y_{n-1} y_n}(\boldsymbol{x}_{\overline{n}}^0)\right|$).

ii. If $0 \in \left|\frac{\partial \tilde{F}_{\neg u_n}}{\partial x_1 y_1 \ldots x_{n-1} y_{n-1} y_n}(\boldsymbol{x}_{\underline{n}}^0)\right|$ (or $0 \in \left|\frac{\partial \tilde{F}_{\neg u_n}}{\partial x_1 y_1 \ldots x_{n-1} y_{n-1} y_n}(\boldsymbol{x}_{\overline{n}}^0)\right|$), then stop and signal failure.

(j) Use the formula in Theorem 5.1 to compute the degree contribution of that face.

2. Do for $\boldsymbol{y}_{\underline{n}}$ and $\boldsymbol{y}_{\overline{n}}$
   (a) Same as Step 1a except change x_k to y_k, $\tilde{\boldsymbol{x}}_k$ to $\tilde{\boldsymbol{y}}_k$, $\boldsymbol{x}_k$ to $\boldsymbol{y}_k$ and $\boldsymbol{u}_k$ to $\boldsymbol{v}_k$.
   (b) Same as Step 1b.
   (c) Same as Step 1c except change $\boldsymbol{y}_n^0$ to $\boldsymbol{x}_n^0$, $\boldsymbol{y}_n$ to $\boldsymbol{x}_n$ and 0 to $\check{x}_n$.
   (d) Same as Step 1d except change y_k to x_k, $\tilde{\boldsymbol{y}}_k$ to $\tilde{\boldsymbol{x}}_k$, $\boldsymbol{y}_k$ to $\boldsymbol{x}_k$, $\boldsymbol{x}_{\underline{n}}^0$ to $\boldsymbol{y}_{\underline{n}}^0$, $\boldsymbol{x}_{\overline{n}}^0$ to $\boldsymbol{y}_{\overline{n}}^0$, $\boldsymbol{x}_{\underline{n}}$ to $\boldsymbol{y}_{\underline{n}}$ and $\boldsymbol{x}_{\overline{n}}$ to $\boldsymbol{y}_{\overline{n}}$.
   (e) Same as Step 1e except change $\boldsymbol{x}_{\underline{n}}^0$ to $\boldsymbol{y}_{\underline{n}}^0$ and $\boldsymbol{x}_{\overline{n}}^0$ to $\boldsymbol{y}_{\overline{n}}^0$.
   (f) Same as Step 1f except change $\boldsymbol{y}_n^0$ to $\boldsymbol{x}_n^0$, $\boldsymbol{y}_n^1$ to $\boldsymbol{x}_n^1$ and $\boldsymbol{y}_n$ to $\boldsymbol{x}_n$.
   (g) Same as Step 1g except change $\boldsymbol{y}_n \setminus \boldsymbol{y}_n^1$ to $\boldsymbol{x}_n \setminus \boldsymbol{x}_n^1$.
   (h) Same as Step 1h except change $\boldsymbol{x}_{\underline{n}}^0$ to $\boldsymbol{y}_{\underline{n}}^0$ and $\boldsymbol{x}_{\overline{n}}^0$ to $\boldsymbol{y}_{\overline{n}}^0$.
   (i) Same as Step 1i except change
   $\left|\frac{\partial \tilde{F}_{\neg u_n}}{\partial x_1 y_1 \ldots x_{n-1} y_{n-1} y_n}(\boldsymbol{x}_{\underline{n}}^0)\right|$ to $\left|\frac{\partial \tilde{F}_{\neg u_n}}{\partial x_1 y_1 \ldots x_{n-1} y_{n-1} x_n}(\boldsymbol{y}_{\underline{n}}^0)\right|$ and
   $\left|\frac{\partial \tilde{F}_{\neg u_n}}{\partial x_1 y_1 \ldots x_{n-1} y_{n-1} y_n}(\boldsymbol{x}_{\overline{n}}^0)\right|$ to $\left|\frac{\partial \tilde{F}_{\neg u_n}}{\partial x_1 y_1 \ldots x_{n-1} y_{n-1} x_n}(\boldsymbol{y}_{\overline{n}}^0)\right|$.
   (j) Same as Step 1j.
3. Add the degree contributions of the four faces obtained in steps 1 and 2 to get the degree.

END OF ALGORITHM

An Explanation of the Algorithm

1. In the box-setting phase, in Step 2, the width $\mathrm{w}(\boldsymbol{x}_k)$ of $\boldsymbol{x}_k$ depends on the accuracy of the approximate solution $\check{x}$ of the system $F(x) = 0$. $\mathrm{w}(\boldsymbol{x}_k)$ should be much larger than $|\check{x}_k - x_k^*|$. But, at the same time, it should not be too large since the quadratic model needs to be accurate over the box.
2. In the search phase, in Step 1b (or 2b), we check the sign of u_n on that face and discard that face at the earliest possible time if $u_n < 0$ on that face, since we know the degree contribution of that face is 0 according to the formula

in Theorem 5.1. This will save time significantly if it happens that $u_n < 0$ on that face. It did happen for all the test problems. (See §8 for the test results.)

3. In the search phase, in Step 1e (or 2e), we precondition the system $\tilde{F}_{\neg u_n}$ before we use an interval Newton method, so that the method will succeed (see §1.2 and §3). The system $\tilde{F}_{\neg u_n}$ is nonsingular over the subfaces under consideration. In the actual implementation, `LINPACK` routines `DGECO` and `DGESL` were used to compute the preconditioners.
4. In the search phase, in Step 1f (or 2f), we first expand the subinterval $\boldsymbol{y}_n^0$ (or $\boldsymbol{x}_n^0$) by $\epsilon = \frac{1}{2}\mathrm{w}(\boldsymbol{y}_n^0)$ at both ends. If the existence and uniqueness of the zero of $\tilde{F}_{\neg u_n}$ can be verified over the corresponding subface, then we expand the subinterval by 2ϵ at both ends, and then 4ϵ and so on until the existence and uniqueness of the zero can not be established.
5. In the search phase, in Step 1g (or 2g), the underlying idea is that the farther away the interval $\boldsymbol{y}_n^l$ is from the interval $\boldsymbol{y}_n^0$ whose corresponding subface of $\boldsymbol{x}_{\underline{n}}$ (or $\boldsymbol{x}_{\overline{n}}$) contains a unique solution of $\tilde{F}_{\neg u_n} = 0$ and/or the narrower the interval $\boldsymbol{y}_n^l$ is, the more probable it is that we can verify that $\tilde{F}_{\neg u_n} \neq 0$ over the subface of $\boldsymbol{x}_{\underline{n}}$ (or $\boldsymbol{x}_{\overline{n}}$) which corresponds to $\boldsymbol{y}_n^l$.

6.2. Computational Complexity.

Derivation of the Computational Complexity

Box-setting Phase: Step 1 is of order $\mathcal{O}(n^3)$. Step 2 is of order $\mathcal{O}(n)$. Step 3 is of order $\mathcal{O}(n^2)$.
Thus, the order of this phase is $\mathcal{O}(n^3)$.

Elimination Phase: Step 1ai and 1bi are of order $\mathcal{O}(n^2)$. Step 1aii and 1bii are of order $\mathcal{O}(1)$.
Thus, the order of this phase is $\mathcal{O}(n^3)$.

Search Phase: Step 1a and 2a are of order $\mathcal{O}(n^3)$. Step 1b and 2b are of order $\mathcal{O}(n^2)$. Step 1c and 2c are of order $\mathcal{O}(1)$. Step 1d and 2d are of order $\mathcal{O}(n^3)$. Step 1e and 2e are of order $\mathcal{O}(n^3)$. Step 1f and 2f are of order $N_{infl}{*}\mathcal{O}(n^3)$. (See explanation below.) Step 1g and 2g are of order $N_{proc}{*}\mathcal{O}(n^3)$. (See explanation below.) Step 1h and 2h are of order $\mathcal{O}(n^2)$. Step 1i and 2i are of order $\mathcal{O}(n^3)$. Step 1j and 2j are of order $\mathcal{O}(1)$. The last step of this phase is of order $\mathcal{O}(1)$ too.
Thus, the order of this phase is $\mathcal{O}(n^3)$.

The order of the overall algorithm is thus $\mathcal{O}(n^3)$.

Remarks

1. There are two performance measures in the algorithm, N_{infl} and N_{proc}. N_{infl} is the number of inflations the algorithm did in Step 1f or 2f. N_{proc} is the number of subintervals of $\boldsymbol{y}_n \setminus \boldsymbol{y}_n^1$ the algorithm processed in Step 1g or subintervals of $\boldsymbol{x}_n \setminus \boldsymbol{x}_n^1$ the algorithm processed in Step 2g, i.e. number of $\boldsymbol{y}_n^l$'s plus number of $\boldsymbol{y}_n^u$'s in Step 1g or number of $\boldsymbol{x}_n^l$'s plus number of $\boldsymbol{x}_n^u$'s in Step 2g . (See the algorithm in §6.1).
2. The order of the algorithm cannot be improved, since computing preconditioners of the original system and the system $\tilde{F}_{\neg u_n}$ is necessary and computing each preconditioner is of order $\mathcal{O}(n^3)$.

7. Test Problems and Test Environment.

7.1. Test Problems. Before describing the test set, we introduce one more problem. Motivated by [6, Lemma 2.4], we considered systems of the following form.

EXAMPLE 4. *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.

The homotopy h in Example 4 has a simple bifurcation point at $t = t_1$, where the two paths cross obliquely. That is, there are two solutions to $f(x) = 0$ near $x = 0$, for all t near t_1 and on either side of t_1. Furthermore, the quadratic terms in the Taylor expansion for f do not vanish at $t = t_1$.

The test set consists of Example 2, Example 3 with $\epsilon = +10^{-6}$ and -10^{-6}, and Example 4 with $n = 5, 10, 20, 40, 80, 160, 320$. For all the test problems, we used $(0, 0, \ldots, 0)$ as a good approximate solution to the problem $F(x) = 0$. Actually, it's the exact solution in Example 2 and Example 4. $\mathrm{w}(\boldsymbol{x}_k)$ and $\mathrm{w}(\boldsymbol{y}_k)$ were set to 10^{-3} for $1 \le k \le n-1$. $\mathrm{w}(\boldsymbol{x}_n)$ and $\mathrm{w}(\boldsymbol{y}_n)$ were computed automatically by the algorithm. In fact, $\mathrm{w}(\boldsymbol{x}_k)$ and $\mathrm{w}(\boldsymbol{y}_k)$, $1 \le k \le n-1$, can also be computed automatically by the algorithm, depending on the accuracy of the approximate solution. At present, we used the known true solutions to Example 2 and Example 4 and the known approximate solution to Example 3 to test the algorithm and set the widths apparently small but otherwise arbitrary.

For all the problems, the algorithm succeeded and returned a degree of 2.

7.2. Test Environment. The algorithm in §6.1 was programmed in the Fortran 90 environment developed and described in [8, 9]. Similarly, all the functions in the test problems were programmed using the same Fortran 90 system, and internal symbolic representations of the functions were generated prior to execution of the numerical tests. In the actual tests, generic routines then interpreted the internal representations to obtain both floating point and internal values.

The `LINPACK` routines `DGECO` and `DGESL` were used in Step 1 of the box-setting phase, and in Step 1e, 2e, 1f and 2f of the search phase to compute the preconditioners. (See the algorithm and its explanation in §6.1.)

The Sun Fortran 90 compiler version 1.2 was used on a Sparc Ultra model 140 with optimization level 0. Execution times were measured using the routine `DSECND`. All times are given in CPU seconds.

8. Numerical Results.

We present the numerical results in Table 8.1 and some statistical data in Table 8.2.

The column labels of Table 8.1 are as follows.

Problem: names of the problems identified in §7.1.
n**:** number of independent variables.
Success: whether the algorithm was successful.
Degree: yopological degree returned by the algorithm.
CPU Time: CPU time in seconds of the algorithm.

TABLE 8.1
Numerical Results

Problem	n	Success	Degree	CPU Time	Time Ratio
Example 2	2	Yes	2	0.0761	
Example 3 ($\epsilon = +10^{-6}$)	2	Yes	2	0.0511	
Example 3 ($\epsilon = -10^{-6}$)	2	Yes	2	0.0513	
Example 4	5	Yes	2	0.6806	
Example 4	10	Yes	2	3.3403	4.91
Example 4	20	Yes	2	19.440	5.82
Example 4	40	Yes	2	140.34	7.22
Example 4	80	Yes	2	1123.6	8.01
Example 4	160	Yes	2	8891.3	7.91
Example 4	320	Yes	2	65395.5	7.36

Time Ratio: This only applies to Example 4. It's the ratio of two successive CPU times.

The column labels of Table 8.2 are as follows.

Problem: names of the problems identified in §7.1.

n**:** number of independent variables.

N_{infl}**:** number of inflations the algorithm did in Step 1f or 2f for the indicated face $\boldsymbol{x}_{\underline{n}}$, $\boldsymbol{x}_{\overline{n}}$, $\boldsymbol{y}_{\underline{n}}$ or $\boldsymbol{x}_{\overline{y}}$ (see explanation of the computational complexity of the algorithm in §6.2).

N_{proc}**:** number of intervals the algorithm processed in Step 1g or 2g for the indicated face $\boldsymbol{x}_{\underline{n}}$, $\boldsymbol{x}_{\overline{n}}$, $\boldsymbol{y}_{\underline{n}}$ or $\boldsymbol{x}_{\overline{y}}$ (see explanation of the computational complexity of the algorithm in §6.2).

We can see from Table 8.1 that the algorithm was successful on all the problems in the test set. The algorithm is of order $\mathcal{O}\left(n^3\right)$. But the are many $\mathcal{O}\left(n^3\right)$ and $\mathcal{O}\left(n^2\right)$ steps. Some steps have many $\mathcal{O}\left(n^3\right)$ and $\mathcal{O}\left(n^2\right)$ substeps, and some of the substeps still have many $\mathcal{O}\left(n^2\right)$ structures. So, when n was small, those lower order structures had significant influence on the CPU time. But, when n was large, the $\mathcal{O}\left(n^3\right)$ terms dominated. We can see this from the time ratios of Example 4 in Table 8.1.

In Table 8.2, for all the problems, there were two faces of $\boldsymbol{x}_{\underline{n}}$, $\boldsymbol{x}_{\overline{n}}$, $\boldsymbol{y}_{\underline{n}}$ and $\boldsymbol{y}_{\overline{n}}$ for which $N_{infl} = 0$. This is because the algorithm verified that $u_n < 0$ on each of those two faces in Step 1b or 2b, and returned a degree contribution of each of those two faces as 0. So, the algorithm didn't proceed to Step 1f or 2f. For the same reason, $N_{proc} = 0$ for those two faces. For the remaining two faces for which the algorithm did proceed to Step 1f or 2f, N_{infl} is small.

In Step 1g or 2g which immediately follows the inflations, $N_{proc} = 0$ for Example 2 and Example 3. This is because the inflations had covered the whole interval $\boldsymbol{y}_n$. (See the algorithm in §6.1 and the explanation of its computational complexity in §6.2.) More significant is that $N_{proc} = 2$ in Example 4 regardless of small n or large n. This is because only one interval was processed to verify that $\tilde{F}_{\neg u_n} = 0$ has no solutions when $x_n \in \boldsymbol{x}_n^l$ and only one interval was processed to verify that $\tilde{F}_{\neg u_n} = 0$ has no solutions when $x_n \in \boldsymbol{x}_n^u$. (See the algorithm in §6.1 and the explanation of its computational complexity in §6.2.) This means that the algorithm was quite efficient.

TABLE 8.2
Statistical Data

Problem	n	N_{infl}				N_{proc}			
		$\boldsymbol{x}_{\underline{n}}$	$\boldsymbol{x}_{\overline{n}}$	$\boldsymbol{y}_{\underline{n}}$	$\boldsymbol{y}_{\overline{n}}$	$\boldsymbol{x}_{\underline{n}}$	$\boldsymbol{x}_{\overline{n}}$	$\boldsymbol{y}_{\underline{n}}$	$\boldsymbol{y}_{\overline{n}}$
Example 2	2	6	6	0	0	0	0	0	0
Example 3 ($\epsilon = +10^{-6}$)	2	2	2	0	0	0	0	0	0
Example 3 ($\epsilon = -10^{-6}$)	2	2	2	0	0	0	0	0	0
Example 4	5	0	0	5	5	0	0	2	2
Example 4	10	0	0	5	5	0	0	2	2
Example 4	20	0	0	4	4	0	0	2	2
Example 4	40	0	0	4	4	0	0	2	2
Example 4	80	0	0	4	4	0	0	2	2
Example 4	160	0	0	4	4	0	0	2	2
Example 4	320	0	0	3	3	0	0	2	2

9. Conclusions and Future Work. When we tested the algorithm, we took advantage of knowing the true solutions. (See §7.1.) For this reason, we set $\mathrm{w}(\boldsymbol{x}_k)$ and $\mathrm{w}(\boldsymbol{y}_k)$, $1 \le k \le n-1$ somewhat arbitrarily. But we plan to have the algorithm eventually compute these, based on the accuracy of the approximate solution obtained by a floating point algorithm and the accuracy of the quadratic model.

We presented an algorithm which was designed to work for the case that the rank deficiency of the Jacobian matrix at the singular solution is one. But the analysis in §5 and the algorithm in §6.1 can be generalized to general rank deficiency. Also, at present, it is assumed that the second derivatives $\frac{\partial^2 f_n}{\partial x_k \partial x_l}$, $1 \le k \le n, 1 \le l \le n$ don't vanish simultaneously at the singular solution. In fact, the analysis in §5 and the algorithm in §6.1 can be generalized to the general case that the derivatives of f_n of order 1 through r ($r \ge 2$) vanish simultaneously at the singular solution. But computing higher order derivatives may be expensive. Those two generalizations can also be combined, i.e. any rank deficiency and any order of derivatives of f_n that vanish. We will pursue these generalizations in the future.

Another future direction of this study is to apply the algorithms to bifurcation problems and other physical models.

REFERENCES

[1] O. ABERTH, *Computation of topological degree using interval arithmetic, and applications*, Math. Comp., 62 (1994), pp. 171–178.
[2] G. ALEFELD AND J. HERZBERGER, *Introduction to Interval Computations*, Academic Press, New York, 1983.
[3] P. ALEXANDROFF AND H. HOPF, *Topologie*, Chelsea, 1935.
[4] J. CRONIN, *Fixed Points and Topological Degree in Nonlinear Analysis*, American Mathematical Society, Providence, RI, 1964.
[5] E. R. HANSEN, *Global Optimization Using Interval Analysis*, Marcel Dekker, Inc., New York, 1992.
[6] H. JÜRGENS, H.-O. PEITGEN, AND D. SAUPE, *Topological perturbations in the numerical nonlinear eigenvalue and bifurcation problems*, in Analysis and Computation of Fixed Points, S. M. Robinson, ed., New York, 1980, Academic Press, pp. 139–181.

[7] R. B. Kearfott, *Computing the Degree of Maps and a Generalized Method of Bisection*, PhD thesis, University of Utah, Department of Mathematics, 1977.
[8] ———, *A Fortran 90 environment for research and prototyping of enclosure algorithms for nonlinear equations and global optimization*, ACM Trans. Math. Software, 21 (1995), pp. 63–78.
[9] ———, *Rigorous Global Search: Continuous Problems*, Kluwer, Dordrecht, Netherlands, 1996.
[10] A. Neumaier, *Interval Methods for Systems of Equations*, Cambridge University Press, Cambridge, England, 1990.
[11] H. Ratschek and J. Rokne, *New Computer Methods for Global Optimization*, Wiley, New York, 1988.
[12] S. M. Rump, *Verification methods for dense and sparse systems of equations*, in Topics in Validated Computations, J. Herzberger, ed., Amsterdam, 1994, Elsevier Science Publishers, pp. 63–135.
[13] F. Stenger, *Computing the topological degree of a mapping in* $\mathbb{R}^n$, Numer. Math., 25 (1975), pp. 23–38.
[14] M. Stynes, *An Algorithm for the Numerical Calculation of the Degree of a Mapping*, PhD thesis, Oregon State University, Department of Mathematics, Corvallis, Oregon, 1977.