Existence Verification for Singular Zeros of Nonlinear Systems

by

R. Baker Kearfott, rbk@usl.edu
Department of Mathematics, University of Southwestern Louisiana

and

Jianwei Dian, dian@usl.edu
Department of Mathematics, University of Southwestern Louisiana

The General Question

$$\boxed{\begin{minipage}{0.8\linewidth}Given $F : \boldsymbol{x} \to \mathbb{R}^n$ and $\boldsymbol{x} \in \mathbb{IR}^n$, \textit{rigorously} verify:\\ $\bullet$ there exists a unique $x^* \in \boldsymbol{x}$ such that $F(x^*) = 0$,\end{minipage}} \tag{1}$$

Computer arithmetic can be used to verify the assertion in Problem (1), with the aid of interval extensions and *computational fixed point theorems*.

The General Question

Uses

- Producing rigorous bounds on approximate solutions to linear and nonlinear systems (The approximate solutions can be computed with traditional techniques.)
  - in analysis of stability of structures, where one wants to prove that all eigenvalues have negative real parts
  - in robust computational geometry (surface intersection problems, etc.)
- As a tool in branch and bound algorithms in global optimization.
- As a tool in the verification that *all* zeros of a nonlinear system have been found in a region of $\mathbb{R}^n$.

The Nonsingular Case

Traditional Interval Newton Methods

Assumptions (roughly stated):

1. The Jacobi matrix $F'(x^*)$ is nonsingular.
2. x^* is near the center of $\boldsymbol{x}$.
3. The component widths of $\boldsymbol{x}$ are small.
4. $\boldsymbol{N}(F; \boldsymbol{x}, \check{x})$ is the image of $\boldsymbol{x}$ under an appropriate, preconditioned interval Newton method, with $\check{x}$ the center of $\boldsymbol{x}$.

Then:

1. The preconditioned $F'(\boldsymbol{x})$ is approximately the identity matrix.
2. Thus, $\boldsymbol{N}(F; \boldsymbol{x}, \check{x}) \subset \boldsymbol{x}$. This proves that there is a unique solution of $F(x) = 0$ in $\boldsymbol{x}$.

Singularities

When the Jacobi matrix $F'(x^*)$ is singular, computations as above cannot possibly prove existence and uniqueness.

For such systems, the best that a preconditioner can do is reduce the Jacobi matrix to approximately the form

$$\begin{pmatrix} 1 & 0 & \ldots & 0 & \overbrace{* \ldots *}^{n \text{ - rank}} \\ 0 & 1 & 0 \ldots & 0 & * \ldots * \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \ldots & 0 & 1 & * \ldots * \\ 0 & \ldots & 0 & 0 & 0 \ldots 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \ldots & 0 & 0 & 0 \ldots 0 \end{pmatrix}.$$

The Topological Degree

Uses : Verification of at Least One Solution

1. The *topological degree* (to be explained shortly) may be computed over $\boldsymbol{x}$.
2. If the topological degree is non-zero, there is at least one solution of $F(x) = 0$ in $\boldsymbol{x}$.
3. No conclusion can be reached if the topological degree is zero.

The Topological Degree

Uses : Verification of the Exact Multiplicity

1. If $F : \mathbb{C}^n \to \mathbb{C}^n$, then the topological degree of F over $\boldsymbol{x}$ gives the exact number of solutions, counting multiplicities.
2. If $F : \mathbb{R}^n \to \mathbb{R}^n$, and F can be extended analytically into $\mathbb{C}^n$, then computations can verify existence of an exact solution or solutions (with multiplicity computed by the algorithm) within a small region of complex space containing $\boldsymbol{x}$.

The Topological Degree

How is it Computed?

- $\mathrm{d}(F, \boldsymbol{x}, 0)$ depends only on values of F on $\partial\boldsymbol{x}$.
- Define

$$\begin{aligned} F_{\neg k}(x) \;=\; & (f_1(x), \ldots, f_{k-1}(x), \\ & \quad f_{k+1}(x), \ldots, f_n(x)), \end{aligned}$$

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

The Topological Degree

Computational Cost

1. Directly finding all zeros of $F_{\neg k}$ on $\partial \boldsymbol{x}$ can be done in a straightforward branch and bound algorithm. However, that is perhaps too expensive for mere verification purposes.

2. The structure of the preconditioned system can be used to greatly simplify the computations.

3. The widths of the box $\boldsymbol{x}$ constructed about the approximate solution can be chosen so that only several one-dimensional searches need be done to compute $\mathrm{d}(F, \boldsymbol{z}, 0)$, where $F : \mathbb{C}^n \to \mathbb{C}^n$.

Structure of the System

Notation and Assumptions

- For $F : \mathbb{R}^n \to \mathbb{R}^n$, extend F to complex space: $z = x + iy$, $u_k(x, y) = \Re(f_k(z))$ and $v_k(x, y) = \Im(f_k(z))$.
- Define $\tilde{F}(x, y) = (u_1(x, y), v_1(x, y), \ldots, u_n(x, y), v_n(x, y)) : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$.
- Assume $F(\check{x}) \approx 0$.
- Assume F has been preconditioned (say, through an incomplete LU factorization). Also assume $F'(x^*)$ has null space of dimension 1.

Structure of the System

Consequences

$F'(\check{x})$ is approximately the form

$$\begin{pmatrix} 1 & 0 & \ldots & 0 & * \\ 0 & 1 & 0\ldots & 0 & * \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \ldots & 0 & 1 & * \\ 0 & \ldots & 0 & 0 & 0 \end{pmatrix}.$$

So,

$$\begin{aligned} f_k(x) &= (x_k - \check{x}_k) + \frac{\partial f_k}{\partial x_n}(\check{x})(x_n - \check{x}_n) \\ &\quad + \mathcal{O}\left(\|x - \check{x}\|^2\right) \end{aligned}$$

for $1 \le k \le n-1$.

$$\begin{aligned} f_n(x) &= \frac{1}{2}\sum_{k,l=1}^{n} \frac{\partial^2 f_n}{\partial x_k \partial x_l}(\check{x})(x_k - \check{x}_k)(x_l - \check{x}_l) \\ &\quad + \mathcal{O}\left(\|x - \check{x}\|^3\right) \end{aligned}$$

Structure of the System

Consequences (continued)

For $1 \le k \le (n-1)$,

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

So, for $1 \le k \le (n-1)$,

$$\begin{aligned} u_k(x,y) &\approx (x_k - \check{x}_k) + \frac{\partial f_k}{\partial x_n}(\check{x})(x_n - \check{x}_n) \\ v_k(x,y) &\approx y_k + \frac{\partial f_k}{\partial x_n}(\check{x}) y_n \end{aligned}$$

Structure of the System

Consequences (continued)

The following things are useful in the search phase (to be explained shortly) of the topological degree approach.

1. If x_n is known precisely, formally solving $\boldsymbol{u}_k(\boldsymbol{x}, \boldsymbol{y}) = 0$ for x_k gives $\boldsymbol{x}_k$ with $\mathrm{w}(\boldsymbol{x}_k) = \mathcal{O}\left(\|(\boldsymbol{x} - \check{x}, \boldsymbol{y})\|^2\right)$, $1 \le k \le n-1$.

2. If y_n is known precisely, formally solving $\boldsymbol{v}_k(\boldsymbol{x}, \boldsymbol{y}) = 0$ for y_k gives $\boldsymbol{y}_k$ with $\mathrm{w}(\boldsymbol{y}_k) = \mathcal{O}\left(\|(\boldsymbol{x} - \check{x}, \boldsymbol{y})\|^2\right)$, $1 \le k \le n-1$.

Computation of $\mathrm{d}(\tilde{F}, \boldsymbol{z}, 0)$

1. Define
   $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = ([\underline{x}_1, \overline{x}_1], \ldots, [\underline{x}_n, \overline{x}_n])$
   and

   $\boldsymbol{y} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = ([\underline{y}_1, \overline{y}_1], \ldots, [\underline{y}_n, \overline{y}_n])$.
   and
   $$\boldsymbol{z} = (\boldsymbol{x}_1, \boldsymbol{y}_1, \ldots, \boldsymbol{x}_n, \boldsymbol{y}_n)$$
2. The center of $\boldsymbol{x}_k$ is $\check{x}_k$.

   The center of $\boldsymbol{y}_k$ is 0.
3. Define $\boldsymbol{x}_{\underline{k}}$ as $(\boldsymbol{x}, \boldsymbol{y})$ with $[\underline{x}_k, \overline{x}_k]$ replaced by $\underline{x}_k$, and define $\boldsymbol{x}_{\overline{k}}$ as $(\boldsymbol{x}, \boldsymbol{y})$ with $[\underline{x}_k, \overline{x}_k]$ replaced by $\overline{x}_k$. Similarly define $\boldsymbol{y}_{\underline{k}}$ and $\boldsymbol{y}_{\overline{k}}$.
4. Consider
   $$\begin{aligned} \tilde{F}_{\neg u_n}(x, y) &= (u_1(x, y), v_1(x, y), \ldots, \\ &\quad u_{n-1}(x, y), v_{n-1}(x, y), v_n(x, y)) \end{aligned}$$
   on the boundary of $(\boldsymbol{x}, \boldsymbol{y})$.

Computation of $\mathrm{d}(\tilde{F}, \boldsymbol{z}, 0)$

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

$$\Longrightarrow \; u_k(x, y) \approx (\underline{x}_k - \check{x}_k) + \frac{\partial f_k}{\partial x_n}(\check{x})(x_n - \check{x}_n) = 0$$

$$\Longrightarrow \; \frac{|\underline{x}_k - \check{x}_k|}{|\partial f_k / \partial x_n(\check{x})|} = |x_n - \check{x}_n|$$

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

Similarly,
on $\boldsymbol{x}_{\overline{k}}$,

$$\tilde{F}_{\neg u_n}(x, y) = 0 \Longrightarrow \frac{\mathrm{w}(\boldsymbol{x}_k)}{|\partial f_k / \partial x_n(\check{x})|} \le \mathrm{w}(\boldsymbol{x}_n)$$

Computation of $\mathrm{d}(\tilde{F}, \boldsymbol{z}, 0)$

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

$$\Longrightarrow \; u_k(x, y) \approx \underline{y}_k + \frac{\partial f_k}{\partial x_n}(\check{x}) y_n = 0$$
$$\Longrightarrow \; \frac{|\underline{y}_k|}{|\partial f_k / \partial x_n(\check{x})|} = |y_n|$$
$$\Longrightarrow \; \frac{\mathrm{w}(\boldsymbol{y}_k)}{|\partial f_k / \partial x_n(\check{x})|} \le \mathrm{w}(\boldsymbol{y}_n)$$

Similarly,
on $\boldsymbol{y}_{\overline{k}}$,

$$\tilde{F}_{\neg u_n}(x, y) = 0 \Longrightarrow \frac{\mathrm{w}(\boldsymbol{y}_k)}{|\partial f_k / \partial x_n(\check{x})|} \le \mathrm{w}(\boldsymbol{y}_n)$$

Computation of $\mathrm{d}(\tilde{F}, \boldsymbol{z}, 0)$

1. Thus, if $\boldsymbol{x}_n$ is chosen so that
$$\mathrm{w}(\boldsymbol{x}_n) \le \frac{1}{2} \min_{1 \le k \le n-1} \left\{ \frac{\mathrm{w}(\boldsymbol{x}_k)}{|\partial f_k / \partial x_n(\check{x})|} \right\},$$
then it is unlikely that $u_k(x, y) = 0$ on either $\boldsymbol{x}_{\underline{k}}$ or $\boldsymbol{x}_{\overline{k}}$.
2. Similarly, if $\boldsymbol{y}_n$ is chosen so that
$$\mathrm{w}(\boldsymbol{y}_n) \le \frac{1}{2} \min_{1 \le k \le n-1} \left\{ \frac{\mathrm{w}(\boldsymbol{y}_k)}{|\partial f_k / \partial x_n(\check{x})|} \right\},$$
then it is unlikely that $v_k(x, y) = 0$ on either $\boldsymbol{y}_{\underline{k}}$ or $\boldsymbol{y}_{\overline{k}}$.
3. So, we can eliminate $4n - 4$ of the $4n$ faces of the boundary of $(\boldsymbol{x}, \boldsymbol{y})$, since we have arranged to verify $\tilde{F}_{\neg u_n}(x, y) \ne 0$ on each of these faces. Thus, we only need to search the four faces $\boldsymbol{x}_{\underline{n}}, \boldsymbol{x}_{\overline{n}}, \boldsymbol{y}_{\underline{n}}$ and $\boldsymbol{y}_{\overline{n}}$, regardless of how large n is.

Some Hints

If

1. the approximations

$$f_k(x) \approx (x_k - \check{x}_k) + \frac{\partial f_k}{\partial x_n}(\check{x})(x_n - \check{x}_n)$$

   for $1 \le k \le n-1$.

$$f_n(x) \approx \frac{1}{2}\sum_{k,l=1}^{n} \frac{\partial^2 f_n}{\partial x_k \partial x_l}(\check{x})(x_k - \check{x}_k)(x_l - \check{x}_l)$$

   are exact; and

2. $$\sum_{k,l=1}^{n} \frac{\partial^2 f_n}{\partial x_k \partial x_l}(\check{x})\alpha_k\alpha_l \neq 0,$$

   where $\alpha_k = \partial f_k/\partial x_n(\check{x})$ for $1 \le k \le n-1$, and $\alpha_n = -1$,

then, $\mathrm{d}(\tilde{F}, \boldsymbol{z}, 0) = 2$.

Some Hints

Also, under the above assumptions,

- $\tilde{F}_{\neg u_n}(x, y) = 0$ has no solutions on each of the $4n - 4$ faces $\boldsymbol{x}_{\underline{k}}, \boldsymbol{x}_{\overline{k}}, \boldsymbol{y}_{\underline{k}}$ and $\boldsymbol{y}_{\overline{k}}, 1 \leq k \leq n - 1$.
- $\tilde{F}_{\neg u_n}(x, y) = 0$ has a unique solution on each of the 4 faces $\boldsymbol{x}_{\underline{n}}, \boldsymbol{x}_{\overline{n}}, \boldsymbol{y}_{\underline{n}}$ and $\boldsymbol{y}_{\overline{n}}$.

We also know where the solution is, for example, on $\boldsymbol{x}_{\underline{k}}$ the solution is $(\check{x}_1, 0, \check{x}_2, 0, \ldots, \check{x}_{n-1}, 0, \underline{x}_n, 0)$. This knowledge helps in the search phase which handles the faces $\boldsymbol{x}_{\underline{n}}, \boldsymbol{x}_{\overline{n}}, \boldsymbol{y}_{\underline{n}}$ and $\boldsymbol{y}_{\overline{n}}$ when the approximations are not exact.

The Actual Algorithm

Construction of the Box $\boldsymbol{z}$

1. For $1 \le k \le n-1$, $\boldsymbol{x}_k$ is chosen to be centered at $\check{x}_k$. We need to take into consideration the accuracy of the approximate solver which finds the approximate solution of $F(x) = 0$.
2. For $1 \le k \le n-1$, $\boldsymbol{y}_k$ is chosen to be centered at 0. The width of $\boldsymbol{y}_k$ is chosen to be small, with some freedom.
3. $\boldsymbol{x}_n$ is chosen so that
$$\mathrm{w}(\boldsymbol{x}_n) \le \frac{1}{2} \min_{1 \le k \le n-1} \left\{ \frac{\mathrm{w}(\boldsymbol{x}_k)}{|\partial f_k / \partial x_n(\check{x})|} \right\}.$$
4. $\boldsymbol{y}_n$ is chosen so that
$$\mathrm{w}(\boldsymbol{y}_n) \le \frac{1}{2} \min_{1 \le k \le n-1} \left\{ \frac{\mathrm{w}(\boldsymbol{y}_k)}{|\partial f_k / \partial x_n(\check{x})|} \right\}.$$

The Actual Algorithm

Elimination Phase

For $k = 1$ to $n - 1$

1. Use mean-value interval evaluations of $u_k(x, y)$ over $\boldsymbol{x}_{\underline{k}}$ and $\boldsymbol{x}_{\overline{k}}$ to show that $u_k(x, y) \neq 0$ on these faces of $\boldsymbol{z}$. This implies $\tilde{F}_{\neg u_n}(x, y) = 0$ has no solutions on $\boldsymbol{x}_{\underline{k}}$ and $\boldsymbol{x}_{\overline{k}}$
2. Use second-order interval evaluations of $v_k(x, y)$ over $\boldsymbol{y}_{\underline{k}}$ and $\boldsymbol{y}_{\overline{k}}$ to show that $v_k(x, y) \neq 0$ on these faces of $\boldsymbol{z}$. This implies $\tilde{F}_{\neg u_n}(x, y) = 0$ has no solutions on $\boldsymbol{y}_{\underline{k}}$ and $\boldsymbol{y}_{\overline{k}}$

The Actual Algorithm

Search Phase

1. On $\boldsymbol{x}_{\underline{n}}$ and $\boldsymbol{x}_{\overline{n}}$:
   (a) Narrow $\boldsymbol{x}_k$: use mean-value extensions $\boldsymbol{u}_k(\boldsymbol{x}, \boldsymbol{y}) = 0$ to solve for $\boldsymbol{x}_k$ with width $\mathcal{O}\left(\|(\boldsymbol{x} - \check{x}, \boldsymbol{y})\|^2\right)$, $1 \leq k \leq n-1$.

The Actual Algorithm

Search Phase (continued)

(b) Search $\boldsymbol{y}_n$:

i.A. A small subinterval $\boldsymbol{y}_n^0$ of $\boldsymbol{y}_n$ is chosen centered at 0.

B. The mean-value extensions for $\boldsymbol{v}_k(\boldsymbol{x}, \boldsymbol{y}) = 0$ are used to solve for $\boldsymbol{y}_k$ with width $\mathcal{O}\left(\max(\|(\boldsymbol{x} - \check{x}, \boldsymbol{y})\|^2, \|\boldsymbol{y}_n^0\|)\right)$, $1 \le k \le n-1$.

C. An interval Newton method can be set up for $\tilde{F}_{\neg u_n}$ to verify existence and uniqueness of a zero in $\boldsymbol{y}_n^0$.

ii. $\boldsymbol{y}_n^0$ is inflated as much as possible provided the existence and uniqueness of the zero can be verified.

iii. Verify $\tilde{F}_{\neg u_n} = 0$ has no solutions in the rest of $\boldsymbol{y}_n$

The Actual Algorithm

Search Phase (continued)

2. On $\boldsymbol{y}_{\underline{n}}$ and $\boldsymbol{y}_{\overline{n}}$:

   (a) Narrow $\boldsymbol{y}_k$: use mean-value extensions $\boldsymbol{v}_k(\boldsymbol{x}, \boldsymbol{y}) = 0$ to solve for $\boldsymbol{y}_k$ with width $\mathcal{O}\left(\|(\boldsymbol{x} - \check{x}, \boldsymbol{y})\|^2\right)$, $1 \le k \le n-1$.

The Actual Algorithm

Search Phase (continued)

(b) Search $\boldsymbol{x}_n$:

i.A. A small subinterval $\boldsymbol{x}_n^0$ of $\boldsymbol{x}_n$ is chosen centered at $\check{x}_n$.

B. The mean-value extensions for $\boldsymbol{u}_k(\boldsymbol{x}, \boldsymbol{y}) = 0$ are used to solve for $\boldsymbol{x}_k$ with width $\mathcal{O}\left(\max(\|(\boldsymbol{x} - \check{x}, \boldsymbol{y})\|^2, \|\boldsymbol{x}_n^0\|)\right)$, $1 \le k \le n - 1$.

C. An interval Newton method can be set up for $\tilde{F}_{\neg u_n}$ to verify existence and uniqueness of a zero in $\boldsymbol{x}_n^0$.

ii. $\boldsymbol{x}_n^0$ is inflated as much as possible provided the existence and uniqueness of the zero can be verified.

iii. Verify $\tilde{F}_{\neg u_n} = 0$ has no solutions in the rest of $\boldsymbol{y}_n$

The Actual Algorithm

Search Phase (continued)

3. For each solution to $\tilde{F}_{\neg u_n} = 0$ found in Steps 1b and 2b, compute the sign of u_n. Eliminate the solutions with negative u_n.

4. For each solution to $\tilde{F}_{\neg u_n} = 0$ left in Steps 3, compute the sign of the determinant to get an orientation. Sum to get the degree.

Preliminary Experimental Results

1. Elimination phase: for all experiments that had been done, it was always successful to eliminate the $4n-4$ faces $\boldsymbol{x}_{\underline{k}}, \boldsymbol{x}_{\overline{k}}, \boldsymbol{y}_{\underline{k}}$ and $\boldsymbol{y}_{\overline{k}}, 1 \le k \le n-1$.
2. Search phase: for some of the experiments, it successfully searched the four faces $\boldsymbol{x}_{\underline{n}}, \boldsymbol{x}_{\overline{n}}, \boldsymbol{y}_{\underline{n}}$ and $\boldsymbol{y}_{\overline{n}}$, and verified the degree was 2. For others, different problems occurred, indicating the need to adjust some parameters.