BIT 21 (1981), 203–211

BOUNDING SOLUTIONS OF SYSTEMS OF EQUATIONS USING INTERVAL ANALYSIS

ELDON HANSEN and SAUMYENDRA SENGUPTA

Abstract.

We introduce some variations of the interval Newton method for bounding solutions to a set of n nonlinear equations. It is pointed out that previous implementations of Krawczyk's method are very inefficient and an improved version is given. A superior type of Newton method is introduced.

1. Introduction.

Given a vector $f=(f_1,\ldots f_n)^T$ of n real, nonlinear functions of a real vector $x=(x_1,\ldots,x_n)^T$, we consider some Newton-like methods for finding and bounding solutions to

$$f(x) = 0\,. \tag{1.1}$$

These methods use interval analysis to obtain error bounds on the solutions.

One such method due to Krawczyk [9] is discussed in detail. It is shown that previous implementations of this method are very inefficient, and an improved version is given. We shall also introduce a new method which is faster than the Krawczyk method even in the improved version we develop.

We assume the reader is familiar with interval analysis. Any relevant concepts not defined here are discussed in [10].

2. A survey of interval Newton methods.

R. E. Moore [10] first introduced an interval analytic method for finding and bounding a solution y of (1.1). Let $\dot{x}$ be an approximate solution. Using Taylor's theorem and expanding $f(y)$ about x, we obtain

$$f(x)+J(\xi)(y-x) = f(y) = 0 \tag{2.1}$$

where $J(\xi)$ is the Jacobian evaluated at a point ξ. Moore observed that if X is an interval vector containing both x and y, then $\xi \in X$. Hence, he replaced $J(\xi)$ in (2.1) by the interval matrix $J(X)$. The set (say Z) of points z satisfying

$$f(x)+J(X)(z-x) = 0 \tag{2.2}$$

contains y. For simplicity, we assume x is the midpoint of X.

Received October 20, 1980.

The size of the set Z depends on the widths of the interval elements of $J(X)$. Hansen [2], [4] showed that by writing the Taylor expansion of $f(x)$ in an appropriate way, the widths of these interval elements could be substantially reduced. This reduces the size of Z and speeds convergence of the Newton methods.

We would like to know the set Z. However, as pointed out by Hansen [3], this set can be difficult to represent. Instead, interval Newton methods find a box (interval vector) containing Z. Geometrically it is a parallelepiped with sides parallel to the coordinate axes and thus easy to describe.

The first interval Newton method introduced by Moore [10] involved finding a kind of inverse of the interval matrix $J(X)$. That is, it required an interval matrix M containing J^{-1} for every real (i.e., non-interval) matrix $J \in J(X)$ and assumes that each such J is nonsingular. For the new method described below, this nonsingularity is not necessary.

Hansen [2] pointed out that it is not necessary to find an interval inverse in order to solve the linear equations in an interval Newton method. Instead, a Gaussian elimination procedure could be used as in the real counterpart. Define J_c to be the center of $J(X)$, i.e. each element of J_c is the midpoint of the corresponding interval element of $J(X)$. Hansen and Smith [6] showed that a set of linear algebraic equations such as (2.2) with interval coefficients is best solved by first premultiplying (2.2) by an approximate inverse of J_c. Let B be this approximation. We thus rewrite (2.2) as

$$Bf(x)+BJ(X)(z-x) = 0 . \tag{2.3}$$

The products $Bf(x)$ and $BJ(X)$ are computed in interval arithmetic to bound rounding errors.

Krawczyk [9] introduced a variation of the interval Newton method which avoided the Gaussian elimination of an interval matrix by not attempting to obtain a sharp solution of (2.3). Thus he computes the box

$$K(X) = x-Bf(x)+[I-BJ(X)](X-x) . \tag{2.4}$$

This box contains every solution of (2.2). In effect, this solves (approximately) the ith equation of (2.3) for a bound $K_i(X)$ on the ith component of the solution set Z.

This is a kind of simultaneous iteration. We shall see that a corresponding successive iteration can be used which greatly improves convergence. Successive iteration is also used in our improved method.

3. The Krawczyk method.

As pointed out in Section 2, the Krawczyk method involves computation of the box $K(X)$ given by (2.4). If a solution y of (1.1) is contained in a box X, then it is also contained in $K(X)$ (see [9]). Since $K(X)$ may not be contained in X, we use the iteration

$$X^{(i+1)} = X^{(i)} \cap K(X^{(i)}) \quad (i=0,1,2,\ldots)$$

where the initial box $X^{(0)}$ is given.

As described by Krawczyk and others (e.g., see [11], and [12]) who have used this method, it is a method of simultaneous iteration. However, convergence is improved if it is used in a successive iteration mode. Thus, a component K_i $(i=1,\ldots,n)$ of $K(X)$ should be computed as

$$K_i = x_i - g_i + \sum_{j=1}^{i-1} R_{ij}(K_j' - x_j) + \sum_{j=i}^{n} R_{ij}(X_j - x_j) \tag{3.1}$$

where

$$g = Bf(x),\ R = I - BJ(X),\ K_j' = K_j \cap X_j .$$

Note that we find the intersection K_j' of K_j and X_j as soon as K_j is found using the best currently available data.

It might appear as if another modification of the Krawczyk method could be useful. As described by previous authors, the matrix

$$R = I - BJ(X)$$

is computed explicitly. This involves computing the matrix product $BJ(X)$. But we need only $R(X-x)$ which could be obtained as

$$R(X-x) = X - x - BJ(X)(X-x)$$

where $J(X)(X-x)$ is computed first. This procedure would involve multiplying a vector by a matrix (twice) but not a matrix by a matrix and hence involving fewer operations.

Unfortunately, this more efficient calculation tends to increase the number of iterations necessary to obtain a solution of prescribed accuracy. We can see this as follows.

Since x_j $(j=1,\ldots,n)$ is the midpoint of X_j, we can write

$$X_j - x_j = \tfrac{1}{2} w_j [-1,1]$$

where w_j is the width of X_j. Therefore

$$\begin{aligned}[J(X)(X-x)]_i &= \sum_{j=1}^{n} [J(X)]_{ij}(X-x)_j \\ &= \tfrac{1}{2}[-1,1] \sum_{j=1}^{n} |[J(X)]_{ij}| w_j .\end{aligned}$$

Here we have used the absolute value of an interval which is defined as follows.

If $V=[v_1,v_2]$, then $|V| = \max(|v_1|,|v_2|)$. When we multiply $J(X)(X-x)$ by B, we find the kth element of the result

$$[BJ(X)(X-x)]_k = \tfrac{1}{2}[-1,1] \sum_{i=1}^{n} |b_{ki}| \sum_{j=1}^{n} |[J(X)]_{ij}| w_j . \tag{3.2}$$

If, instead, we compute $BJ(X)$ first, we obtain

$$[BJ(X)(X-x)]_k = \tfrac{1}{2}[-1,1] \sum_{j=1}^{n} \left| \sum_{i=1}^{n} b_{ki}[J(X)]_{ij} \right| w_j .$$

This result is obviously a narrower interval in general than that given by (3.2).

In the problems of low dimension on which we have tried these options, it was more efficient overall to compute the matrix product $BJ(X)$ explicitly. However, this may not be the case for large problems.

4. A more efficient method.

In each iteration of the Krawczyk method the box $K(X)$ is computed (see (2.4)). This box bounds the solution set of the linearized equation (2.3). However, it is not the smallest such box. We now present a method which also bounds the solution to (2.3). However, the box which it obtains is generally smaller than $K(X)$. Since each iteration of our method tends to produce a greater reduction of the current box than does Krawczyk's method, fewer steps are required for numerical convergence.

Denote $g = Bf(x)$ and $P = BJ(X)$ so that equation (2.3) becomes

$$g + P(z-x) = 0 . \tag{4.1}$$

Hopefully, P closely approximates the identity matrix. Thus, we simply solve the ith equation for the ith variable and replace the others by bounding intervals. As in our improved version of the Krawczyk method, we use successive iteration. In effect, the Krawczyk method adds the term $(P_{ii}-1)\ (X_i - x_i)$ before solving which widens the resulting interval.

Write the interval matrix P as

$$P = L + D + U \tag{4.2}$$

where the matrices L, D, and U are lower triangular, diagonal, and upper triangular, respectively. Our approximate solution X' is obtained as

$$Y = x - D^{-1}[g + L(X'-x) + U(X-x)], \quad X' = Y \cap X . \tag{4.3}$$

As each new component Y_i $(i=1,\ldots,n)$ is obtained, it is immediately intersected with X_i so that the newest result $X_i' = Y_i \cap X_i$ can be used in finding $Y_{i+1},\ldots, Y_n$. Thus we compute componentwise, for $i=1,\ldots n$,

$$Y_i = x_i - (D_{ii})^{-1}\left[g_i + \sum_{j=1}^{i-1} P_{ij}(X_j' - x_j) + \sum_{j=i+1}^{n} P_{ij}(X_j - x_j) \right], \tag{4.4a}$$

$$X_i' = Y_i \cap X_i . \tag{4.4b}$$

Note that even though P is supposed to approximate the identity matrix, it is possible for an interval D_{ii} to contain zero for one or more values of i. This creates

no real difficulty and we simply use extended interval arithmetic to compute Y_i from (4.4a). The intersection (4.4b) then produces a finite result. We give the details below.

In [5], Hansen derived a globally convergent, one-dimensional interval Newton method using extended interval arithmetic. At the time of publication of that paper, he was unaware that extended interval arithmetic had already been used in the interval Newton method by Alefeld [1].

We now consider the computational details when D_{ii} contains zero. We make use of extended interval arithmetic as introduced by Hanson [7] and by Kahan [8].

Let $A=[a_1, a_2]$ and $B=[b_1, b_2]$ be finite intervals. If B does not contain zero, we can divide A by B using ordinary interval arithmetic. The resulting interval is the set

$$\{a/b: a \in A, b \in B\} .$$

We want this same set in the extended case. When $0 \in B$ we have the following cases:

$$A/B = \begin{cases} [a_2/b_1, +\infty] \text{ if } a_2 \leqq 0 \text{ and } b_2 = 0 , \\ [-\infty, a_2/b_2] \cup [a_2/b_1, +\infty] \quad \text{if } a_2 < 0,\ b_1 < 0, \text{ and } b_2 > 0 , \\ [-\infty, a_2/b_2] \text{ if } a_2 \leqq 0 \text{ and } b_1 = 0 , \\ [-\infty, a_1/b_1] \text{ if } a_1 \geqq 0 \text{ and } b_2 = 0 , \\ [-\infty, a_1/b_1] \cup [a_1/b_2, +\infty] \quad \text{if } a_1 > 0,\ b_1 < 0, \text{ and } b_2 > 0 , \\ [a_1/b_2, +\infty] \text{ if } a_1 \geqq 0 \text{ and } b_1 = 0 , \\ [-\infty, +\infty] \text{ if } a_1 < 0 \text{ and } a_2 > 0 . \end{cases} \tag{4.5}$$

The computation of Y_i from (4.4a) can be completed using (4.5) and the following rules of extended interval arithmetic:

$$x_i - [c_i, +\infty] = [-\infty, x_i - c_i] ,$$

$$x_i - [-\infty, d_i] = [x_i - d_i, \infty] ,$$

$$x_i - [-\infty, \infty] = [-\infty, \infty] ,$$

$$x_i - [-\infty, d_i] \cup [c_i, +\infty] = [-\infty, x_i - c_i] \cup [x_i - d_i, \infty] .$$

5. Convergence.

The iteration defined by (4.3) is

$$Y^{(k)} = x^{(k)} - (D^{(k)})^{-1}[g^{(k)} + L^{(k)}(X^{(k+1)} - x^{(k)}) + U^{(k)}(X^{(k)} - x^{(k)})] \tag{5.1}$$

$$X^{(k+1)} = Y^{(k)} \cap X^{(k)} \qquad (k = 0, 1, 2, \ldots) .$$

In this section, we prove that this algorithm converges under appropriate conditions. To this end, denote

$$\alpha_k = \max w(X_i^{(k)}),\ \delta_k = \max |P_{ii}^{(k)} - 1|\ ,\quad \varrho_k = \max \sum_{\substack{j=1 \\ j \neq i}}^{n} |P_{ij}^{(k)}|$$

where each maximum is for $i = 1, \ldots, n$ and $w(X_i^{(k)})$ denotes the width of $X_i^{(k)}$.

THEOREM. *If f has a single simple zero x^* in $X^{(0)}$ and if for some $k = 0, 1, 2, \ldots$ the conditions $\delta_k < 2/3^{\frac{1}{2}} - 1$ and $\varrho_k < (1-\delta)/2$ hold, then $X^{(k)} \to x^*$.*

If $w(X^{(k)})$ is sufficiently small, the conditions on δ_k and ϱ_k will hold. In fact, if $X^{(k)}$ were a single point, we would have $\delta_k = \varrho_k = 0$. Since $w(J_{ij}(X)) = O(w(X))$ (see [10]), δ_k and ϱ_k are arbitrarily small for $w(X^{(k)})$ sufficiently small.

From the derivation of our algorithm, it follows that $x^* \in X^{(k)}$ for all $k = 0, 1, 2, \ldots$ since $x^* \in X^{(0)}$. We use this fact in the following proof of the theorem.

Note that $X^{(k+1)} \subset X^{(k)}$ so that if we replace $X^{(k+1)}$ by $X^{(k)}$ in the right member of (5.1), the result contains $Y^{(k)}$. Since $x^{(k)}$ is the midpoint of $X^{(k)}$,

$$X_j^{(k)} - x_j^{(k)} = \tfrac{1}{2} w(X_j^{(k)})[-1, 1] \subset \tfrac{1}{2}\alpha_k[-1, 1]\ .$$

Therefore from (5.1)

$$(5.2)\quad Y_i^{(k)} \subset X_i^{(k)} - \left\{ g^{(k)} + \tfrac{1}{2}\alpha_k \sum_{\substack{j=1 \\ j \neq 1}}^{n} |P_{ij}^{(k)}|[-1, 1] \right\} \Big/ P_{ii}^{(k)}$$

$$\subset x_i^{(k)} - \{ g^{(k)} + \tfrac{1}{2}\alpha_k \varrho_k [-1, 1] \} / [1 - \delta, 1 + \delta]$$

for $i = 1, \ldots, n$, from which $w(Y_i^{(k)}) < |g^{(k)}| 2\delta_k/(1-\delta_k^2) + \alpha_k \varrho_k/(1-\delta)$.

If we expand $f(x^{(k)})$ about x^*, then in the same way we obtained equation (2.3), we find $g^{(k)} \in P^{(k)}(x^{(k)} - x^*)$. Replacing the point x^* by $X^{(k)}$ which contains it, we can proceed as before and obtain

$$g^{(k)} \in \tfrac{1}{2}\alpha_k(1 + \delta_k + \varrho_k)\ [-1, 1]\ .$$

Using this result and the hypotheses of the theorem, we find from (5.2) that

$$w(Y_i^{(k)}) < \alpha_k\ .$$

That is, the widest interval component of $Y^{(k)}$ (and hence of $X^{(k+1)}$) is strictly less than that of $X^{(k)}$.

Because of the inclusion monotonicity of interval arithmetic, $P^{(k+1)}$ will be contained in $P^{(k)}$ for all $k = 0, 1, 2, \ldots$ This implies that δ_k and ϱ_k are monotonically decreasing with k. Therefore if the hypotheses of the theorem are satisfied for any specific k, they are satisfied for all larger values of k. Hence $X^{(k+1)}$ is strictly contained in $X^{(k)}$ for all sufficiently large k. This completes the proof. ∎

6. A simplification.

When $0 \in D_{ii}$, it is possible for X_i' to be composed of two disjoint intervals. If this were the case for all $i=1,\ldots,n$, the box X' would be composed of 2^n disjoint boxes. We wish to prevent the number of boxes from getting large in this way. Also, if X_j' is composed of two intervals, we do not wish to have to use each separately to find X_i' for $i>j$. We now consider how to simplify the computations and reduce the number of boxes generated.

If X_j' is composed of two intervals, we do not use X_j' in (4.4a) when computing X_i'. Instead, we simply use the single interval X_j.

If X_i' is composed of two intervals for more than one value of i, we replace X_i' by X_i for only one value of i. Thus X' will be composed of only two boxes. We choose the particular value of i by retaining the intervals with the largest gap.

Let I denote the set of values of i for which X_i' is two disjoint intervals. For $i \in I$, denote

$$X_i' = [a_i, b_i] \cup [c_i, d_i] .$$

The gap between the disjoint intervals $[a_i, b_i]$ and $[c_i, d_i]$ is of length $c_i - b_i$. (We are free to assume that $b_i < c_i$.) Let j be the index of the largest gap so that

$$c_j - b_j \geqq c_i - b_i$$

for all $i \in I$. Then we use X_j' but we use X_i rather than X_i' for the other values of $i \in I$. Thus the new set X' will be composed of two boxes; one whose jth component is $[a_j, b_j]$ and one whose jth component is $[c_j, d_j]$. The components of the two boxes is the same for all $i \neq j$.

7. Multiple boxes.

When applying an interval Newton method, we are usually interested in finding the solution(s) of (1.1) in a given box $X^{(0)}$. It can happen that little or no progress is made in reducing the size of the current box during a step of the method. In this case, it is common practice to divide the box in half (say) and apply the algorithm to each sub-box separately. Thus, our method introduces no new aspect as far as the multiplicity of boxes is concerned. A novelty occurs in that if distinct solutions occur in $X^{(0)}$, our method tends to split a box automatically into sub-boxes with each solution in a separate box. Using extended interval arithmetic, it is much less frequently necessary to split a box simply because of lack of progress.

8. Experimental results.

We have compared our method to the improved version of the Krawczyk method described in Section 3. The computational effort to perform a step of each method is about the same. Hence only the number of steps is reported and no timing is given. In the experiments, our method has always required fewer iterations than Krawczyk's to achieve numerical convergence. Our experience is restricted to problems of low dimension but we believe our method is superior for higher dimensions also.

We have also compared these methods with the methods suggested by Hansen [2] in which equation (2.3) is solved by Gaussian elimination. The latter method was not competitive in the few comparisons we made. Hence no numerical results are given for it.

In Table 1, we summarize some representative numerical results. In each case, the iteration was terminated when the width of each final box bounding a solution was less than 10^{-6}. The width, w, of a box with components $X_i=[a_i, b_i]$ $(i=1,\ldots,n)$ is defined to be

$$w = \max_{1\leq i\leq n} (b_i - a_i) .$$

When the initial box contained more than one solution, each was found to this accuracy.

Various functions were used in our experiments. The ones used to obtain the results in Table 1, were as follows.

The first function, $f_1(x)$, was the gradient of the so-called three hump camel function which is a two-dimensional function that has been frequently used in testing optimization programs. Its gradient is

$$f_1(x) = \begin{bmatrix} 6x_1^5 - 25.2x_1^3 + 24x_1 - 6x_2 \\ 12x_2 - 6x_1 \end{bmatrix}.$$

The second and third functions were also two-dimensional. The components are the real and imaginary parts of the polynomials

$$(z^2-4i)(z-1.7) = 0 \quad \text{and} \quad (z^2-4i)^2 = 0$$

so that the functions were

$$f_2(x) = \begin{bmatrix} x_1^3 - 3x_1x_2^2 - 1.7x_1^2 + 1.7x_2^2 + 4x_2 \\ x_2^3 - 3x_1^2x_2 + 3.4x_1x_2 + 4x_1 - 6.8 \end{bmatrix} \quad \text{and}$$

$$f_3(x) = \begin{bmatrix} x_1^4 - 6x_1^2x_2^2 + x_2^4 + 16x_1x_2 - 16 \\ 4x_1^3x_2 - 4x_1x_2^3 - 8x_1^2 + 8x_2^2 \end{bmatrix}.$$

The fourth function was designed to be easily programmable for arbitrary dimension. It was chosen to be the gradient of the function

$$\sum_{i=1}^{n} (x_i-1)^2 + \left(1 - \alpha \sum_{i=1}^{n} x_i^2\right)^2 .$$

Different choices of the parameter α can make the problem "easy" or "difficult". We chose $\alpha=0.35$ so that the number of iterations to solve the problem was moderately small. The gradient has components

$$[f_4(x)]_j = 0.6x_j - 2 + 0.49x_j \sum_{i=1}^{n} x_i^2 \qquad (j=1,\ldots,n) .$$

Table 1 shows numerical results for $n=2$ and 5.

Note that a large number of steps was required to bound the multiple zero of the function $f_3(x)$. The rate of convergence is linear for both methods.

The experiments were done on the HP9830B computer.

Table 1. *A comparison of methods.*

Function	Initial interval (same for each component)	Number of solutions in initial box	Number of steps	
			New method	Krawczyk method
f_1	$[-2,3]$	5	36	134
f_2	$[-2,2]$	3	81	101
f_3	$[1,2]$	1 (double)	1025	1310
$f_4(n=2)$	$[-1,1]$	1	4	5
$f_5(n=5)$	$[-1,1]$	1	13	17

REFERENCES

1. Götz Alefeld, *Intervallrechnung über den komplexen Zahlen und einige Anwendungen*, doctoral dissertation, University of Karlsruhe, 1968.
2. E. R. Hansen, *On solving systems of equations using interval arithmetic*, Math. Comp. 22 (1968), 374–384.
3. E. R. Hansen, *On linear algebraic equations with interval coefficients.* Topics in Interval Analysis, E. R. Hansen, ed., Oxford University Press, London, 1969.
4. E. R. Hansen, *Interval forms of Newton's method*, Computing 20 (1978), 153–163.
5. E. R. Hansen, *A globally convergent interval method for computing and bounding real roots*, BIT 18 (1978), 415–424.
6. E. R. Hansen, and R. R. Smith, *Interval arithmetic in matrix computations*, part II, SIAM Jour. Numer. Anal. 4 (1967), 1–9.
7. Richard Hanson, *Interval arithmetic as a closed arithmetic system on a computer*, Jet Propulsion Lab Report 197, June, 1968.
8. W. M. Kahan, *A more complete interval arithmetic*, Lecture notes for a summer course at the University of Michigan, 1968.
9. R. Krawczyk, *Newton-Algorithmen zur Bestimmung von Nullstellen mit Fehlerschranken*, Computing 4 (1969), 187–201.
10. R. E. Moore, *Interval Analysis*, Prentice–Hall, Englewood Cliffs, 1966.
11. R. E. Moore, *Methods and applications of interval analysis*, SIAM, Philadelphia, 1979.
12. M. A. Wolfe, *A modification of Krawczyk's algorithm*, SIAM Jour. Numer. Anal. 17 (1980), 376–379.

LOCKHEED MISSILES AND SPACE CO.
SUNNYVALE
CALIFORNIA
U.S.A.

DEPT. OF PURE AND APPLIED MATHEMATICS
WASHINGTON STATE UNIVERSITY
PULLMAN, WASHINGTON
U.S.A.