Reliable Computing **7:** 399–408, 2001.
© 2001 *Kluwer Academic Publishers. Printed in the Netherlands.*

Cheap and Tight Bounds on the Solution Set of Perturbed Systems of Nonlinear Equations

LUBOMIR V. KOLEV and IVO P. NENOV
Faculty of Automatica, Technical University of Sofia, 1756 Sofia, Bulgaria,
e-mail: lkolev@bmei.acad.bg, ivo@netwiz.net

(Received: 6 July 2000; accepted: 28 November 2000)

Abstract. This paper presents an iterative method for computing an outer interval bound on the solution set of parameters-dependent systems of non-linear equations for the case where the parameters take on their values within preset intervals. The method is based on a recently suggested alternative linear interval enclosure of factorable non-linear functions in a given box. It comprises two stages: during the first stage, a relatively narrow starting box is determined using an appropriate inflation technique while the second stage aims at reducing the width of the starting box.

Two algorithms implementing the method have been programmed in a C++ environment. Numerical examples seem to indicate that the second algorithm is rather efficient computation-wise.

The method is self-validating: the fulfillment of a simple inclusion rule checked during its second stage ensures that the interval bound thus found is an outer approximation to the solution set of the perturbed system investigated.

1. Introduction

The paper addresses the well-known problem of bounding the solution set of perturbed (i.e. parameters-dependent) systems of non-linear equations (e.g. [1]–[3], [10]–[12]). More specifically, let the system considered be

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

$$p \in \boldsymbol{p}, \tag{1.1b}$$

where $f : U \subseteq R^n \times R^m$, $D \subseteq R^n$, and $E \subseteq R^m$ are closed and connected sets with $D \times E \subseteq U$, and $\boldsymbol{p}$ is an m-dimensional interval vector in E. (For simplicity of notation, following [4], [8], throughout the paper interval quantities will be denoted by bold face letters while ordinary font letters will stand for real non-interval quantities.)

It is assumed that a pair $(x^0, p^0) \in U$ is known such that $f(x^0, p^0) = 0$ with $p^0 \in \boldsymbol{p}$; p^0 is usually the center of $\boldsymbol{p}$. The solution set $S_f(\boldsymbol{p})$ of (1.1) is the set

$$S_f(\boldsymbol{p}) := \{x : f(x, p) = 0,\ p \in \boldsymbol{p}\}. \tag{1.2}$$

The interval hull of $S_f(\boldsymbol{p})$ will be denoted $\boldsymbol{x}^*$; any other interval $\boldsymbol{x}$ such that $\boldsymbol{x}^* \subset \boldsymbol{x}$ will be referred to as an interval (outer) bound on $S_f(\boldsymbol{p})$. The width of $\boldsymbol{x}$ (or $\boldsymbol{x}^*$) serves as a measure for the sensitivity of the solution $\boldsymbol{x}(p)$ when p varies around p^0 in $\boldsymbol{p}$.

A method for determining $\boldsymbol{x}^*$ is suggested in [3]. It reduces to globally solving $2n$ constrained optimization problems. As it is rather time-consuming its applicability is limited to systems of low size n. Most often, a tight interval bound $\boldsymbol{x}$ is sought (e.g. [10]–[12]). In [2], [3], [10]–[12] use is made of either an interval extension $\boldsymbol{J}(\boldsymbol{x}^s, \boldsymbol{p})$ of the Jacobian to compute $\boldsymbol{x}$ or $\boldsymbol{x}^*$ ($\boldsymbol{x}^s \supset \boldsymbol{x}$) of (1.1) or an interval slope matrix.

In the present paper, we suggest a new approach to tackling the problem of finding a bound $\boldsymbol{x}$ on S_f. It is based on an alternative linear interval enclosure of factorable non-linear functions in a given box [5]–[7].

The paper is organized as follows. Section 2 presents the basic approach adopted and the main results thereby obtained. The new method for computing the bound $\boldsymbol{x}$ is presented in Section 3. Two numerical examples illustrating the applicability of the method suggested are given in Section 4. The paper ends up with final remarks in Section 5.

2. Main Results

Let $g : z \in R^q \rightarrow R$ be a continuous factorable function. It is known [7] that g can be enclosed by the following affine linear interval function

$$L_g(z) = \sum_{j=1}^{q} a_j z_j + \boldsymbol{b} \tag{2.1}$$

(where a_j are real numbers and $\boldsymbol{b}$ is an interval) having the property

$$g(z) \in L_g(z), \qquad z \in \boldsymbol{z}. \tag{2.2}$$

Similar formulae are valid in the case where $g : z \in R^q \rightarrow R^r$. Now

$$L_g(z) = Az + \boldsymbol{b}, \qquad z \in \boldsymbol{z}, \tag{2.3}$$

where A is a real matrix and $\boldsymbol{b}$ is an interval vector; for the new notation, property (2.2) is also valid. Constructive procedures for determining A and $\boldsymbol{b}$ are presented in [5]–[7].

Referring back to system (1.1), let $\boldsymbol{x}^s$ be a box large enough to contain $S_f(\boldsymbol{p})$, $\boldsymbol{x}^*$ and the bound $\boldsymbol{x}$ associated with a given $\boldsymbol{p}$. The main result of the section is formulated in the following theorem.

THEOREM 2.1. *Let $\boldsymbol{x}^* \subset \boldsymbol{x}^s$ and $\boldsymbol{x} \subset \boldsymbol{x}^s$. Furthermore, let*

$$L_f(\boldsymbol{x}^s, p) = A^x x + A^p p + \boldsymbol{b}, \qquad x \in \boldsymbol{x}^s, \ \ p \in \boldsymbol{p} \tag{2.4}$$

be the linear interval enclosure of (1.1) in $z = (\boldsymbol{x}^s, \boldsymbol{p})$. Also, let $S_L(\boldsymbol{p})$ denote the solution set of the linear interval system

$$A^x x + A^p p + \boldsymbol{b} = 0, \qquad p \in \boldsymbol{p}. \tag{2.5}$$

Then

$$S_f(\boldsymbol{p}) \subset S_L(\boldsymbol{p}). \tag{2.6}$$

Proof. Denote (2.4) equivalently as

$$L_f(z) = Az + \boldsymbol{b}, \qquad z \in \boldsymbol{z}, \tag{2.7}$$

where $z = (x, p)$. On account of the inclusion property (2.2)

$$f(z) = Az + \boldsymbol{b}, \qquad \forall z \in \boldsymbol{z}. \tag{2.8}$$

If $y \in \boldsymbol{z}$ is a zero of (1.1), then $f(y) = 0$. Hence from (2.8)

$$0 \in Ay + \boldsymbol{b}. \tag{2.9}$$

Let $\boldsymbol{b} = [\overline{b}, \underline{b}]$. The inclusion (2.9) can be written as

$$0 \leq Ay + \underline{b} \tag{2.10a}$$

and

$$0 \geq Ay + \overline{b} \tag{2.10b}$$

or equivalently

$$0 = Ay + b, \qquad b \in \boldsymbol{b}. \tag{2.11}$$

Returning back to the components A^x and A^p of A

$$A^x x + A^p p + b = 0, \qquad p \in \boldsymbol{p}, \ \ b \in \boldsymbol{b}. \tag{2.12}$$

So, if $x \in S_f(\boldsymbol{p})$, then there exists a pair (x, p) satisfying (2.12). But (2.12) defines the solution set $S_L(\boldsymbol{p})$ of (2.5). Hence $x \in S_f(\boldsymbol{p})$ implies $x \in S_L(\boldsymbol{p})$ which completes the proof. □

It is easily seen from (2.12) that the solution set $S_L(\boldsymbol{p})$ is a convex polyhedron.

Using elementary set-theoretical considerations, the following corollary can be readily proved.

COROLLARY 2.1. *The solution set $S_f(\boldsymbol{p})$ of (1.1) is also contained in the intersection*

$$S_{fp} = S_f(\boldsymbol{p}) \cap \boldsymbol{x}^s. \tag{2.13}$$

Let $\boldsymbol{h}_1$ denote the interval hull of S_{fp}. Then $\boldsymbol{h}_1$ is a bound on the solution set $S_f(\boldsymbol{p})$.

We can find a slightly wider bound than $\boldsymbol{h}_1$ in the following way. Rewrite (2.12) in the form

$$A^x x + \boldsymbol{b}' = 0, \qquad b' = \boldsymbol{b}', \tag{2.14a}$$

where

$$\boldsymbol{b}' = A^p \boldsymbol{p} + \boldsymbol{b}. \tag{2.14b}$$

Using the same argument as in Theorem 2.1 and Corollary 2.1, now we have the following results.

THEOREM 2.2. *Let $\boldsymbol{x}^* \subset \boldsymbol{x}^s$, $\boldsymbol{x} \subset \boldsymbol{x}^s$ and let $S_L(\boldsymbol{b}')$ denote the solution set of (2.14). Then*

$$S_f(\boldsymbol{p}) \subset S_L(\boldsymbol{b}'). \tag{2.15}$$

COROLLARY 2.2. *The solution set $S_f(\boldsymbol{p})$ of (1.1) is also contained in the intersection*

$$S_{fb'}(\boldsymbol{p}) = S_L(\boldsymbol{b}') \cap \boldsymbol{x}^s. \tag{2.16}$$

Let $\boldsymbol{h}_2$ be the interval hull of $S_{fb'}$; then $\boldsymbol{h}_2$ is another bound on the solution set of (1.1). It follows from elementary set-inclusion considerations that

$$\boldsymbol{h}_2 \supset \boldsymbol{h}_1. \tag{2.17}$$

It is easily seen that $\boldsymbol{h}_1$ or $\boldsymbol{h}_2$ can be determined by solving $2n$ linear programming problems associated with (2.12) or (2.14), respectively. Such an approach, however, appears to be rather costly for larger n. Therefore, a slightly wider but by far less expensive bound $\boldsymbol{h}_3$ will be suggested now. It is based on the following theorem.

THEOREM 2.3. *Let $\boldsymbol{x}^* \subset \boldsymbol{x}^s$, $\boldsymbol{x} \subset \boldsymbol{x}^s$ and*

$$\boldsymbol{h}_3 = -(A^x)^{-1}\boldsymbol{b}'. \tag{2.18}$$

Then

$$S_f(\boldsymbol{p}) \subset \boldsymbol{h}_3 \tag{2.19}$$

and

$$\boldsymbol{h}_1 \subset \boldsymbol{h}_2 \subseteq \boldsymbol{h}_3. \tag{2.20}$$

COROLLARY 2.3. *The solution set $S_f(\boldsymbol{p})$ of (1.1) is also contained in the intersection $\boldsymbol{h}_3 \cap \boldsymbol{x}^s$.*

The proof of the above theorem and corollary follows directly from Theorem 2.2 and Corollary 2.2.

Unlike $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$, the bound $\boldsymbol{x} = \boldsymbol{h}_3$ is determined in a comparatively much cheaper manner by just one single inversion of the real matrix A^x and a subsequent multiplication by an interval vector.

Remark 2.1. The bound $\boldsymbol{h}_3$ can be improved if rather than using (2.18) $\boldsymbol{h}_3$ is computed as follows

$$\boldsymbol{h}_3' = -C\boldsymbol{b} - \lfloor CA^p \rfloor \boldsymbol{p}, \tag{2.21}$$

where C is the inverse of A^x. The validity of (2.21) follows from (2.12) if (2.12) is first premultiplied by matrix C. Moreover, it is easily seen that

$$\boldsymbol{h}_3' \subseteq \boldsymbol{h}_3. \tag{2.22}$$

Indeed, from (2.14) and (2.18)

$$\boldsymbol{h}_3 = -C\boldsymbol{b} - C\lfloor A^p \boldsymbol{p} \rfloor. \tag{2.23}$$

Comparison of (2.21) with (2.23) and application of the subdistributivity property leads to (2.22). It should however be borne in mind that formulae (2.18), (2.14b) require a lesser volume of computation than (2.21) and may turn out to be a better choice for large-size problems.

Henceforth, to simplify presentation, only the cruder bound $\boldsymbol{h}_3$ will be used.

3. The New Method

In this section, we present a method for determining a bound $\boldsymbol{x}$ on the solution set $S_f(\boldsymbol{p})$ of (1.1). It consists of two stages: during the first stage, a "good" starting box $\boldsymbol{x}^s$ is determined; the second stage is based on Theorem 2.3 and aims at improving $\boldsymbol{x}^s$ by making it narrower.

From a computational efficiency point of view the selection of a good starting box for the second stage of the present method is of great importance. Indeed, if $\boldsymbol{x}^s$ is chosen too large, the second stage will take too many iterations to converge; conversely, if $\boldsymbol{x}^s$ is not large enough, it might not contain the outer solution $\boldsymbol{x} = \boldsymbol{h}_3$ as required by Theorem 2.3.

We start by presenting the first stage of the new method. This stage can be implemented in two different ways using the following two procedures.

PROCEDURE 3.1. We choose $p^0 = p^c$ (p^c is the centre of $\boldsymbol{p}$) and determine x^0 as the corresponding solution of $f(x, p^0)$. Now a narrow box $\boldsymbol{x}^0$ of small width ε_0 centered at x^0 is introduced and (1.1) is enclosed by the linear interval form (2.4) in $\boldsymbol{z}^0 = (\boldsymbol{x}^0, \boldsymbol{p})$, i.e. we determine

$$L_f(\boldsymbol{x}^0, \boldsymbol{p}) = A_0^x x + A_0^p p + \boldsymbol{b}_0, \tag{3.1a}$$

$$x \in \boldsymbol{x}^0, \qquad p \in \boldsymbol{p}. \tag{3.1b}$$

It is to be stressed that (3.1) is an enclosure of (1.1) only in z^0. However, (3.1a) will be used as a linear approximation of (1.1) in a larger box $z^1 = (\boldsymbol{x}^1, \boldsymbol{p})$. The component $\boldsymbol{x}^1$ of z^1 is determined in the following way. First, based on Theorem 2.3 we compute

$$x^1 = -(A_0^x)^{-1} b_0', \tag{3.2a}$$

where

$$b_0' = A_0^p p + b_0. \tag{3.2b}$$

Now the iterative procedure is started by putting $x^0 = x^1$ and going back to (3.1).

PROCEDURE 3.2. It is similar in structure to the previous procedure. The only difference lies in the way the component x^1 is determined at each iteration. We start as in Procedure 3.1 by computing x^1 using (3.2). At this point x^1 is renamed x' and the new x^1 is found by taking the union

$$x^1 = x' \cap x^0. \tag{3.2c}$$

Next we let $x^0 = x^1$ and the iterations continue from (3.1) as in the previous procedure.

At this point, we need the following assumption.

ASSUMPTION 3.1. *For a given box* p *Procedure 3.1 (Procedure 3.2) is convergent to a stationary interval vector* x^s *having the property*

$$x^* \subset x^s. \tag{3.3}$$

This assumption seems to be fulfilled most often in practice for relatively small boxes p and under reasonable requirements (such is given in e.g. [3], [10]–[12]) on the non-linear function f in (1.1). The inclusion (3.3) is expected because of the fact that at each iteration k before convergence the current approximation $L_f(x^{(k)}, p)$ of $f(x, p)$ becomes better and the box $x^{(k)}$ larger than $L_f(x^{(k-1)}, p)$ and $x^{(k-1)}$, respectively.

In practice, Procedure 3.1 (Procedure 3.2) is terminated whenever the distance between two successive iterations $x^{(k)}$ and $x^{(k-1)}$ becomes smaller than an accuracy ε_1. This approximate stationary box denoted as x^a may be smaller than the stationary box x^s. To facilitate inclusion (3.3), we inflate x^a, i.e. we let

$$x^s = x_c^a + (1 + \varepsilon_2)[-R, R], \tag{3.4}$$

where R is the radius of x^a and $\varepsilon_2 \geq 0$.

After the box x^s has been determined by (3.4) we proceed to the second stage of the present method. Now we try to reduce x^s using the following procedure.

PROCEDURE 3.3. We let $x^0 = x^s$ and construct the corresponding linear approximation of $f(x, p)$ in (x^0, p) using (3.1). By (3.2a) and (3.2b) we find the corresponding box x'. Next, a new box x^1 is introduced by the intersection.

$$x^1 = x' \cap x^0. \tag{3.5}$$

Now we let $x^0 = x^1$ and the iterative process continues from (3.1). It terminates when the distance between two successive boxes becomes smaller than an accuracy ε_3.

The distance used in the stopping criterion for Procedures 3.1 to 3.3 is computed as the maximum among the absolute values of the differences between the widths of the corresponding components.

It is seen that the method suggested above can be implemented as:

a) Algorithm A1 which is based on Procedures 3.1 and 3.3;

b) Algorithm A2 which uses Procedures 3.2 and 3.3.

Experimental evidence seems to indicate that Algorithm A2 requires less iterations than Algorithm A1 to solve the perturbed problem considered.

The second stage of the present method permits to computationally test the validity of inclusion (3.3) in Assumption 3.1. More precisely, we have the following result.

THEOREM 3.1. *Let* $\boldsymbol{x}^s$ *be determined by Procedure 3.1 or Procedure 3.2 using (3.4). Let* $\boldsymbol{x}^{(k)}$ *be the box obtained at the k-th iteration of Procedure 3.3 with* $\boldsymbol{x}^0 = \boldsymbol{x}^s$. *If the condition*

$$\boldsymbol{x}^{(k)} \subset \mathrm{int}(\boldsymbol{x}^s) \tag{3.6}$$

*(*int *denoting interior) is fulfilled for some* $k \geq 1$, *then the second stage of the method validates assumption (3.3).*

Proof. On account of Corollary 2.3 the solution set $S_f(\boldsymbol{p})$ as well as its interval hull $\boldsymbol{x}^*$ cannot have points lying outside the intersection $\boldsymbol{h}_3 \cap \boldsymbol{x}^s$. Thus, $\boldsymbol{x}^*$ cannot have points outside $\boldsymbol{x}^{(k)} \cap \boldsymbol{x}^s$. Now assume that (3.6) holds for some k. In this case, $\boldsymbol{x}^{(k)}$ lies strictly within $\boldsymbol{x}^s$ and is therefore encircled by a "ring" (formed by the difference $\boldsymbol{x}^s / \boldsymbol{x}^{(k)}$) which does not contain points belonging to $\boldsymbol{x}^*$. On the other hand, $\boldsymbol{x}^{(k)}$ is bound to contain $\boldsymbol{x}^*$ by construction, i.e.

$$\boldsymbol{x}^* \subset \boldsymbol{x}^{(k)}. \tag{3.7}$$

Finally, on account of (3.7), the validity of (3.6) implies the inclusion (3.3)

$$\boldsymbol{x}^* \subset \boldsymbol{x}^s,$$

which concludes the proof. □

Remark 3.1. We can reduce the overestimation of the bound $\boldsymbol{x}$ obtained by the present method appealing to the well-known technique of partitioning the parameter box $\boldsymbol{p}$ into a given number N of subboxes $\boldsymbol{p}^{(\nu)}$. We then apply the method to each subbox $\boldsymbol{p}^{(\nu)}$ to get a corresponding bound $\boldsymbol{x}^{(\nu)}$. The box $\boldsymbol{x}$ bounding the solution set of the original problem is now obtained as the interval hull of the union of all boxes $\boldsymbol{x}^{(\nu)}$. Obviously, such an approach is only applicable to problems where the dimension m of the parameter vector $\boldsymbol{p}$ is small.

4. Numerical Examples

In this section we give two examples illustrating the applicability of the method suggested. The examples have been solved by both Algorithms A1 and A2 with

Table 1.

		Stage 1	Stage 2	Total
N_i	Algorithm 1	9	4	13
	Algorithm 2	6	4	10

$\varepsilon_0 = \varepsilon_1 = \varepsilon_3 = 10^{-4}$. The algorithms were programmed using the algorithmic language C++. The linear interval enclosures (2.3) were generated automatically by a procedure that implements the approach suggested in [7].

EXAMPLE 4.1. The system of equation is

$$\begin{aligned} (e - x_1)/p_1 - x_3 &= 0, \\ x_1/p_2 - x_3 &= 0, \\ x_2 - x_1^2/(1 + x_1^2) &= 0, \end{aligned} \tag{4.1}$$

where e is a constant. In this example $x = (x_1, x_2, x_3)$ and $p = (p_1, p_2)$. We chose $e = 3.25$ and $p_1^0 = 2000$, $p_2^0 = 1000$. The corresponding point solution x^0 is

$$x^0 = (1.083, 0.5399, 0.001083). \tag{4.2}$$

The parameter vector $\boldsymbol{p}$ was chosen to be

$$\boldsymbol{p} = ([1800, 2200], [900, 1100]) \tag{4.3}$$

For this simple example, the interval hull $\boldsymbol{x}^*$ of the solution set of (4.1), (4.3) can be easily computed to be (approximately)

$$\boldsymbol{x}^* = ([0.9435, 1.2327], [0.4709, 0.6031], [0.00098, 0.001211]). \tag{4.4}$$

Application of Algorithms A1 and A2 with $\varepsilon_2 = 0.05$ led to the following bound on (4.4):

$$\boldsymbol{x} = ([0.9129, 1.254], [0.4546, 0.6182], [0.0009618, 0.001216]). \tag{4.5}$$

It is seen that the box (4.5) is an outer approximation of the solution set (4.4) of the perturbed system (4.1), (4.3).

The satisfaction of the inclusion (3.6) ensuring the validity of (4.5) was achieved for both algorithms at the first iteration of the second stage, i.e. for $k = 1$ of Procedure 3.3.

Table 1 lists the number of iterations N_i needed to terminate stages 1 and 2 of the respective algorithms as well as the total number of iterations for each algorithm. It is seen that Algorithm A2 requires less iterations as compared to Algorithm A1.

Table 2.

		Stage 1	Stage 2	Total
N_i	Algorithm 1	27	8	35
	Algorithm 2	13	7	20

EXAMPLE 4.2. In this example the perturbed system is

$$10^{-9}(e^{38x_1} - 1) + p_1x_1 - 1.6722x_2 + 0.6689x_3 - 8.0267 = 0,$$
$$1.98 \cdot 10^{-9}(e^{38x_2} - 1) + 0.6622x_1 + p_2x_2 + 0.6622x_3 + 4.0535 = 0, \quad (4.6a)$$
$$10^{-9}(e^{38x_3} - 1) + x_1 - x_2 + p_3x_3 - 6 = 0,$$

$$p = (p_1, p_2, p_3) \in ([0.6020, 0.7358], [1.2110, 1.4801], [3.6, 4.4]) \quad (4.6b)$$

and models an electric circuit containing a transistor, a diode and two resistors [9].

Application of Algorithms A1 and A2 yielded the following results, respectively:

$$\boldsymbol{x} = ([0.5401, 0.5682], [-3.8926, -3.1153], [0.3483, 0.5387]), \quad (4.7)$$
$$\boldsymbol{x} = ([0.5402, 0.5680], [-3.8910, -3.1194], [0.3473, 0.5331]). \quad (4.8)$$

For Algorithm A1, $\varepsilon_2 = 0.05$ and the fulfillment of (3.6) was achieved at $k = 1$ of Procedure 3.3. For Algorithm A2 we chose $\varepsilon_2 = 0$ and nevertheless (3.6) was satisfied already at $k = 2$ of Procedure 3.3. Thus, both bounds (4.7) and (4.8) are guaranteed to contain the solution set (4.6).

The numbers of iterations corresponding to the two algorithms are listed in Table 2. Once again, as in Example 4.1 Algorithm A2 outperforms Algorithm A1.

Example 4.2 was solved in [9] by an algorithm similar in structure to algorithm A2 (however, in [9] each iteration of both the first and second stage of the algorithm requires the solution of a corresponding linear interval system) and the following bound was obtained

$$\boldsymbol{x} = ([0.5103, 0.5778], [-4.3520, -2.6756], [0.3483, 0.5898]). \quad (4.9)$$

It is worth noting that the bound (4.9) is more conservative as compared to (4.7) and (4.8) and at the same time took more iterations to be reached: total number of iterations 166 (85 iterations for the first stage and 81 iterations for second stage).

5. Conclusion

A method for tackling the problem of bounding the solution set of a parameters-dependent non-linear systems of equations (1.1) by an interval box $\boldsymbol{x}$ has been proposed. The method is based on a recently suggested linear interval enclosure

(2.3) of the non-linear system involved. This approach is rather general since (2.3) can be constructed for the broad class of factorable functions, containing functions that may be only continuous.

The theoretical basis of the method is provided in Section 2: Theorems 2.1 to 2.3. The method proper is presented in Section 3 where two two-staged algorithms are suggested. According to Theorem 3.1, their second stage (Procedure 3.3) involves the computational verification of the validity of the algorithms.

The new method is implemented as a computer program written in C++. Numerical evidence seems to indicate that it provides cheap and tight bounds on the solution set of the perturbed non-linear systems investigated. These bounds are, however, not rigorous since the present implementation of the method does not account for round-off errors. It is the intention of the authors to develop an algorithm and a computer program which will implement the method with complete computational rigor, thus providing infallible outer bounds on the perturbed solution set.

Acknowledgements

The authors are thankful to the referees for their remarks and suggestions.

References

1. Adams, E.: On Sets of Solutions of Collections of Nonlinear Systems in $\mathbb{R}^n$, in: Nickel, K. (ed.), *Interval Mathematics*, Academic Press, New York—London—Toronto, 1980, pp. 247–256.
2. Gay, D.: Computing Perturbation Bounds for Nonlinear Algebraic Equations, *SIAM J. Numer. Anal.* **20** (1983), pp. 638–651.
3. Hansen, E.: *Global Optimization Using Interval Analysis*, Marcel Dekker Inc., New York, 1992.
4. Kearfott, R. B.: *Rigorous Global Search: Continuous Problems*, Kluwer Academic Publishers, Dordrecht—Boston—London, 1996.
5. Kolev, L.: A New Method for Global Solution of Nonlinear Equations, *Reliable Computing* **4** (2) (1998), pp. 125–146.
6. Kolev, L.: An Improved Method for Global Solution of Nonlinear Systems, *Reliable Computing* **5** (2) (1999), pp. 103–111.
7. Kolev, L.: Automatic Computation of a Linear Interval Enclosure, *Reliable Computing* **7** (1) (2001), pp. 17–28.
8. Kolev, L.: Interval Methods in Circuit Theory, in: Toshev, G., Hinova, I., and Kolev, L., *Theoretical Electrotechnics, Part 4*, Technica, Sofia, 1993, pp. 7–63.
9. Kolev, L. and Mladenov, V.: Worst-Case Tolerance Analysis of Non-Linear Circuits Using an Interval Method, in: *Proc. of the X International Symp. on Theoretical Electrical Eng.*, Magdeburg, Germany, September 6–9, 1999, pp. 621–623.
10. Krawczyk, R.: Interval Iteration for Including a Set of Solutions, *Comp.* **32** (1984), pp. 13–31.
11. Neumaier, A.: *Interval Methods for Systems of Equations*, Cambridge Univ. Press, Cambridge, 1990.
12. Neumaier, A.: Rigorous Sensitivity Analysis for Parameter-Dependent Systems of Equations, *J. Math. Anal. and Appl.* **144** (1.1) (1989), pp. 16–25.