136

IEEE TRANSACTIONS ON

CIRCUITS AND SYSTEMS

I: FUNDAMENTAL THEORY AND APPLICATIONS

A PUBLICATION OF THE IEEE CIRCUITS AND SYSTEMS SOCIETY

WWW.IEEE-CAS.ORG

DECEMBER 2002 VOLUME 49 NUMBER 12 ITCAEX (ISSN 1057-7122)

PAPERS

Analog Circuits and Filters

Cellular Neural Networks

Chaos and Bifurcation

Digital Circuits and Filters

Digital Integrated Circuits

Multidimensional Signals and Systems

Nonlinear Circuits and Systems

Power Electronics and Systems

Systems and Control

TRANSACTIONS BRIEFS

(Contents Continued on Back Cover)

IEEE

Worst-Case Tolerance Analysis of Linear DC and AC Electric Circuits

Lubomir Kolev, *Senior Member, IEEE*

***Abstract*—This paper addresses the problem of worst-case tolerance analysis of steady states in linear dc and ac electric circuits. The statement of the problem considered is in the form of linear algebraic equations whose elements are, in the general case, nonlinear functions of a given set of independent interval parameters. Three kinds of solutions are considered: 1) outer solution; 2) inner solution and 3) exact solution. A direct method for computing an outer solution and an iterative method for finding an inner solution are suggested. The inner and outer solutions thus found provide a tight two-sided bound on the exact solution of the tolerance problem investigated. The exact solution can be determined if certain monotonicity conditions are fulfilled. The verification of the conditions involves solving several associated outer solution problems. The computational efficiency of the methods suggested is demonstrated by a numerical example.**

***Index Terms*—Interval analysis, interval methods, linear equations with dependent elements, worst-case tolerance analysis.**

I. Introduction

WORST-CASE tolerance analysis of linear circuits (systems) is a well-established research area in circuit theory. The traditional approach to handling this problem is to use the Monte Carlo method. Starting with the pioneer works [1] and [2], an alternative approach based on the application of interval analysis technique [3]–[5] has been in existence for several decades. The methods utilizing the latter approach are known as interval methods (cf. [6], [7] and the bibliography therein cited). Because of their high reliability the interest in these methods has lately considerably increased.

Most worst-case tolerance analysis problems for linear circuits can be formulated in the following ways:

1) in explicit form as corresponding global optimization problems [2], [6, Ch. 2], [8], [9];
2) in implicit form using a system of linear interval equations [1], [6, Ch. 3], [10]–[13].

This paper falls into the latter group of investigations. All known methods pertaining to this group are based, in one way or another, on the exact or approximate solution of a system of linear algebraic equations whose elements are either independent intervals (in the case of dc circuits [6], [10]) or are assumed to be independent intervals (in the case of ac circuits [11]–[13]). The requirement for independence of the elements imposes stringent restrictions on the form of the system used: tableau forms [6] or the so-called hybrid form [11]–[13]. The system must have independent interval elements since the interval methods for solving linear systems applied so far to the solution of tolerance problems are only capable of treating such systems.

A more general and more flexible approach is adopted in this paper. According to this approach, the worst-case tolerance problem considered is solved using a corresponding system of linear algebraic equations whose elements are now interdependent. More precisely, each element can be a linear or even nonlinear function of a given set of independent parameters. These take on their values within certain prescribed intervals. The interval solution of the linear system is then transformed in a nonlinear fashion to provide the solution of the tolerance problem on hand. The new approach covers all possible worst-case tolerance problems related to dc and ac linear circuits. For brevity, these problems will be referred to as tolerance analysis (TA) problems. The following three kinds of interval solutions to such problems will be considered:

1) outer solution;
2) inner solution;
3) exact solution.

The exact solution is the narrowest possible interval solution of the TA problem considered (the rigorous definition of the unique exact solution will be given in the next section). An outer solution is any interval solution that contains the exact solution. Similarly, an inner solution is any interval solution that is contained in the exact solution. Since the determination of the exact solution is not always possible with acceptable computational cost, cheap and tight inner and outer solutions will provide a good two-sided bound on the exact solution. Such outer and inner solutions are obtained in the present paper. After presenting the formulation of the TA problems in Section II, a simple direct method for computing an outer solution is suggested in Section III. It reduces essentially to inverting a real $n \times n$ matrix and solving a system of n real linear algebraic equations, n being the size of the original real linear interval system describing the TA problem considered. In Section IV, a simple iterative method for computing a tight inner solution is proposed. Finally, a more involved method for determining the exact solution is presented in Section V. The latter method is, however, applicable only if certain monotonicity conditions are satisfied. It requires solving several associated outer solution problems to computationally check the above monotonicity conditions. The efficiency of the methods presented is illustrated by way of a numerical example in Section VI. The paper ends up with several concluding remarks.

Manuscript received June 22, 2001; revised June 26, 2002. This paper was recommended by Associate Editor N. M. K. Rao.

The author is with the Faculty of Automatica, Technical University of Sofia, 1756 Sofia, Bulgaria (e-mail: lkolev@vmei.acad.bg).

Digital Object Identifier 10.1109/TCSI.2002.805700

1057-7122/02$17.00 © 2002 IEEE

II. Formulation of the AT Problems

In this paper, an arbitrary TA problem can be formulated in the following manner. First, a real linear algebraic system of equations is set up

$$A(p)x = b(p) \tag{1a}$$

where p is an m-dimensional parameter vector, $A(p)$ and $b(p)$ are an $n \times n$ matrix and an n-dimensional vector, respectively. The elements of $A(p)$ and $b(p)$ are, in general, nonlinear functions of m parameters

$$a_{ij}(p) = a_{ij}(p_1, \dots, p_m) \tag{1b}$$
$$b_i(p) = b_i(p_1, \dots p_m) \tag{1c}$$

and the parameters take on their values within some prescribed intervals, i.e.,

$$p_k \in \boldsymbol{p}_k, \quad k = 1, \dots, m \tag{1d}$$

Here and henceforth, ordinary font letters will denote real quantities while bold face letters will stand for their interval counterparts. Thus, $p = (p_1, \dots, p_m)$ and $\boldsymbol{p} = (\boldsymbol{p}_1, \dots, \boldsymbol{p}_m)$ will denote a real and an interval vector of m components, respectively.

The next step to formulating the TA problem is to define the relationship

$$u = g(x) \tag{2}$$

which specifies the vector of output variables u and where $g : R^n \rightarrow R^q$, $1 \leq q \leq n$. The pair (1a)–(1d) and (2) formulates the TA problem to be solved.

At this point, the following assumption is needed.

Assumption 1: Each matrix $A(p)$, $p \in \boldsymbol{p}$, is nonsingular.

Later, it will be shown that Assumption 1 can be easily verified numerically by a certain sufficient condition (Theorem 1).

The solution set of the pair (1a)–(1d) and (2) is the set

$$S(\boldsymbol{p}) := \left\{u : u = g(x), x = A^{-1}(p)b(p),\ p \in \boldsymbol{p}\right\}. \tag{3}$$

The interval hull of $S(\boldsymbol{p})$ will be denoted $\boldsymbol{u}^*$ and $\boldsymbol{u}^*$ will be called exact (interval hull) solution to problem (1a)–(1d), (2). Any other interval $\boldsymbol{u}'$ such that $\boldsymbol{u}^* \subset \boldsymbol{u}'$ will be referred to as an outer solution to (1a)–(1d), (2). Similarly, an interval vector $\boldsymbol{u}''$ with the property $\boldsymbol{u}'' \subseteq \boldsymbol{u}^*$ will be referred to as an inner solution to (1a)–(1d), (2).

The description (1a)–(1d), (2) is rather general and covers all possible dc and ac TA problems formulated in [6, Sec. 3, problems 3.1–3.10] . Typically, functions (1b) and (1c) involved in most TA problems are affine (linear) functions of the elements of $\boldsymbol{p}$

$$a_{ij}(p) = \alpha_{ij} + \sum_{k=1}^{m} \alpha_{ijk} p_k \tag{4a}$$
$$b_i(p) = \beta_i + \sum_{k=1}^{n} \beta_{ik} p_k. \tag{4b}$$

It is, however, known [6] that in the general case, some of the elements a_{ij} will be nonlinear functions if the circuit investigated involves dependent current sources with interval coefficients. Also, some of the elements b_j will be nonlinear functions if the magnitudes of some voltage or current sources are not known exactly (and are therefore given as intervals) and loop analysis is used to set up the TA system of equations. The nonlinear functions encountered in practice are continuously differentiable with respect to the parameters considered.

The specific TA problem considered is finally defined by choosing the function $g(x)$ in (2). In some cases, $g(x)$ is a linear function. Thus, if we are interested in the determination of the tolerances on all the components of x (as in [6, prob. 3.7, 3.8]), then

$$u = x \tag{5a}$$

so

$$g = E \tag{5b}$$

(where E denotes the identity matrix). In the overwhelming majority of applications, the problem is to estimate only the range of n' components $\boldsymbol{u}_i^*$ and $n' \ll n$ (typically $n' = 1$ in the case of direct current electrical circuits or $n' = 2$ in the case of alternative current electrical circuits). If we want to find the tolerance on one single component x_k of x, then (2) become

$$u_k = e_k^T \cdot x \tag{6}$$

where e_k^T is the transpose of the kth column of E. If the problem is to estimate the tolerance on n' output variables (with $n' > 1$), then the pair (1a)–(1d) and (6) is solved n' times.

In the general case [6, prob. 3.9, 3.10], $g(x)$ is a nonlinear function. For example, if we want to find the tolerance on the magnitude of a nodal voltage $\dot{V}_k$ [6, problem 3.9]), then

$$u = x_k^2 + x_{k+1}^2 \tag{7}$$

where x_k and x_{k+1} are the real and imaginary parts of the complex nodal voltage $\dot{V}_k$.

Once the TA problem is defined by fixing (1a)–(1d) and (2), we have finally to specify which of the solutions outer, inner, or exact is to be found.

III. Outer Solution

In this section, first a method for determining an outer solution $\boldsymbol{x}'$ to problem (1a)–(1d) and (5a)–(5b) is suggested. The solution $\boldsymbol{x}'$ is then used to find an outer solution $\boldsymbol{u}'$ to problem (1a)–(1d) and (2).

The derivation of the method is based on the general approach employed in [16], [17], on the one hand, and the result for the case of affine functions $a_{ij}(p_1, \dots, p_m)$ [19], on the other.

To apply this approach, we need the following preliminary facts [14]–[16]. First, let $f : \mathrm{x} \subset R^n \rightarrow R^q$ be a continuously differentiable function. The function $f(x)$ can be enclosed in the interval vector $\boldsymbol{x}$ by the following linear interval form:

$$L_f(x) = Ax + \boldsymbol{a}, \quad x \in \boldsymbol{x} \tag{8}$$

where A is a $q \times n$ real (noninterval) matrix while $\boldsymbol{a}$ is an interval vector. The form (8) can be determined in an automatic way using the algorithm of [16]. It has the inclusion property

$$f(x) \in \boldsymbol{L}_f(x), \quad x \in \boldsymbol{x}. \tag{9}$$

Secondly, consider the product

$$xy, \quad x \in \boldsymbol{x}, \quad y \in \boldsymbol{y} \tag{10}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are intervals. If x_0, y_0 and r_x, r_y are the respective centers and radii, then [14]

$$xy \in -x_0 y_0 + y_0 x + x_0 y + [-r_x r_y, r_x r_y]. \tag{11}$$

In accordance with (8), the corresponding linear interval forms of (4a) and (4b) are

$$\boldsymbol{L}_{ij}(p) = \sum_{k=1}^{m} \alpha_{ijk} p_k + \boldsymbol{a}_{ij}, \quad p \in \boldsymbol{p} \tag{12a}$$

$$\boldsymbol{l}_i(p) = \sum_{k=1}^{m} \beta_{ik} p_k + \boldsymbol{b}_i, \quad p \in \boldsymbol{p} \tag{12b}$$

and have the inclusion property

$$a_{ij}(p) \in \boldsymbol{L}_{ij}(p), \quad p \in \boldsymbol{p} \tag{13a}$$

$$b_i(p) \in \boldsymbol{l}_i(p), \quad p \in \boldsymbol{p}. \tag{13b}$$

The approach herein adopted to determining an outer solution to (1a)–(1d) is based on the use of (8)–(13) and the method for solving perturbed systems of equations in [17]. For this purpose, (1a)–(1d) is written in the form

$$f(x, p) = A(p)x - b(p) = 0, \quad p \in \boldsymbol{p}. \tag{14}$$

We temporarily assume $\boldsymbol{x}$ is a known interval vector. Then, $f(x, p)$ can be enclosed in $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{p})$ by the linear interval form

$$\boldsymbol{L}_f(x, p) = A^x x + A^p p + \boldsymbol{v}, \quad x \in \boldsymbol{x}, \quad p \in \boldsymbol{p} \tag{15}$$

where A^x and A^p are $n \times n$ and $n \times m$ real matrices. On account of the inclusion property

$$f(x, p) \in \boldsymbol{L}_f(x, p), \quad x \in \boldsymbol{x}, \quad p \in \boldsymbol{p}. \tag{16}$$

Now, we shall obtain explicit expressions for A^x and A^p and $\boldsymbol{v}$. With this in mind, we first introduce the shorter notation $\boldsymbol{L}_{ij}$ for the intervals $\boldsymbol{L}_{ij}(\boldsymbol{p})$ and $\boldsymbol{l}_i$ for the intervals $\boldsymbol{l}_i(\boldsymbol{p})$. Let $\boldsymbol{L}$ denote the interval matrix whose elements are $\boldsymbol{L}_{ij}$ while L^0 denotes its center. Also, let x^0 be the center of vector $\boldsymbol{x}$. Then, on account of (11)

$$A(p)x \in L^0 x + L x^0 + \boldsymbol{c}, \quad x \in \boldsymbol{x}, \quad L \in \boldsymbol{L} \tag{17a}$$

where

$$\boldsymbol{c} = -L^0 x^0 + [-r^c, r^c]. \tag{17b}$$

In (17b), the radius r^c of $\boldsymbol{c}$ is given by

$$r^c = R^L r^x \tag{17c}$$

where R^L is the radius of $\boldsymbol{L}$ and r^x is the radius of $\boldsymbol{x}$. It is seen from (12a) that the elements R^L_{ij} of R^L are computed as

$$R^L_{ij} = \sum_{k=1}^{m} |\alpha_{ijk}| r^p_k + R^a_{ij} \tag{17d}$$

where r^p_k is kth component of the radius r^p of $\boldsymbol{p}$ and R^a_{ij} is the radius of $\boldsymbol{a}_{ij}$. On account of (12a), the elements L^0_{ij} of L^0 are

$$L^0_{ij} = a^0_{ij} + \sum_{k=1}^{m} \alpha_{ijk} p^0_k. \tag{17e}$$

On the other hand, using (12a)

$$\left(\boldsymbol{L}x^0\right)_i = \sum_{j=1}^{n} \boldsymbol{L}_{ij} x^0_j = \sum_{k=1}^{m} \alpha_{ik} \boldsymbol{p}_k + \boldsymbol{a}_i \tag{18a}$$

where

$$\alpha_{ik} = \sum_{j=1}^{n} x^0_j \alpha_{ijk} \tag{18b}$$

$$\boldsymbol{a}_i = \sum_{j=1}^{n} x^0_j \boldsymbol{a}_{ij}. \tag{18c}$$

In a similar way

$$l^0_i = b^0_i + \sum_{k=1}^{m} \beta_{ik} p^0_k \tag{19a}$$

$$\boldsymbol{l}_i = \boldsymbol{b}_i + \sum_{k=1}^{m} \beta_{ik} \boldsymbol{p}_k. \tag{19b}$$

Let $\boldsymbol{l}$ be the interval vector whose components are defined through (19b). Combining

$$b(p) \in \boldsymbol{l} \tag{20}$$

(17a), and (18a), we finally get the explicit expressions for A^x, A^p and $\boldsymbol{v}$ in (15)

$$A^x = L^0 \tag{21a}$$

$$A^p_{ik} = \sum_{j=1}^{n} \alpha_{ijk} x^0_j - \beta_{ik} \tag{21b}$$

$$\boldsymbol{v} = \boldsymbol{a} + \boldsymbol{b} + \boldsymbol{c}. \tag{21c}$$

Now consider the linear interval system related to (15)

$$L^0 x + A^p p + \boldsymbol{v} = 0, \quad p \in \boldsymbol{p} \tag{22}$$

Let $B = (L^0)^{-1}$ and $C = BA^p$. From (22), the solution for x denoted $\boldsymbol{h}$ is

$$\boldsymbol{h} = -C\boldsymbol{p} - B(\boldsymbol{a} + \boldsymbol{b}) - B\boldsymbol{c}. \tag{23}$$

So far, we have assumed that $\boldsymbol{x}$ is a known interval vector. Now, we proceed to determining $\boldsymbol{x}$ as an outer solution $\boldsymbol{x}'$.

First, we determine the center x^0 of $\boldsymbol{x}'$ as the solution of

$$L^0 x^0 = l^0 \tag{24}$$

where L^0 and l^0 are defined by (17e) and (19a), respectively.

We have next to determine the radius r of $\boldsymbol{x}'$. On account of (23), (17a)–(17e), and (21)

$$r^h = |C|r^p + |B|\left(r^a + r^b\right) + |B|Rr^s. \tag{25}$$

Let for simplicity of notation

$$c = |C|r^p + |B|\left(r^a + r^b\right) \tag{26a}$$

$$D = |B|R. \tag{26b}$$

Taking into account (25), a reasonable choice for r^h is to determine it as the solution y^* of the equation

$$y = c + Dy \tag{27}$$

or equivalently

$$(I - D)y = c. \tag{28}$$

The main result of this section is the following theorem.

Theorem 1: Assume the solution y^* to system (28) is positive. Then

1) the interval vector

$$\boldsymbol{x} = x^0 + \boldsymbol{h}' \tag{29a}$$

where

$$\boldsymbol{h}' = [-y^*, y^*] \tag{29b}$$

is an outer solution to (1a)–(1d);

2) matrix $A(p)$ is nonsingular for each $p \in \boldsymbol{p}$.

The proof of the above theorem is similar to that of Theorem 2 in [19] and will therefore be omitted. The method used to solve (28) should however guarantee the nonsingularity of matrix $I - D$.

Based on Theorem 1, the present method for determining an outer solution to system (1a)–(1d) comprises the following computations. First, we evaluate matrix L^0 as well as matrix R using (17e) and (17d). Next, L^0 is inverted to get matrix B. Using (26), we set up system (28). If the solution of (28) is positive, then the outer solution is obtained from (29). If, on the other hand, system (28) does not have a positive solution, the method is not applicable.

If the TA problem is to find an outer solution $\boldsymbol{u}'$ to (1a)–(1d) and (2), $\boldsymbol{u}'$ can be computed as follows:

$$\boldsymbol{u} = g(\boldsymbol{x}) \tag{30}$$

where $\boldsymbol{u}$ is the range of g in $\boldsymbol{x}$.

The above method for computing an outer approximation to the exact solution of the TA problem considered will be referred to as method M1.

IV. Inner Solution

In this section, first a simple iterative method for determining an inner solution $\boldsymbol{x}''$ to system (1a)–(1d) will be presented. In fact, we compute individually each component $\boldsymbol{x}''_k$ of $\boldsymbol{x}''$. As in the previous section, the method will then be extended to determining the component $\boldsymbol{u}''_k$ of the inner solution $\boldsymbol{u}''$ of problem (1a)–(1d) and (2).

The method is based on a local optimization technique and appeals to two procedures which determine separately the lower endpoint $\underline{x''}_k$ and the upper endpoint $\overline{x}''_k$ of $\boldsymbol{x}''_k$. Each procedure makes use of the derivatives of x_k with respect to p_l, $l = 1, \ldots, m$. These derivatives are computed in the following way.

System (1a) is written in detailed form as

$$\sum_{j=1}^{n} a_{ij}\left(p_1, \ldots m\right) x_j = b_i\left(p_1, \ldots m\right) = 0, \quad i = 1, \ldots, n. \tag{31}$$

We are interested in expressing the derivative of x_j with respect to p_l, $l = 1, \ldots m$. With this in mind, we differentiate (31) in p_l and on account of (1b) and (1c) we get

$$\sum_{j=1}^{n} a_{ij}(p)\frac{\partial x_j}{\partial p_l} = \gamma_{il}(p) - \sum_{j=1}^{n} \eta_{ijl}(p)x_j, \quad i = 1, \ldots, n \tag{32a}$$

where

$$\gamma_{il}(p) = \frac{\partial b_i(p)}{\partial p_l} \tag{32b}$$

$$\eta_{ijl}(p) = \frac{\partial a_{ij}(p)}{\partial p_l}. \tag{32c}$$

Systems (32a) and (32b)–(32c) will be rewritten as

$$A(p)\frac{dx}{dp_l} = \gamma_l(p) - \eta_l(p)x(p) \tag{32d}$$

where $\gamma_l(p)$ is a column vector and $\eta_l(p)$ is a matrix. Hence, if $B(p) = A^{-1}(p)$

$$\frac{dx}{dp_l}(p) = B(p)\left(\gamma_l(p) - \eta_l(p)x(p)\right), \quad p \in \boldsymbol{p}. \tag{33}$$

If we are interested in the kth component of dx/dp_l, then (33) becomes

$$\frac{dx_k}{dp_l}(p) = B_k(p)\left(\gamma_l(p) - \eta_l(p)x(p)\right), \quad p \in \boldsymbol{p} \tag{34}$$

where $B_k(p)$ is the kth row of $B(p)$. Now, as can be easily seen, $B_k(p)$ can be computed in a most efficient manner as follows. For a fixed p we solve the real system

$$A^T(p)y = e_k, \quad p \in \boldsymbol{p} \tag{35}$$

where A^T denotes the transpose of A and e^k is the kth column of the identity matrix. Finally, B_k is obtained as y^T.

We have the following procedure for finding the lower endpoint $\underline{x}''_k$ of $\boldsymbol{x}''_k$.

Procedure 1: For a fixed k we start by evaluating the derivative $d_{kl}(p) = dx_k/dp_l(p)$ for $p = p^0$. Let $d^0_{kl} = d_{kl}(p^0)$, $A^0 = A(p^0)$, $B^0_k = B_k(p^0)$, $\gamma^0_l = \gamma_l(p^0)$, $\eta^0_l = \eta(p^0)$ and $x^0 = x(p^0)$. On account of (34)

$$d^0_{kl} = B^0_k\left(\gamma^0_l - \eta^0_l x^0\right) \tag{36}$$

where

$$B^0_k = \left(y^0\right)^T \tag{37a}$$

and y^0 is the solution of the real system

$$\left(A^0\right)^T y = e_k. \tag{37b}$$

Now, we determine new values p_l^1, $l = 1, \ldots, m$, using the following formula

$$p_l^1 = \begin{cases} \underline{p}_l, & \text{if } d_{kl} \geq 0 \\ \overline{p}_l, & \text{if } d_{kl} < 0 \end{cases}, l = 1, \ldots, m \tag{38}$$

and form the vector $p^1 = (p_l^1, \ldots, p_m^1)$. We then solve the system

$$A\left(p^1\right) x = b\left(p^1\right) \tag{39}$$

to find the vector x^1. If

$$x_k^1 < x_k^0 \tag{40}$$

x_k^1 is renamed x_k^0, p_l^1 are renamed p_l^0, and the procedure is resumed from the start; otherwise, the procedure is terminated and the inner bound on $\underline{x}_k^*$ is given by the corresponding component x_k^0.

A similar procedure is valid for determining the upper endpoint $\overline{x}_k''$ of $\boldsymbol{x}_k''$.

Procedure 2: For a fixed k, we let $p = p^0$ and repeat the computations (36) and (37a)–(37b) of Procedure 1. Now, we determine a new vector y^1 with components x^1 using the formula

$$p_l^1 = \begin{cases} \overline{p}_l, & \text{if } d_{kl} \geq 0 \\ \underline{p}_l, & \text{if } d_{kl} < 0 \end{cases}, l = 1, \ldots, m. \tag{41}$$

Then, the corresponding system (39) is solved and a new vector x^1 is thus found. If

$$x_k^1 > x_k^0 \tag{42}$$

x_k^1 is renamed x_k^0, and p_l^1 are renamed p_l^0 and the procedure is resumed from the start; otherwise the procedure is stopped and the inner bound on $\overline{x}_k^*$ is given by the corresponding component x_k^0.

It is seen that Procedures 1 and 2 implement a strategy which is based on a local optimization scheme. If the actual dependencies of dx_k/dp_l on dp_l in $\boldsymbol{p}$ are monotone functions, these procedures will produce, in fact, the exact solution $\boldsymbol{x}_k^*$. In the general case, Procedures 1 and 2 will only provide inner bounds on $\boldsymbol{x}_k^*$.

Combined with the outer bounds $\boldsymbol{x}'$ the pair $(x'', \boldsymbol{x}')$ provides a two-sided estimate of the exact solution $\boldsymbol{x}^*$ of system (1a)–(1d). The width of the interval vector $\boldsymbol{w} = \boldsymbol{x}' - \boldsymbol{x}''$ can serve as a measure of the accuracy of the approximations $\boldsymbol{x}'$ and $\boldsymbol{x}''$.

The above approach can also be applied in the case of handling the TA problem (1a)–(1d) and (2). The only difference is that now we compute the derivatives of u with respect to p_l. Thus, for u defined by (7)

$$\frac{du}{dp_l}(p) = 2x_k(p)\frac{dx_k}{dp_l}(p) + 2x_{k+1}(p)\frac{dx_{k+1}}{dp_l}(p), \quad p \in \boldsymbol{p}. \tag{43}$$

The inner solution $\boldsymbol{u}''$ is then found using Procedures 1 and 2 that have been modified accordingly.

V. Exact Solution

In this section, the exact solution to the TA problem (1a)–(1d) and (2), or problem (1a)–(1d) and (5b) will be sought. For simplicity of the presentation, first a method for computing the exact solution $\boldsymbol{x}_k^*$ of the simpler TA problem (1a)–(1d) and (5b) will be suggested. The method is applicable only if the derivatives dx_k/dp_l are guaranteed to be monotone in $\boldsymbol{p}$. These monotonicity conditions can be checked in the following way.

Let $\boldsymbol{x}'$ be the outer solution of (1a)–(1d) computed by the method presented in Section III. Similarly, let $\boldsymbol{B}_k$, $\boldsymbol{\gamma}$ and $\boldsymbol{\eta}$ denote the enclosure of the respective quantities for $p \in \boldsymbol{p}$. Then, we can define $\boldsymbol{D}_{kl}$ as follows:

$$\boldsymbol{D}_{kl} = \boldsymbol{B}_k\left(\boldsymbol{\gamma}_l - \boldsymbol{\eta}_l \boldsymbol{x}'\right) \tag{44}$$

and obviously

$$\frac{dx_k}{dp_l}(p) \in \boldsymbol{D}_{kl}, \quad p \in \boldsymbol{p}. \tag{45}$$

Hence, the derivative considered is guaranteed to be monotone in $\boldsymbol{p}$ if $0 \notin \boldsymbol{D}_{kl}$.

For the special case of linear functions $a_{ij}(p)$ and $b_i(p)$ the matrices γ and η are constant.

As in the previous section, $\boldsymbol{B}_k$ can be computed as an outer solution y of the following system:

$$A^T(p) y = e_k, \quad p \in \boldsymbol{p}. \tag{46}$$

To get a narrow interval vector $\boldsymbol{y}$, system (46) will be solved by the method M1 from Section III. Finally, $\boldsymbol{B}_k$ is obtained as $\boldsymbol{y}^T$.

Using (45), we determine the estimates $\boldsymbol{D}_{kl}$. Now we make the following assumption.

Assumption 2: We assume that each estimate $\boldsymbol{D}_{kl}$, $l = 1, \ldots, m$, satisfies either the condition

$$\boldsymbol{D}_{kl} \geq 0 \tag{47a}$$

or the condition

$$\boldsymbol{D}_{kl} \leq 0. \tag{47b}$$

On account of inclusion (45) the fulfillment of Assumption 2 guarantees that x_k is monotone with respect to each parameter p_l. Now, we define two vectors $\underline{p}^{(k)}$ and $\overline{p}^{(k)}$ as follows:

$$\underline{p}_l^{(k)} = \begin{cases} \underline{p}_l, & \text{if } \boldsymbol{D}_{kl} \geq 0 \\ \overline{p}_l, & \text{if } \boldsymbol{D}_{kl} \leq 0 \end{cases}, l = 1, \ldots, m \tag{48a}$$

$$\overline{p}_l^{(k)} = \begin{cases} \overline{p}_l, & \text{if } \boldsymbol{D}_{kl} \geq 0 \\ \underline{p}_l, & \text{if } \boldsymbol{D}_{kl} \leq 0 \end{cases}, l = 1, \ldots, m. \tag{48b}$$

The exact solution $\boldsymbol{x}_k^*$ of system (1a)–(1d) can be found using the following theorem.

Theorem 2: If Assumption 2 holds, then the kth component $\boldsymbol{x}_k^* = (\underline{x}_k^*, \overline{x}_k^*)$ of the exact solution $\boldsymbol{x}^*$ is determined as follows.

1) $\underline{x}_k^*$ is equal to the kth component of the solution of

$$A\left(\underline{p}^{(k)}\right) x = b\left(\underline{p}^{(k)}\right). \tag{49a}$$

2) $\overline{x}_k^*$ is equal to the kth component of the solution of

$$A\left(\overline{p}^{(k)}\right) x = b\left(\overline{p}^{(k)}\right) \tag{49b}$$

where the vector $\underline{p}^{(k)}$ and $\overline{p}^{(k)}$ are determined according to (48a)–(48b).

The theorem follows directly from the above considerations about the monotonicity of $dx/dp^{(l)}$.

On the basis of the aforegoing we have the following procedure for determining one component $\boldsymbol{x}_k^*$ of the exact solution $\boldsymbol{x}^*$. Before initiating the procedure, however, we solve system (28) to get an outer solution $\boldsymbol{x}'$ of (1a)–(1d).

Procedure 3: For a given k, solve system (46) using method M1 to find the interval vector $\boldsymbol{B}_k$. By (44) compute $\boldsymbol{D}_{kl}$, $l = 1, \dots, m$. Check conditions (47). If all of them are satisfied, determine the two real vectors $\underline{y}^{(k)}$ and $\overline{y}^{(k)}$ using (48a)–(48b). Finally, solve systems (49a)–(49b) to get the lower end-point $\underline{x}_k^*$ and the upper end-point $\overline{x}_k^*$ of the kth component $\boldsymbol{x}_k^*$ of the exact solution to system (1a)–(1d).

In some cases, it is possible to determine $\boldsymbol{x}_k^*$ even if not all conditions (47a)–(47b) are satisfied, that is if Procedure 3 is not applicable. Indeed, let (for a fixed k) I_1 denote the set of those indices l for which either (47a) or (47b) holds while I_2 denotes the set of indices l for which (47a)–(47b) is violated. Using (48a)–(48b) in which now $l \in I_1$, we can determine those components of the vectors $\underline{p}$ and $\overline{p}$ which are guaranteed to take on end-point values. Thus, each vector can be partitioned into two parts as follows:

$$p = \left(\underline{p}^{(1)}, p^{(2)}\right) \tag{50a}$$

or

$$p = \left(\overline{p}^{(1)}, p^{(2)}\right) \tag{50b}$$

where only the components $p_l^{(2)}$, $l \in I_2$ of $p_l^{(2)}$ are allowed to take on their values within the corresponding intervals $\boldsymbol{p}_l$ forming the interval vector $\boldsymbol{p}^{(2)}$. Let the index set I_1 have m_1 members and I_2 have m_2 members. Since according to (50a)–(50b) the first m_1 components of p are fixed to end-point values it is seen that the original interval vector $\boldsymbol{p}$ has been reduced to a new m_2-dimensional interval vector $\boldsymbol{p}^{(2)}$ whose components are $\boldsymbol{p}_l$ with $l \in I_2$.

We first consider partition (50a). We shall present a procedure for determining the lower end-point $\underline{x}_k^*$ of $\boldsymbol{x}_k^*$.

Procedure 4: For a fixed k, find the outer solution $\boldsymbol{x}$ of the following modified system of type (1a)–(1d)

$$A\left(\underline{p}^{(1)}, p^{(2)}\right) x = b\left(\underline{p}^{(1)}, p^{(2)}\right), \quad p^{(2)} \in \boldsymbol{p}^{(2)}. \tag{51a}$$

Also, find the outer solution $\boldsymbol{y}$ of the modified system

$$A\left(\underline{p}^{(1)}, \boldsymbol{p}^{(2)}\right) y = e_k \tag{51b}$$

to get the corresponding interval vector $\boldsymbol{B}_k$. Thus, we can compute by (43) the elements $\boldsymbol{D}_{kl}$, $l \in I_2$ and check the monotonicity conditions (47a)–(47b). Now we assume that these conditions are fulfilled. Using (48a) we find the reduced-sized vector $\underline{p}^{(2)}$ whose components are $\underline{p}_l^{(2)}$, $l \in I_2$. Finally, $\underline{x}_{(k)}^*$ is computed as the kth component of the solution of

$$A\left(\underline{p}^{(1)}, \underline{p}^{(2)}\right) x = b\left(\underline{p}^{(1)}, \underline{p}^{(2)}\right). \tag{52}$$

We next present a procedure for determining the upper end-point $\overline{x}_k^*$ of $\boldsymbol{x}_k^*$.

Procedure 5: It has, essentially, the some structure as Procedure 4. Now we solve systems (51a)–(51b) in which $\underline{p}^{(1)}$ is replaced with $\overline{p}^{(1)}$. Thus, we compute the elements $\boldsymbol{D}_{kl}$ corresponding to $(\overline{p}^{(1)}, \boldsymbol{p}^{(2)})$. Once again we assume that the monotonicity conditions (47a)–(47b) are fulfilled. Using (48a)–(48b) we find the real vector $\overline{p}^{(2)}$ whose components are $\overline{p}_l^{(2)}$, $l \in I_2$. Finally, $\overline{x}_k^*$ is computed as the kth component of the solution of

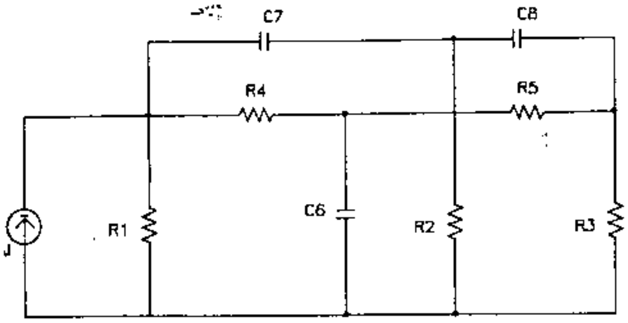

Fig. 1. Notch filter circuit.

$$A\left(\overline{p}^{(1)}, \overline{p}^{(2)}\right) x = b\left(\overline{p}^{(1)}, \overline{p}^{(2)}\right). \tag{53}$$

The above approach to computing $\boldsymbol{x}_k^*$ will be referred to as method M3.

Method 3 can be used only if both Procedures 4 and 5 are applicable. If, however, the monotonicity conditions (47a)–(47b) are not fulfilled for all components of $\boldsymbol{p}^{(2)}$, a new attempt can be made to determine $\boldsymbol{x}_k^*$. With this in mind, we treat $\boldsymbol{p}^{(2)}$ as a new reduced-size parameter vector. We then partition the new $\boldsymbol{p}^{(2)}$ into two parts and apply once again method M3. This new computational scheme will also be referred to as method M3.

Method M3 is also applicable in the general case of nonlinear relation (2). Thus, for function (7) we have once again to appeal to formula (43) where, however, all real variables are to be replaced with their interval counterparts. Afterwards, we employ Procedures 4 and 5 that have been modified accordingly.

VI. Numerical Example

The new methods will be illustrated with the following example. The linear AC circuit considered [11] is shown in Fig. 1. The nominal (center) values of the interval element parameters are

$$R_i^c = 10^4\ \Omega, \quad i = 1, \dots, 5 \tag{54a}$$

$$C_6^c = 2 \cdot 10^{-4}\ \mathrm{F}, \quad C_7^c = C_7^c = 10^{-4}\ \mathrm{F} \tag{54b}$$

while the fixed quantities are

$$\omega = 10^3\ \mathrm{s}^{-1}, \quad J = 10^{-3}\ \mathrm{A}. \tag{54c}$$

The TA problem to be solved consists in finding the outer, inner, and exact solutions related to the real part V of the output voltage $\dot{V}_3$ for several values of the tolerances on R_i and C_i. Thus, the TA problem considered here is of the type (1a)–(1d) and (6).

The numerical experiments are done in a MATLAB environment on a 400-MHz Pentium II PC. The interval arithmetic operations are implemented using the toolbox INTLAB. For space limitation, the numerical results obtained will be reported to only four decimal places.

A. Outer Solutions

Unlike the algorithms using real (complex) calculations, the results obtained by interval algorithms depend on the system describing the problem on hand. Thus, the nominal value of V will be the same notwithstanding whether the system of equations used is set up by nodal analysis, loop analysis or in tableau form. It will be shown that the result for the outer solution, however, is different depending on what type of analysis equations is employed. For the purpose of comparison three types of system of equations will be considered.

System S1: This is a system of equations of the type used in [10] or [6, Ch. 3]. It consists of $2n = 2(n' + m')$ equations where n' and m' is the number of unknown nodal voltages and branch currents, respectively

$$Ax = b \tag{55a}$$

and the interval entries lie only in the first 16 positions along the main diagonal of the coefficient matrix A. The interval coefficients are dependent since

$$a_{ii} = a_{i+m}, \ i + m = p_i, \ p_i \in \boldsymbol{p}_i, \quad i = 1, \ldots, m. \tag{55b}$$

For the circuit analyzed $n' = 4$, $m' = 8$ so $n = 24$ and

$$\boldsymbol{p}_i = \boldsymbol{R}_i = R_i^c + t\,[-R_i^c, R_i^c] \tag{56a}$$

where t is the tolerance chosen and $[-R_i^c, R_i^c]$ is a symmetric interval

$$\boldsymbol{p}_i = \boldsymbol{X}_i, \quad i = 6, 7, 8 \tag{56b}$$

with

$$\boldsymbol{X}_i = \frac{1}{\boldsymbol{B}_i} \quad, B_i^c = \omega C_i^c, \quad \boldsymbol{B}_i = B_i^c + t\,[-B_i^c, B_i^c]. \tag{56c}$$

The real part V of the output voltage is represented by the component x_{19} of the real vector x.

System S2: This is the so-called hybrid system of equations [11]–[13]. It has the same structure as system S1, i.e., the interval dependencies are again given by (55b), but now system (55a) has reduced size involving $n = 2m'$ equations. For the circuit studied $n = 16$,

$$\begin{aligned} p_1 &= G_1, \ p_2 = B_6, \ p_3 = G_4, \ p_4 = G_2 \\ p_5 &= R_4, \ p_4 = R_5 \ldots, \ p_7 = X_7, \ p_8 = X_8 \end{aligned} \tag{57}$$

and $G_i = 1/R_i$. Now V is given by x_3.

System S3: In this case, system (55a) is set up using nodal analysis and involves $n = 2n'$ equations. Now

$$a_{ij} = \sum \alpha_{ijk} p_k, \quad i, j = 1, \ldots, n \tag{58a}$$

where α_{ijk} can be 0, $+1$ or -1. For the circuit investigated $n = 8$ and

$$p_k = G_k, \ k = 1, \ldots, 5, \tag{58b}$$

$$p_k = B_k, \ k = 6, 7, 8. \tag{58c}$$

The output variable V is given by x_3.

We first compare the outer solutions for system S1 obtained by the present method M1 and Hansen's method for two values of the tolerance t. Although a better version [17] of Hansen's method has here been used than that employed in [12] and [13], the comparison of the results obtained by the two methods and listed in Table I shows that the present method outperforms Hansen's method. Indeed, the widths of the intervals V' obtained by M1 are smaller as compared to these corresponding to Hansen's method. These results confirm the theoretical prediction since, unlike Hansen's method, the present method accounts for the dependencies between the elements of the system of equations employed. At the same time, both methods require the same computing time t.

TABLE I
COMPARISON OF THE OUTER SOLUTIONS V' OBTAINED BY THE PRESENT METHOD M1 AND HANSEN'S METHOD

			Present method M1		Hansen's method	
t	System	n	V' (V)	τ (s)	V' (V)	τ (s)
0.05	S1	24	[0.1760, 0.4567]	0.06	[0.1115, 0.5310]	0.06
0.1	S1	24	[-0.0550, 0.6837]	0.06	[-0.2157, 0.8785]	0.06

TABLE II
DATA ON V' OBTAINED BY M1 FOR SYSTEMS S1, S2, S3

t	System	n	V'(V)	τ (s)
0.05	S1	24	[0.1760, 0.4567]	0.06
0.05	S2	16	[0.1787, 0.4477]	0.05
0.05	S3	8	[0.1421, 0.4891]	0.06
0.1	S1	24	[-0.0550, 0.6837]	0.06
0.1	S2	16	[-0.0316, 0.6350]	0.06
0.1	S3	8	[-0.4050, 1.0273]	0.06

In Table II, we present data on the outer solutions V' for two values of t, obtained by the present method M1 using systems S1, S2 and S3. It is seen that for both values of t the outer solution is the narrowest if system S2 is used while the computation time is practically the same.

B. Inner Solutions

The inner solutions V'' for V are obtained by Procedures 1 and 2 (method M2). Table III presents data on the width of V'', the total number of iterations N needed by both Procedure 1 and Procedure 2, and the execution time ι corresponding to the case where the system of equations used is S1.

The same intervals V'' are obtained when systems S2 and S3 are used.

To assess the effectiveness of the present method, we also found the inner solutions for V using the Monte-Carlo method. Table IV lists results corresponding to two values t and two values of the number of trials N_t.

The comparison of the results for V'' and ι in Tables III and IV clearly shows that the present method M2 is superior to the Monte-Carlo method. Indeed, method M2 provides wider intervals V'' and hence tighter approximations to the exact solutions V^*. At the same time, it requires less computation time although M2 was applied to system S1 ($n = 24$) while the Monte-Carlo method was implemented using the smaller system S3 ($n = 8$).

TABLE III
DATA ON THE INNER SOLUTIONS V'' OBTAINED BY THE PRESENT METHOD M2

t	System	n	V'' (V)	N	τ (s)
0.05	S1	24	[0.2122, 0.4340]	5	0.12
0.1	S1	24	[0.1196, 0.5630]	5	0.12

TABLE IV
DATA ON V'' OBTAINED BY THE MOTE-CARLO METHOD

t	System	n	N_t	V'' (V)	τ (s)
0.05	S3	8	1000	[0.2457, 0.3931]	1.43
0.05	S3	8	10000	[0.2354, 0.4002]	14.33
0.1	S3	8	1000	[0.1736, 0.4591]	1.43
0.1	S3	8	10000	[0.1651, 0.4900]	14.39

TABLE V
TWO-SIDED BOUNDS ON THE ENDPOINTS OF THE EXACT SOLUTION V^*

Methods	Bounds on $\underline{V}^*$	Bounds on $\overline{V}^*$	τ(s)
M1 + M2	[0.1787, 0.2122]	[0.4340, 0.4477]	0.17
MC + H	[0.1115, 0.2457]	[0.3931, 0.5310]	1.49

The inner and outer solutions obtained by methods M2 and M1, respectively, provide tight two-sided bounds on each endpoint $\underline{V}^*$ and $\overline{V}^*$ of the exact solution V^*. Table V lists data on these bounds (corresponding to $t = 0.05$, S2 and $N_t = 1000$) obtained by methods M1 and M2, on the one hand, and method Monte-Carlo (MC) and Hansen's method (H), on the other.

C. Exact Solution

Using method M3 (Procedure 4 and 5), the exact solution V^* has been obtained for all systems of equations when $t = 0.05$. Table VI lists results for the endpoints of V^*, the total number of iterations N (corresponding to the determination of both $\underline{V}^*$ and $\overline{V}^*$) and the required computing time ι.

If, however, $t = 0.1$ the exact solution has been reached only for S1 and S2—Table VII.

Method M3 is inapplicable with system S3 since for $t = 0.1$ none of the monotonicity conditions (47a)–(47b) is satisfied. However, if t is reduced to $t = 0.07$, the following result has been obtained (Table VIII).

In this case, method M3 reaches for five iterations only the exact lower endpoint $\underline{V}^*$. It cannot converge to the upper endpoint $\overline{V}^*$ since after 5 iterations conditions (47a)–(47b) remain unsatisfied for $l = 4$ and $l = 5$. However, as is seen in the table, the method provides a good two-sided bound on $\overline{V}^*$.

D. Analysis of the Numerical Results

The comparative analysis of the numerical results obtained can be summarized as follows. The conservatism of the outer solutions depends on the type of system of equations used. According to this criterion, the best results (narrowest solution intervals) are obtained when system S2 is employed. This is seemingly a valid conclusion for low- and medium-size circuits. However, for large-size TA problems, it is hoped that system S3 will be preferable. Further numerical experiments are needed to make a decision on this point. The exact solution is obtained with least computational effort if, again, system S2 is used. It should, however, be verified if this conclusion remains valid for large-scale circuits.

TABLE VI
DATA ON THE EXACT SOLUTION V^* OBTAINED BY THE PRESENT METHOD M3 FOR $t = 0.05$

t	System	n	V^*	N	τ(s)
0.05	S1	24	[0.2122, 0.4340]	4	0.22
	S2	16	[0.2122, 0.4340]	4	0.22
	S3	8	[0.2122, 0.4340]	6	032

TABLE VII
DATA ON V^* OBTAINED BY M3 FOR $t = 0.1$

t	System	n	V^*	N	τ(s)
0.1	S1	24	[0.1196, 0.5630]	7	0.44
	S2	16	[0.1196, 0.5630]	5	0.27

TABLE VIII
DATA ON V^* OBTAINED BY M3 FOR $t = 0.07$

t	System	n	$\underline{V}^*$	Bounds on $\overline{V}^*$	N	τ(s)
0.07	S3	8	0.1736	0.4709 ∓ 0.0241	10	0.55

It should be also stressed that at least for medium-size circuits the numerical efficiency of the present approach seems to be superior to that of the Monte Carlo method. It is expected that this conclusion remains valid also for large-scale circuits.

A final important remark is to be made. It should be noted that in all experiments the inner solution is in fact equal to the exact solution, i.e.,

$$x'' = x \tag{59}$$

(cf. Table III, Tables VI and VII). It is believed that the equality (59) is a generic characteristic for a large class of linear electric circuits. The derivation of a simple criterion establishing the validity of (59) and circumventing the verification of the costly monotonicity conditions (47a)–(47b) would substantially improve the overall efficiency of the new approach to solving the TA problems.

VII. CONCLUSION

A general framework (1a)–(1d) and (2) for treating worst-case TA problems has been considered. The linear algebraic system (1a)–(1d) has elements that are nonlinear functions of a set of parameters belonging to given intervals.

A simple method M1 for determining an outer solution to the TA problem considered has been suggested in Section III. It is based on Theorem 1 and reduces essentially to setting up and inverting an $n \times n$ real matrix and solving a system of n real linear equations (28), n being the size of the original system (1a)–(1d). The method is self-validating: it is applicable only if the solution to system (28) is positive.

In Section IV, an iterative method M2 (Procedures 1 and 2) is suggested which permits to determine an inner TA solution.

A more involved method M3 (Procedure 3 or Procedures 4 and 5) for computing the exact interval solution of the TA problem considered has been presented in Section V. It is based on the use of Theorem 2 and method M1. The latter is applied to compute tight enclosures D_{kl} of the derivatives dx_k/dp_l according to formula (44). If all derivative monotonicity conditions (47a)–(47b) hold, method M3 is guaranteed to determine the exact solution. In some cases, the method can provide the exact solution even if not all monotonicity conditions (47a)–(47b) are satisfied.

A numerical example has been solved in Section VI. It illustrates the applicability of the above methods to determining an outer, an inner and the exact solution to TA problems. The experimental results seem to show that at least for low- and medium-size circuits the present approach is superior to the Monte Carlo method as regards both accuracy and computing time. Further investigations are needed to decide whether the same conclusion is valid in the case of large-size circuits.

The three methods suggested can be extended to encompass TA problems related to transients in linear circuits. This generalization will be presented in a subsequent publication.

REFERENCES

[1] K. Reinschke, *Zuferlässichkeit von Systemen, Band 2* . Berlin, Germany: Verlag Technik., 1974, ch. 4, pp. 151–191.

[2] S. Skelboe, "True worst-case tolerance analysis of linear electric circuits by interval arithmetic," Inst. of Circuit Theory and Telecommunications. Tech. Univ. of Denmark, Lingby, Denmark, Rep. 1T, II, 1977.

[3] R. E. More, *Interval Analysis*. Englewood Cliffs, NG: Prentice-Hall, 1966.

[4] ——, *Methods and Applications of Interval Analysis*. Philadelphia, PA: SIAM, 1979.

[5] G. Alefeld and J. Herzberger, *Introduction to Interval Computation*. New York: Academic, 1983.

[6] L. V. Kolev, *Interval Methods for Circuit Analysis*. Singapore: World Scientific, 1993.

[7] ——, "Interval analysis for circuits," in *Encyclopedia of Electrical and Electronics Engineering*, J. Webster, Ed. Chichester, U.K.: Wiley, 1999, vol. 10, pp. 673–686.

[8] S. Skelboe, "True worst-case tolerance analysis of linear electric circuits by interval arithmetic." *IEEE Trans. Circuits Syst.*, vol. CAS-26. pp. 874–879, Oct. 1979.

[9] L. V. Kolev, B. M. Mladenov, and S. S. Vladov, "Interval mathematics algorithms for tolerance analysis," *IEEE Trans. Circuits Syst.*, vol. 35, pp. 967–975, Aug. 1988.

[10] L. V. Kolev and S. S. Vladov, "Linear circuits tolerance analysis via systems of linear interval equations," in *Proc. 6th Int. Symp. Networks, Systems, and Signal Processing*, June 28–July 1 1989, pp. 57–61.

[11] K. Okumura, "Tolerance analysis of linear electric circuits using interval analysis," in *Proc. Int. Conf. Interval Analysis and Applications*, Lafayette, IN, Feb.–Mar. 27–2, 1993.

[12] W. Tian, X.-T. Ling, and R.-W. Lin, "Novel methods for circuit worst-case tolerance analysis," *IEEE Trans. Circuits Syst. I*, vol. 43, pp. 272–278, Apr. 1996.

[13] M. W. Tian and C.-J. R. Shi, "Worst-case tolerance analysis of linear analog circuits using sensitivity bands," *IEEE Trans. Circuits Syst. I*, vol. 47, pp. 1138–1145, Aug. 2000.

[14] L. Kolev, "A new method for global solution of sysytems of nonlinear equations," *Rel. Comput.*, vol. 4, pp. 1–21, 1998.

[15] ——, "An improved method for global solution of nonlinear systems." *Rel. Comput.*, vol. 5, pp. 103–111, 1999.

[16] ——, "Automatic computation of a linear interval enclosure," *Rel. Comput.*, vol. 7, pp. 17–28, 2001.

[17] L. Kolev and I. Nenov, "Cheap and tight bounds on the solution set of perturbed systems," *Rel. Comput.*, vol. 7, pp. 399–408, 2001.

[18] S. Ning and R. Kearfort, "Comparison of some methods for solving linear interval equations," *SIAM J. Numer. Anal.*, vol. 34, pp. 1289–1305, 1997.

[19] L. Kolev, "Outer solution of linear systems whose elements are affine functions of interval parameters," *Rel. Comput.*, to be published.

Lubomir Kolev (M'00–SM'00) received the M.Sc. degree in electrical engineering from Moscow Power Institute, Moscow, U.S.S.R., in 1966, and Ph.D. and D.Sc. degrees from the Higher Institute for Mechnical and Electrical Engineering, Sofia, Bulgaria, in 1973, and 1989, respectively.

From 1966 to 1968, he was an Engineer at the Institute for Technical Cybernetics, Bulgarian Academy of Sciences, Sofia, Bulgaria. Since December 1968. he has been with the Electrical Engineering Department, Faculty of Automatica, Technical University of Sofia, Sofia, Bulgaria, where he is currently a Full Professor. From 1976 to 1978, he was a Senior Lecturer at the Higher School of Mohamedia, Rabat, Morocco, and from 1990 to 1993, as Associate Professor at the University of Zimbabwe, Harare, Zimbabwe. His research interests are mainly in the arias of linear and nonlinear circuits and systems analysis using modern mathematical tools. He has over 100 published papers and is the author of a monograph "*Interval Methods for Circuit Analysis*," (Singapore:Word Scientific, 1993).

130

PROCEEDINGS

ISTET'01

XI. International Symposium
on
Theoretical Electrical Engineering

"Blick auf dem Pöstlingberg, um 1840" OÖ Landesmuseum Inv.Nr.: OA L I 68/5

August 19 - 22, 2001
Linz, Austria

Edited by

K. Schlacher and R. Weigel (General Co-Chairmen)
A. Kugi and A. Springer (Technical Co-Chairmen)

Volume 2(2)

Communications I

Signal and Systems

Communictions II

A general interval method for tolerance analysis

Lubomir Kolev, Ivo Nenov

Abstract **- In this paper, an interval method for tolerance analysis of electric circuits is proposed. It is rather general and can be applied to solving tolerance problems both for linear and nonlinear circuits. The problems to be solved can be either of the deterministic, worst-case type (with independent or dependent parameters) or of the probabilistic type (when there are statistical dependencies between the parameters). Under a computationally verifiable condition, the method suggested guarantees to yield an interval solution that encloses the actual set of solutions to the specific tolerance problem considered. Numerical examples related to the worst-case tolerance problem for nonlinear dc circuits seem to indicate that the present method has attractive computational performance.**

Index Terms **- Tolerance analysis, interval analysis.**

I. INTRODUCTION

INTERVAL methods for solving various types of electric circuit tolerance analysis problems have been in existence for over twenty years [1]-[9]. The major part among these treats the worst-case (deterministic) tolerance problem for linear circuits. The linear tolerance problem in probabilistic setting is considered in [5], § 2.5. Papers [7] and [8] address the worst-case tolerance problem for nonlinear electric circuits. The known methods, however, differ considerably from one another depending on the class of circuits analyzed (linear or nonlinear), type of problem to be solved (in deterministic or probabilistic setting) and the type of problem formulation used (as a global constrained optimization problem or in the form of an interval linear or nonlinear system of equations).

In this paper, a general interval method for solving any of the known tolerance analysis problems for both linear and nonlinear circuits is suggested. It is based on a modification and generalization of a method proposed recently in [8]. Unlike [8] where the nonlinear worst-case tolerance problem was analyzed, now the system of equations describing the tolerance problem considered is in rather a general form

$$f(x,p)=0 \tag{1a}$$

$$p\in \boldsymbol{p} \tag{1b}$$

where f a n'-dimensional vector function, x is a n-dimensional output variable vector, p is a m-dimensional parameter vector and $\boldsymbol{p}$ is the corresponding interval vector (box). It is assumed that a pair (x^0, p^0) corresponding to the nominal solution 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) is the set

$$S_f(\boldsymbol{p}) := \{x : f(x,p)=0, \quad p\in \boldsymbol{p}\} \tag{2}$$

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 $S_f(\boldsymbol{p})$. In the present paper, the tolerance problem considered is equated to finding a bound $\boldsymbol{x}$ on S_f. A method for computing $\boldsymbol{x}$ is suggested which is based on an alternative linear interval enclosure of non-linear functions in a given box [10] - [12].

To simplify presentation, it is henceforth assumed that n' = n (number of equations equals number of output variables).

II. PRESENTATION OF THE METHOD

It is known that a continuous function $g(z_1, \ldots, z_q)$ can be enclosed in a box $\boldsymbol{z}$ by the following affine linear interval function

$$L_g(z)=\sum_{j=1}^{q} a_j z_j + \boldsymbol{b} \tag{3}$$

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

$$g(z)\in L_g(z), \quad z\in \boldsymbol{z} \tag{4}$$

Similar formulae are valid when g is an r-dimensional function. Now

$$L_g(z)=Az+\boldsymbol{b}, \quad z\in \boldsymbol{z} \tag{5}$$

(A is a real matrix and $\boldsymbol{b}$ is an interval vector) and for the new notation, property (4) is also valid. Constructive procedures for determining A and $\boldsymbol{b}$ are suggested in [10]-[12]. On account of (5) the linear interval enclosure of (1) in the box $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{p})$ will be

$$L_f(\boldsymbol{x} \ , \boldsymbol{p}) = A^x x + A^p p + \boldsymbol{b} \; x\in \boldsymbol{x} \ \ , p\in \boldsymbol{p} \tag{6}$$

The authors are with the Department of Theoretical Electrotechnics, Faculty of Automatica, Technical University of Sofia, 8 Kliment Ohridski Str., Sofia 1000, Bulgaria
(e-mail : lkolev@vmei.acad.bg).

In this section, a method for determining an outer bound $\boldsymbol{x}$ on the solution set $S_f(\boldsymbol{p})$ of (1) is presented. It consists of two stages: during the first stage, a "good" starting box $\boldsymbol{x}^s$ is determined; the second stage aims at improving $\boldsymbol{x}^s$ by making it narrower.

Stage 1. Let p^0 be the center of $\boldsymbol{p}$. First, the nominal solution x^0 is found by solving $f(x, p^0) = 0$. Next a narrow box $\boldsymbol{x}^0$ of width ε_0 centered at x^0 is introduced and, using(6), system (1) is enclosed in $\boldsymbol{z}^0$ by the linear interval form

$$L_f(\boldsymbol{x}^0, \boldsymbol{p}) = A_0^x x + A_0^p p + \boldsymbol{b}_0 \tag{7a}$$

$$x \in \boldsymbol{x}^0, \quad p \in \boldsymbol{p} \tag{7b}$$

Now, (7a) will be used as a linear approximation of (1) in a larger box $\boldsymbol{z}^1 = (\boldsymbol{x}^1 \ , \boldsymbol{p})$. Following [10] - [12] the component $\boldsymbol{x}^1$ of $\boldsymbol{z}^1$ is determined in the following way

$$\boldsymbol{x}^1 = -(A_0^x)^{-1} \boldsymbol{b}_0^{'} \tag{8a}$$

where

$$\boldsymbol{b}_0^{'} = A_0^p \boldsymbol{p} + \boldsymbol{b}_0 \tag{8b}$$

The first stage can be implemented in two different ways using the following two procedures.

Procedure 1. It is initiated by putting $\boldsymbol{x}^0 = \boldsymbol{x}^1$ and going back to (7).

Procedure 2. It starts as Procedure 1 by computing $\boldsymbol{x}^1$ using (8). At this point, $\boldsymbol{x}^1$ is renamed $\boldsymbol{x}'$and the new $\boldsymbol{x}^1$ is found by the union

$$\boldsymbol{x}^1 = \boldsymbol{x}' \cup \boldsymbol{x}^0 \tag{8c}$$

Next the iterations continue (as in the previous procedure) from (7) with $\boldsymbol{x}^0 = \boldsymbol{x}^1$.

It is assumed that Procedure 1 (Procedure 2) converges to a stationary interval vector (box) $\mathbf{x}^s$.

In practice, the respective procedure terminates whenever the distance between two successive iterations $\boldsymbol{x}^1$ and $\boldsymbol{x}^0$ becomes smaller than an accuracy ε_1. This approximate stationary box denoted as $\boldsymbol{x}^*$ differs, in general, from $\boldsymbol{x}^s$ and may be smaller. Therefore, $\boldsymbol{x}^s$ is constructed in the following way

$$\boldsymbol{x}^s = x_c^* + (1 + \varepsilon_2)[-R, R] \tag{9}$$

where R is the radius of $\boldsymbol{x}^*$ and $\varepsilon_2 \geq 0$.

Stage 2. After the box $\boldsymbol{x}^s$ has been determined by (9) we proceed to the second stage of the present method. Now we try to reduce $\boldsymbol{x}^s$ using the following procedure.

Procedure 3. Let $\boldsymbol{x}^0 = \boldsymbol{x}^s$ and construct the corresponding linear approximation of $f(x, p)$ in $(\boldsymbol{x}^0, \boldsymbol{p})$ using (7). By (8) find the corresponding box $\boldsymbol{x}'$and denote it $\boldsymbol{x}^{1}$. Next, a new box $\boldsymbol{x}^1$ is introduced by the intersection

$$\boldsymbol{x}^1 = \boldsymbol{x}' \cap \boldsymbol{x}^0 \tag{10}$$

As before, the iterative process is initiated by putting $\boldsymbol{x}^0 = \boldsymbol{x}^1$ and going back to (7). It is terminated when the distance between two successive boxes becomes smaller than an accuracy ε_3. The corresponding stationary box denoted $\boldsymbol{x}$ is, in fact, the outer solution of the corresponding tolerance problem described by (1).

The distance used in the stopping criterion in Procedures 1 to 3 has been chosen as

$$\mathrm{d} = \max\{|\mathrm{w}(\mathrm{x}_j^1)| - |\mathrm{w}(\mathrm{x}_j^0|\} \tag{11}$$

where w stands for width.

The second stage of the present method permits to computationally test its validity. Indeed, let $\boldsymbol{x}^{(k)}$ be the box obtained at the k th iteration of Procedure 3. If the condition

$$\boldsymbol{x}^{(k)} \subset \mathrm{int}(\boldsymbol{x}^s) \tag{12a}$$

(int denoting interior) is fulfilled for some k $\geq$ 1, then

$$S_f \subset \boldsymbol{x}^* \subset \boldsymbol{x} \tag{12b}$$

i.e. the outer solution thus found contains S_f and its interval hull. The proof of (12) (along with other theoretical aspects of this paper's method) will be published elsewhere.

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

a) algorithm A1 which is based on Procedures 1 and 3;

b) algorithm A2 which uses Procedures 2 and 3.

Experimental evidence seems to indicate that algorithm A2 requires less iterations than algorithm A1 to solve the tolerance analysis problems considered.

It should be noted that besides being more general, the present method differs from the method of [8] also in the way each iteration of Procedures 1 to 3 is carried out. In [8] this is done by approximately solving a linear interval system where all its elements are interval. The corresponding linear interval system in the new method is much simpler (only the right-hand side is interval) whose exact solution (within round-off errors) is computed by (8). This explains the better computational efficiency of the present method which is confirmed by the examples considered in the next section.

III. Numerical Examples

In this section, two worst-case tolerance examples illustrating the applicability and efficiency of the present method are given. The examples have been solved using algorithms A1 and A2. The algorithms were programmed using the algorithmic language C^{++}. The linear interval enclosures (7) were generated automatically by a procedure that implements the approach suggested in [12].

Example 1. In this example, the system of equations (1) is:

$$\begin{aligned} 10^{-9}(e^{38x_1} - 1) + p_1 x_1 - 1.6722x_2 + 0.6689x_3 - 8.0267 = 0 \\ 1.98 \cdot 10^{-9}(e^{38x_2} - 1) + 0.6622x_1 + p_2 x_2 + 0.6622x_3 + 4.0535 = 0 \\ 10^{-9}(e^{38x_3} - 1) + x_1 - x_2 + p_3 x_3 - 6 = 0 \end{aligned} \tag{13a}$$

$$\begin{aligned} p = (p_1, p_2, p_3) \in ([0.6020, 0.7358], \\ [1.2110, 1.4801], \quad [3.6, 4.4]) \end{aligned} \tag{13b}$$

and models a dc electric circuit containing a transistor, a diode and two resistors [9]. We consider the worst-case tolerance problem associated with (13): find an interval (outer) solution $\boldsymbol{x}$ to (13). In this example, the output vector is $x = (x_1, x_2, x_3)$ and the parameter vector is $p = (p_1, p_2, p_3)$. We chose p^0 as the center of p given in (13b). The corresponding nominal solution x^0 was found with accuracy $\varepsilon = 10^{-5}$ using a new nonlinear equations solver (implementing ideas from [11} and [12]):

$$x^0 = (0.5555, \quad -3.518, \quad 0.4685) \tag{14}$$

(the results in (14) are, however, given only to four decimal places).

Application of algorithms A1 and A2 with $\varepsilon_0 = \varepsilon_1 = \varepsilon_3 = 10^{-4}$ yielded the following results, respectively, for the intervalsolution of the tolerance problem considered:

$$x = ([0.5401 \quad 0.5682], \quad [-3.8926 \quad -3.1153], \quad [0.3483 \quad 0.5387] \tag{15}$$

$$x = ([0.5402 \quad 0.5680], \quad [-3.8910 \quad -3.1194], \quad [0.3473 \quad 0.5331] \tag{16}$$

For algorithm A1, $\varepsilon_2 = 0.05$ and the fulfillment of (12a) was achieved at $k = 1$ of Procedure 3. For algorithm A2 we chose $\varepsilon_2 = 0$ and nevertheless (12a) was satisfied already at $k = 2$ of Procedure 3. Thus, both bounds (15) and (16) are guaranteed to contain the solution set of (13). where

TABLE I

	Algorithm 1			*Algorithm 2*		
	Stage 1	*Stage 2*	*Total*	*Stage 1*	*Stage 2*	*Total*
N_1	27	8	35	13	7	20

The numbers of iterations corresponding to the two algorithms are listed in Table 1. It is seen that algorithm A2 requires less iterations as compared to algorithm A1.

The same example was solved in [8] by an algorithm similar in structure to algorithm A2 (however, as mentioned in the previous section, each iteration of both the first and second stage of the algorithm is associated with the solution of a corresponding linear interval system and requires more computation than algorithm A2). The following bound was obtained

$$x = ([0.5103, \ 0.5778], \quad [-4.352, \ -2.6756], \quad [0.3483, \ 0.5898]) \tag{17}$$

It is to be stressed that the bound (17) is more conservative as compared to (15) and (16) and at the same time takes more iterations to be reached: total number of iterations 166 (85 iterations for the first stage and 81 iterations for second stage).

Example 2. This example is a modification of Example 3.2 in [5]. The linear dc circuit of [5] is transformed into a nonlinear circuit by replacing the linear resistors r_3 and r_5 in Fig. 3.2 of [5] with nonlinear elements having respectively: a cubic characteristic

$$i_3 = 10^{-3}\,(2.5\,u^3 - 10.5\,u^2 + 11.8\,u) \tag{18a}$$

$$u = v_{12} - v_{13} \tag{18b}$$

and a diode characteristic

$$i_5 = 10^{-9}\left(e^{38u} - 1\right) \tag{19a}$$

with

$$u = v_{13} + e_5 - v_{14} \tag{19b}$$

All the remaining 9 linear resistors have a ±2% tolerance, i.e. their values lie within the interval [98, 102] Ω and define the parameter box p. The components of the output variable vector x are the branch currents i1, i2,..., i11 and the node voltages v12, v13,..., v16. Again, we consider the worst-case tolerance analysis problem associated with this modified circuit.

A numerical difficulty arose in finding the nominal solution x^0 for this example. It is due to the well-known overflow problem caused by the exponential diode nonlinearity. The overflow was overcome by scaling all variables by a factor of 0.01. Afterwards x^0 was found successfully with accuracy $\varepsilon = 10^{-5}$ using the new nonlinear equations solver. The initial box x^0, where x^0 was searched for, was chosen pretty wide: currents were set between (-1) and 1 A and node voltages between 1 and 100 V. A unique nominal solution is:

$$x^0 = (0.376, \ 0.391 \ 0.016, \ 0.272, \ 0.136, \ 0.360, \ -0.070, \ 0.337, \ 0.097, \ 0.167, \ 0.263, \ 62.36, \ 60.88, \ 33.72, \ 16.69, \ 26.35) \tag{20}$$

was located after 124 splits of x^0 and 3949 equation evaluations in the so-called generalized-interval form [12] (we report equation, not system evaluations since not all equations may be required per iteration). Using algorithm A2, the interval (outer) solution around the nominal solution x^0 was computed after 7 unions (stage 1) and 4 intersections (stage 2) for $\varepsilon_0 = \varepsilon_1 = \varepsilon_3 = 10^{-5}$ and $\varepsilon_2 = 0.05$. The left endpoint x^l and right endpoint x^r of the output variable vector are given below:

$$x^l = (0.366, \ 0.380, \ 0.07, \ 0.260, \ 0.123, \ 0.350, \ -0.079, \ 0.327, \ 0.089, \ 0.159, \ 0.254, \ 61.36, \ 59.84, \ 32.93, \ 15.97, \ 25.48) \tag{21}$$

$$x^r = (0.388, \ 0.404, \ 0.027, \ 0.279, \ 0.157, \ 0.369, \ -0.063, \ 0.350, \ 0.104, \ 0.176, \ 0.273, \ 63.28, \ 61.70, \ 34.74, \ 17.48, \ 27.22) \tag{22}$$

where as in (20) only four digits for each component are given.

IV. Conclusion

An interval method for tackling various classes of tolerance analysis problems (linear and nonlinear, deterministic with independent or dependent parameters, probabilistic) has been suggested. The method is rather general in its approach and equates the original tolerance problem to that of finding an outer interval solution x to the nonlinear system of equations (1) describing the tolerance problem considered. It is based on a recently suggested linear interval enclosure (5). If the computationally verifiable condition (12a) is satisfied, the method guarantees that the obtained solution x is really an outer solution, i.e. the inclusion (12b) is fulfilled.

A computer program implementing the method has been developed in a C^{++} environment. The numerical results obtained so far (including data not reported here) seem rather encouraging.

References

[1] K. Reinschke, *Zuferlässichkeit von Systemen, Band 2* (Chapter 4, pp. 151-191). Berlin : Verlag Technik, , 1974.

[2] S. Skelboe, "True worst-case tolerance analysis of linear electric circuits by interval arithmetic", *IEEE Trans. on Circuits and Systems*, vol. 26, 1979, pp. 874-879.

[3] L.V. Kolev, B.M. Mladenov and S.S. Vladov, "Interval mathematics algorithms for tolerance analysis", *IEEE Trans. on Circuits and Systems*, vol. 35, 1988, pp. 967-975.

[4] L.V. Kolev and S.S. Vladov, "Linear circuits tolerance analysis via systems of linear interval equations", *Proc. of the 6 th Intern. Symp. on Networks, Systems and Signal Processing*, June 28 – July 1, 1989, pp. 57-61.

[5] L.V. Kolev, *Interval Methods for Circuit Analysis*. Singapore, New Jersey, London, Hong Kong: World Scientific, 1993.

[6] K. Okumura, "Tolerance analysis of linear electric circuits using interval analysis", *Intern. Conf. on Interval Analysis and Applications*, 27 February – 2 March 1993, Lafayette, USA, 1993.

[7] W. Tian, X.-T. Ling and R.-W. Lin, "Novel methods for circuit worst-case tolerance analysis", *IEEE Trans. on Circuits and Systems - I*, vol. 43, 1996, pp. 272-278.

[8] 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.

[9] M.W. Tian and C.-J. R. Shi, "Worst-case tolerance analysis of linear analog circuits using sensitivity bands", *IEEE Trans. on Circuits and Systems*, vol. 47, 2000, pp. 1138-1145.

[10] Kolev, L.:A new method for global solution of nonlinear equations,*Reliable Computing* 4(2)(1998), pp.125-146.

[11] Kolev, L.: An improved method for global solution of nonlinear systems, *Reliable Computing* 5 2) (1999), pp. 103-111.

[12] Kolev, L.: Automatic Computation of a Linear Interval Enclosure, *Reliable Computing* 7 (1) (2001), pp. 17-28.

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

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

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

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

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

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

and

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

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 $\boldsymbol{z}^1 = (\boldsymbol{x}^1, \boldsymbol{p})$. The component $\boldsymbol{x}^1$ of $\boldsymbol{z}^1$ is determined in the following way. First, based on Theorem 2.3 we compute

$$x^1 = -(A_0^x)^{-1}\boldsymbol{b}_0', \tag{3.2a}$$

where

$$\boldsymbol{b}_0' = A_0^p\boldsymbol{p} + \boldsymbol{b}_0. \tag{3.2b}$$

Now the iterative procedure is started by putting $\boldsymbol{x}^0 = \boldsymbol{x}^1$ 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 $\boldsymbol{x}^1$ is determined at each iteration. We start as in Procedure 3.1 by computing $\boldsymbol{x}^1$ using (3.2). At this point $\boldsymbol{x}^1$ is renamed $\boldsymbol{x}'$ and the new $\boldsymbol{x}^1$ is found by taking the union

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

Next we let $\boldsymbol{x}^0 = \boldsymbol{x}^1$ 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* $\boldsymbol{p}$ *Procedure 3.1 (Procedure 3.2) is convergent to a stationary interval vector* $\boldsymbol{x}^s$ *having the property*

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

This assumption seems to be fulfilled most often in practice for relatively small boxes $\boldsymbol{p}$ 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 $L_f(\boldsymbol{x}^{(k)}, \boldsymbol{p})$ of $f(x, p)$ becomes better and the box $\boldsymbol{x}^{(k)}$ larger than $L_f(\boldsymbol{x}^{(k-1)}, \boldsymbol{p})$ and $\boldsymbol{x}^{(k-1)}$, respectively.

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

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

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

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

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

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

Now we let $\boldsymbol{x}^0 = \boldsymbol{x}^1$ 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*

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

*(*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

$$\begin{aligned} 10^{-9}(e^{38x_1} - 1) + p_1 x_1 - 1.6722x_2 + 0.6689x_3 - 8.0267 &= 0, \\ 1.98 \cdot 10^{-9}(e^{38x_2} - 1) + 0.6622x_1 + p_2 x_2 + 0.6622x_3 + 4.0535 &= 0, \qquad (4.6a) \\ 10^{-9}(e^{38x_3} - 1) + x_1 - x_2 + p_3 x_3 - 6 &= 0, \end{aligned}$$

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

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:

$$x = ([0.5401, 0.5682], [-3.8926, -3.1153], [0.3483, 0.5387]), \qquad (4.7)$$

$$x = ([0.5402, 0.5680], [-3.8910, -3.1194], [0.3473, 0.5331]). \qquad (4.8)$$

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

$$x = ([0.5103, 0.5778], [-4.3520, -2.6756], [0.3483, 0.5898]). \qquad (4.9)$$

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 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.

118

X. International Symposium on Theoretical Electrical Engineering

September 6 - 9, 1999

Magdeburg, Germany

Conference Proceedings

Hosted by

Otto-von-Guericke-University
Institute for Electronics, Signal Processing and Communications

Sponsored by

IEEE Networking the World DFG VDE ITG SACHSEN ANHALT

Edited by W. Mathis and T. Schindler (Otto-von-Guericke-University of Magdeburg, Germany)

X. International Symposium on Theoretical Electrical Engineering

September 6 - 9, 1999

Magdeburg, Germany

Conference Proceedings

Hosted by

Otto-von-Guericke-University
Institute for Electronics, Signal Processing and Communications

Sponsored by

IEEE Networking the World™ DFG VDE ITG SACHSEN ANHALT

Edited by W. Mathis and T. Schindler (Otto-von-Guericke-University of Magdeburg, Germany)
August 1999 © Otto-von-Guericke-University of Magdeburg, all rights reserved

116 Magdeburg

WORST-CASE TOLERANCE ANALYSIS OF NON-LINEAR CIRCUITS USING AN INTERVAL METHOD

Lubomir V. Kolev and Valeri M. Mladenov

Faculty of Automatics, Technical University of Sofia, 1756 Sofia, BULGARIA
tel. ++359 2 6362388; e-mail: valerim@vmei.acad.bg

Abstract

In this paper, a new iterative interval method applicable to both dc and ac worst-case tolerance analysis of non-linear circuits is presented. Besides being more general, it differs from other known methods in the way the linearized system arising at each iteration is set up and solved (approximately). Another distinction is the fact that now the initial linear tolerance problem (starting the iterations) corresponds to a circuit whose non-linear resistor characteristics are linearized around the corresponding nominal dc operating point of the original non-linear circuit studied.

1 Introduction

Interval methods have proved a reliable tool for solving the worst-case tolerance analysis problem for linear circuits [1]-[6]. The known methods are based on the following two basic approaches: (i) determining the range of a non-linear function relating the output variable to the interval parameters [1]-[4,Ch.2] and (ii) solving a corresponding system of linear equations having interval coefficients [3], [4,Ch.3], [5], [6]. The latter approach is more general in comparison to the former one since it permits the simultaneous determination of all output variables tolerances one is interested in. An additional advantage is the possibility to find approximate solutions in a much easier way than determination of the exact worst-case tolerances.

While the linear circuit worst-case tolerance analysis problem has been largely covered in the literature (e.g. [1]-[6]), its counterpart for non-linear circuits has drawn little attention: the only paper which touches upon the nonlinear tolerance problem seems to be [6], Sect. IV. B. An iterative interval method is suggested there for worst-case tolerance analysis of dc non-linear circuits. At each iteration, an associate linear dc tolerance problem is solved approximately, using an appropriate interval method. The approximate solution thus obtained must contain the tolerance solution sought for all iterations. Therefore, the iterations are initialized by selecting an initial interval vector $X^{(0)}$ which must contain the tolerance solution vector sought. The selection of a good initial vector $X^{(0)}$ remains an open problem. Indeed, if $X^{(0)}$ is chosen too large the iteration process will take too long to converge; conversely, if $X^{(0)}$ is too small it might not contain the tolerance solution. In this paper, a new iterative interval method applicable to both dc and ac worst-case tolerance analysis of non-linear circuits will be presented. Besides being more general, it differs from the method of [6] in the way the linearized system arising at each iteration is set up and solved (approximately). Another distinction is the fact that unlike [6] now the initial linear tolerance problem (starting the iterations) corresponds to a circuit whose non-linear resistor characteristics are linearized around the corresponding nominal dc operating point of the original non-linear circuit studied.

2 Problem statement

To simplify presentation, only the dc tolerance problem is presented here for the special case where the circuit contains only independent current-controlled non-linear resistors. Furthermore, it is assumed that each function $v_\rho = f_\rho(i_\rho)$ describing the v-i characteristic of the p-th .ion-linear element is known exactly (a more general setting, when $v_\rho = f_\rho(i_\rho) + b, b \in B$ where B is a given interval, can be easily encompassed by the new method). Following [4]-p.118, the following system can be set up

$$f_\rho(i_\rho) + r_\rho i_\rho = u_\rho,\ \rho = 1,\dots,m \qquad (1a)$$

$$\sum_{j=1}^{m} \alpha_{kj} i_j = 0,\ k = 1,\dots,n' \qquad (1b)$$

for a circuit having $n' + 1$ nodes, m branches with m linear resistors, m non-linear resistors, and m independent voltage sources u_ρ.

Without loss of generality, it is assumed that only r_ρ (but not u_ρ) have tolerances, i.e. $r_\rho \in R_\rho = \left[\underline{r_\rho}, \overline{r_\rho}\right]$. Finally, system (1) will be written in vector form as

$$\Psi(x) = f(x) + Ax - b = 0, x \in R^n \quad (2a)$$

$$A \in A^I. \qquad (2b)$$

$$f_i(x) = f_i(x_i), \tag{2c}$$

$$A = \{a_{ij}\}, A^I = \{A_{ij}\}, a_{ij} \in A_{ij} \tag{2d}$$

where A_{ij} are independent intervals. The tolerance problem is to find an interval vector X^* which contains (as tightly as possible) the solution set S of (2)

$$S = \{x: \Psi(x) = 0, a_{ij} \in A_{ij}\} \tag{3}$$

The above formulation can be extended to more general dc non-linear circuits as well as to considering various ac tolerance problems in non-linear circuits.

3 The new method

3.1 Algorithm of the method.

Once again, for brevity, only the dc tolerance problem will be covered here.

Let a_{ij}^c and A^c denote the nominal values of the respective quantities. Using some interval methods for dc non-linear analysis [4], [7], the corresponding nominal solution x^c can be found. Now each $f_i(x_i)$ is linearized around x_i^c as

$$f_i(x_i) = f_i(x_i^c) + c_i(x_i - x_i^c),\ c_i = \frac{\partial f_i(x_i^c)}{\partial x_i} \tag{4}$$

and the diagonal matrix C with non-zero entries c_i and vector d with elements $d_i = f_i(x_i^c) - c_i x_i^c$, $i = 1,2,\ldots,n$ are formed. At this stage, the following linear dc tolerance analysis problem is solved

$$Cx + Ax = b - d,\ A \in A^I \tag{5}$$

Let $X^{(0)}$ denote an approximate interval solution of (5) having the property to contain the exact interval solution X^* of (5). At this point, each of $f_i(x_i)$ is represented within the interval $X_i^{(0)}$ by the following linear interval inclusion [7]:

$$f_i(x) = \alpha_i^{(0)} x_i + B_i^{(0)}, x_i \in X_i^{(0)} \tag{6}$$

where $\alpha^{(0)}_i$ is a real number (the slope of f_i within $X_i^{(0)}$) while $B^{(0)}_i$ is a corresponding interval. On introducing the diagonal matrix $C^{(0)}$ whose entries are $\alpha_j^{(0)}$ and the interval vector $B^{(0)}$, the following linear dc tolerance analysis problem is next set up and solved:

$$C^{(0)}x + Ax = b - b^*,\ A \in A^I,\ b^* \in B^{(0)} \tag{7}$$

Let the approximate solution of (7) be $X^{(1)}$. At this stage, we form the union

$$X^{(2)} = X^{(0)} \cup X^{(1)} \tag{7'}$$

and an iterative procedure (Procedure 1) starts from (6) and (7) using (7'). Let $X^{(2\nu+1)}$ be the solution of the corresponding problem (7) at the ν-th iteration. The iterative process is stopped when $X^{(2\nu+1)} \subseteq X^{(2\nu)}$. Now a second procedure (Procedure 2) is started which aims at reducing the width of the last interval vector $X^{(2\nu)}$ obtained at the end of Procedure 1. Let $X^{(0)} = X^{(2\nu)}$. The present procedure has the same structure as Procedure 1. The only difference is that now the union operation in (7') is replaced with the intersection operation

$$X^{(2)} = X^{(0)} \cap X^{(1)} \tag{7''}$$

The process stops when $X^{(2\nu)} \subseteq X^{(2\nu+1)}$. The solution of the original non-linear dc tolerance problem is then given by the result of Procedure 2

The method is easily generalized to the ac case. The only difference is that now (5) and (7) are in complex form.

3.2 Solving linear tolerance problems

Since the iterative method involves the repetitive solution of system (7), its over all efficiency depends strongly on how efficiently (7) is solved at each iteration. In [6], the corresponding linear tolerance problem is solved approximately using an improved version of Hansen's method [8]. The same method could be also used for solving (7). A simpler method is, however, suggested here

System (7) will be rewritten in the form

$$(A^c + \Delta)(x^c + u) = b^c + \delta, \Delta \in \Delta^I, \delta \in \delta^I \tag{8}$$

where the superscript c means centre and Δ, δ and u are the corresponding centred variables. On account of (1a) it is seen that Δ is a diagonal matrix. The following notations are now introduced: $\bar{x}$ - diagonal matrix whose diagonal is x^c, $\bar{\Delta}$ - vector whose elements are Δ_{ii} of Δ, $\tilde{R}_a = (\tilde{R}_{a,1}, \ldots, \tilde{R}_{a,n})$ - radius of $\tilde{\Delta}$, i.e. $\tilde{R}_{a,i} = 0.5(\overline{\tilde{\Delta}_i} - \underline{\tilde{\Delta}_i})$, R_b - radius of b, R - the radius of the unknown (approximate) solution U, $C = (A^c)^{-1}$, $\tilde{C} = C\bar{x}$, R_a - diagonal matrix whose diagonal is formed by the elements of the vector $\tilde{R}_a$. It can be proved that R is the solution of the following real linear system

$$(E - |C|R_a)R = |\tilde{C}|(\tilde{R}_a + R_b) \tag{9}$$

where E is the unit matrix. Finally, the approximate solution X of (8) is the interval vector in centred form

$$X = x^c + [-R, R] \tag{10}$$

It can be shown that X contains the exact interval solution X^* to (8).

4 A numerical example

To illustrate the applicability of the new method, an electrical circuit contain transistor and diode (Example 6.2 in Reference 4) is considered. Vector-function $f(x)=(f_1(x_1)\ f_2(x_2)\ f_3(x_3))^T$, matrix A and vector b from (2a) are given by

$$f_1(x_1) = 10^{-9}(e^{38x_1} - 1)$$

$$f_2(x_2) = 1.98.10^{-9}(e^{38x_2} - 1)$$

$$f_3(x_3) = 10^{-9}(e^{38x_3} - 1)$$

$$A = \begin{bmatrix} 0.6689 & -1.6722 & 0.6689 \\ 0.6622 & 1.3455 & 0.6622 \\ 1 & -1 & 4 \end{bmatrix}$$

$$b = \begin{bmatrix} 8.0267 \\ -4.0535 \\ 6 \end{bmatrix}$$

The tolerances in the diagonal elements of matrix A are chosen to be 20%. Using the algorithm described in section 3.1 and the simple method for solving the linear interval tolerance problem from 3.2, the following interval vector X^*

$$X^* = \begin{bmatrix} 0.5103 & 0.5778 \\ -4.3520 & -2.6756 \\ 0.3483 & 0.4898 \end{bmatrix}$$

is obtained The number of iterations for Procedures 1 and 2 are respectively 85 and 81 while the total number of iterations is 166. The interval vector X^* contains (as tightly as possible) the solution set S of (2).

5 Conclusion

We present a new iterative interval method applicable to both dc and ac worst-case tolerance analysis of non-linear circuits. The method is based on description (2) where only part of the diagonal elements of the matrix are intervals. A main feature of the method is the new linearization technique (6) used at each iteration of the computation process. The fact that only part of the diagonal elements of matrix A are interval allows to develop a simple method for approximate solution of the arising linear interval system (7). The numerical efficiency of the method proposed can be improved if sparse matrix techniques are used in solving system (9). This improvement will be substantial for large-scale circuits when the size of the corresponding system (9) is rather high.

6 Acknowledgments

This work is supported by the National Research Fund of Bulgaria under Contract 554/95.

7 References

[1] S. Skelboe, "True worst-case tolerance analysis of linear electrical circuits by interval arithmetic", *IEEE Trans. on Circuits and Systems*, vol. 26, 1979, pp. 874-879.

[2] L.V. Kolev, V.M. Mladenov, S.S. Vladov, "Interval Mathematics Algorithms for Tolerance Analysis", *IEEE Trans. on Circuits and Systems*, vol. 35, N 8, 1988, pp. 967-975.

[3] L.V. Kolev , S.S. Vladov, "Linear Circuits Tolerance Analysis via Systems of Linear Interval Equations", Proc. of the *6th Intern. Symp. On Networks, Systems and Signal Processing*, June 28-July 1, 1989, pp. 57-61.

[4] L.V. Kolev, Interval methods for circuit analysis. Singapore, New Jersey: World Scientific 1993.

[5] K. Okumura, "Tolerance analysis of linear electric circuits using interval analysis", *Int. Conf. on Interval Analysis and Applications*, 27 Feb.-2 March 1993, Lafayette, USA, 1993

[6] W. Tian, X.-T. Ling and R.-W. Liu, "Novel methods for circuit worst-case tolerance analysis", *IEEE Trans. on Circuits and Systems - I*, vol. 43, 1996, pp. 272-278.

[7] L. Kolev, "An Efficient Method for Global Analysis of Nonlinear Resistive Circuits", *Int. J. Cir. Theor. and Appl.*, vol. 26, 1988, pp. 81-92.

[8] E. Hansen, "Bounding the Solution of Interval Linear Equations", *SIAM J. Numer. Analysis*, vol. 29, 1992, pp. 1493-1503.

X. International Symposium on Theoretical Electrical Engineering

September 6 - 9, 1999

Magdeburg, Germany

Conference Proceedings

International Steering Committee

Hosted by

Otto-von-Guericke-University
Institute for Electronics, Signal Processing and Communications

Sponsored by

IEEE Networking the World

DFG

VDE ITG

SACHSEN ANHALT

Edited by W. Mathis and T. Schindler (Otto-von-Guericke-University of Magdeburg, Germany)

A NEW INTERVAL METHOD FOR GLOBAL OPTIMIZATION

L. V. Kolev
Technical University of Sofia, Faculty of Automatica,Bulgaria, Sofia 1756,
e-mail: lkolev@vmei.acad.bg

Abstract

Interval methods are iterative methods capable of solving the general nonlinear programming problem globally, providing infallible bounds both on the optimum (optima) and the corresponding solution coordinates. However, their computational complexity grows rapidly with the dimension of the problem and the size of the search domain. In this paper, a new interval method is suggested which seems to have improved numerical efficiency. It is based on the use of a new interval linearization of the nonlinear functions involved. Two algorithms for computing it are presented. The new optimization method incorporates seven procedures that are implemented by way of the new interval linearization. A numerical example illustrating the method suggested is also given

1. Introduction

Applying interval analysis techniques to solving various global optimization problems has been a major deterministic approach over the last decades [1] - [5]

In this paper, a new interval method is suggested for global solution of the following constraint optimization problem.

Minimize

$$\varphi_0(x) \tag{1a}$$

subject to the constraints

$$\varphi_i(x) \le 0, \quad i = 1,2,\dots,r \tag{1b}$$

$$x \in X^{(0)} \subset R^n \tag{1c}$$

where x is a n-dimensional vector and $X^{(0)}$ is a given initial search region (a box). The functions in (1a) and (1b) are assumed to be only continuous in $X^{(0)}$.

It should be stressed that problem (1) presents in an equivalent form the general nonlinear programming problem which also involves equality constraints

$$\varphi_i(x) = 0, \quad i = r+1,\dots,r_1 \tag{1d}$$

Indeed, each equality constraint can be represented equivalently by two inequality constraints.

Let $f(x)$, $x \in X \subset R^p$, $p \ge n$ be a continuously differentiable function. Known interval methods for solving (1) are iterative and are based on the following interval linearization of f in X:

$$L(X) = f(x^c) + \sum_{i=1}^{p} G_i(X)(X_i - x_i^c) \tag{2}$$

where x^c is the centre of X and $G_i(X)$ is either the interval extension of the derivative $g_i(x) = \partial f / \partial x_i$ or the corresponding interval slope [3], [5], [9]. In contrast, the present method appeals to a new interval linearization of f in the form:

$$L(X) = \sum_{i=1}^{p} a_i x_i + B, \quad x_i \in X_i \tag{3}$$

where a_i are real numbers and only B is an interval. The use of (3) in the computational scheme of the new global optimization method leads to improved performance as compared to the other known methods since it permits a better (tighter) enclosure of the original nonlinear functions. Another advantage of the alternative form (3) resides in the fact that it is applicable to nonlinear functions that are only continunuos or even discontinuous.

2. New interval linearization of a nonlinear function

Two algorithms for determining the new interval linearization (3) will be presented in this section.

2.1. First algorithm

Let $f(x)$ be a multivariate function $f: D \subset R^p \to R$. The transformation of a nonlinear function $f(x)$, $x \in X$, to the new linear interval form (3) can be done following the approach suggested in [6]-[8]. If $f(x)$ is in separable form, α_i and B can be determined by a procedure given in [6]. For an arbitrary function $f(x)$ (which is continuous or even discontinuous) (3) can be evaluated by following the approach suggested in [7], [8]. First, f is transformed into a system of equations of the so-called semiseparable form [7] by introducing a certain number m of auxiliary variables. Afterwards, each new semiseparable equation is easily transformed into form (3). Thus, a system of m+1 linear interval equations is generated. Finally, the auxiliary variables are eliminated from the latter linear system to yield the linear form (3) corresponding to the original function f.

Thus, the approach outlined above involves the following steps.

Step 1. Transformation to semiseparable form.

A function f is called to be in semiseparable form if [7]

$$f(x) = \sum_{j=1}^{n} f_j(x_j) + \sum_{k=1}^{n} \sum_{\substack{l=1 \\ k \neq l}}^{n} \alpha_{kl} x_k x_l \quad . \qquad (4)$$

(some of these terms may be missing). The transformation of an arbitrary function to a set of functions of the semiseparable form (4) can be done by the approach suggested in [7]. This possibility will be illustrated by the following example

Example 1. Let

$$f(x) = (x_1^2 + x_2^2 - 1)(x_3^2 - x_2) \qquad (5a)$$

with $x \in X$ and

$$X_i = [0,1], \quad i = 1,2,3 \qquad (5b)$$

The problem is to convert (5a) into a set of semiseparable functions. With this in mind, we introduce two auxiliary variables x_4 and x_5 to get

$$\begin{aligned} f(x) &= x_4 x_5 \\ x_4 &= (x_1^2 + x_2^2 - 1) \qquad (6) \\ x_5 &= (x_3^2 - x_2) \end{aligned}$$

Now all expressions in (6) are in semiseparable form. We need also to evaluate the corresponding ranges

$$X_4 = [-1,1] \qquad (7a)$$

$$X_5 = [-1,1] \qquad (7b)$$

Step 2. Enclosing the auxiliary variable expressions.

The above example shows that in the general case $f(x)$ will be transformed by the introduction of a given number m of auxiliary variables into a set of $m+1$ functions of semiseparable form

$$f_i(y), \quad i = 0,1,\ldots,m \qquad (8)$$

where $f_0 = f$, $f_i = x_{n+i}$, $i = 1,2,\ldots,m$, $y \in R^{n+m}$ is the "augmented" vector of variables. For Example 1, system (8) is given by (6) with $y = (x_1, x_2, \ldots, x_5)$, $m = 2$, $f_i = x_{n+i}$, $i = 1,2$.

At this point, each function f_i, $i = 0,1,\ldots,m$ is enclosed by a corresponding linear enclosure (3). Thus, for the example considered

$$f_0 = f \in a_4 x_4 + a_5 x_5 + B_0, \quad x_4 \in X_4, x_5 \in X_5 \qquad (9)$$

$$f_1 = x_4 \in a_1 x_1 + a_2 x_2 + B_1, \quad x_1 \in X_1, x_2 \in X_2 \qquad (10)$$

$$f_2 = x_5 \in -x_2 + a_3 x_3 + B_2, \quad x_3 \in X_3 \qquad (11)$$

In the general case

$$f_0 \in \sum_{j=1}^{n+m} a_{0j} x_j + B_0 \qquad (12)$$

$$x_{n+i} \in \sum_{j=1}^{n} a_{ij} x_j + B_i, \quad i = 1,2,\ldots,m \qquad (13)$$

$$F_i = \sum_{j=1}^{n} a_j X_j + B_i \qquad (13')$$

Step 3. Eliminating the auxiliary variables

In the final step, the auxiliary variables are eliminated using (12) and (13). This possibility will be illustrated by way of Example 1. Substituting (10) and (11) into (9), we get

$$\begin{aligned} f_0 \in\ & a_1 a_4 x_1 + (a_2 a_4 - a_5) x_2 + a_3 a_5 x_3 + \\ & + B_0 + a_4 B_1 + a_5 B_2 \end{aligned} \qquad (14)$$

Finally, $f(x)$ given by (5a) has been enclosed in the box X with sides (5b) by the linear interval expression

$$L(x) = a_1' x_1 + a_2' x_2 + a_3' x_3 + B \qquad (15a)$$

with

$$a_1' = a_1 a_4, \quad a_2' = a_2 a_4 - a_5, \quad a_3' = a_3 a_5 \qquad (15b)$$

$$B = B_0 + a_4 B_1 + a_5 B_2 \qquad (15c)$$

2.2. Second algorithm

Now, the only assumption on f is that f is a factorable function [17], i.e. composed of the four arithmetic operations (+, -, *, /) and the unary operations (sin, exp, log, sqrt, abs, etc.). However, to simplify the presentation, the linear interval enclosure (3) will be first determined for the class of polynomial functions. Later on, the approach adopted for polynomial functions will be extended to arbitrary factoriable functions.

A. Polynomial Functions

In this subsection, an algorithm will be suggested for determining the linear interval enclosure (3) for the special case where f is a polynomial function. This algorithm is based on the notion of a generalized interval [3]. We shall introduce a slightly different generalized representation in the following manner.

Let $X = (X_1, \ldots, X_n)$ and

$$X_i = c_i + V_i, \quad i = 1,\ldots n \qquad (16a)$$

where c_i is the centre of X_i and V_i is a symmetrical interval

$$V_i = [-R_i, R_i] \qquad (16b)$$

R_i being the radius of X_i, i.e.

$$R_i = \bar{x}_i - x_i \qquad (16c)$$

Definition 1. A generalized interval $\tilde{X}$ is defined as the affine function

$$\tilde{X} = \sum_{i=1}^{n} \alpha_i V_i + c_x + V_x, \quad V_x = [-R_x, R_x] \qquad (17)$$

where α_i and c_x are real numbers while V_i and V_x are centred "ordinary" intervals.

Using $\tilde{X}$, any "ordinary" interval can be represented by an appropriate choice of the terms of $\tilde{X}$. Indeed, letting $\alpha_i = 0$ [illegible] and $V_x = 0$ we get $\tilde{X} = V_j +$ [illegible] X_j where X_j is the jth ordinary interval.

Now we shall define the operations of addition and multiplication of generalized intervals (G intervals). Let

$$\tilde{Y} = \sum_{i=1}^{n} \beta_i V_i + c_y + V_y, \quad V_y = [-R_y, R_y] \qquad (3.3)$$

be a G interval. Then we have the following rules.

Addition. Let $\tilde{X}$ and $\tilde{Y}$ be two G intervals given by (17) and (18). The sum of $\tilde{X}$ and $\tilde{Y}$ denoted as $\tilde{X} + \tilde{Y}$ is another G interval $\tilde{Z}$:

$$\tilde{Z} = \sum_{i=1}^{n} \gamma_i V_i + c_z + V_z, \quad V_z = [-R_z, R_z] \qquad (19a)$$

and

$$\gamma_i = \alpha_i + \beta_i, \quad i = 1,\ldots n \qquad (19b)$$

$$c_z = c_x + c_y, \quad R_z = R_x + R_y \qquad (19c)$$

Multiplication. The product $\tilde{X} \cdot \tilde{Y}$ of two G intervals $\tilde{X}$ and $\tilde{Y}$ is another G interval $\tilde{Z}$ if:

$$\gamma_i = c_x\beta_i + c_y\alpha_i, \quad i = 1,\dots,n \tag{20a}$$

$$c_z = c_x c_y + \frac{1}{2}\sum_{i=1}^{n} \alpha_i\beta_i R_i^2 \tag{20b}$$

$$\begin{aligned} R_z = {} & R_x R_y + |c_x| R_y + |c_y| R_x + \sum_{i,j=1}^{n} |\alpha_i\beta_j| R_i R_j + \\ & + R_x \sum_{j=1}^{n} |\beta_j| R_j + R_y \sum_{i=1}^{n} |\alpha_i| R_i + \frac{1}{2}\sum_{i=1}^{n} |\alpha_i\beta_i| R_i^2 \end{aligned} \tag{20c}$$

(R_i is the radius of X_i).

The proof of (19) and (20) is based on elementary properties of adding, multiplying and centring ordinary intervals and is therefore omitted.

Using the above two operations, any intermediate or final result in evaluating the interval extension of a polynomial function can be represented as a generalized interval. Indeed, multiplying an ordinary or G interval by a constant c is a special case of the multiplication $\tilde{Z} = \tilde{X} \cdot \tilde{Y}$. If $\tilde{X} = c = -1$, we get $\tilde{Z} = -\tilde{Y}$ and the rules for the operation of subtraction follow immediately.

Subtraction. For $\tilde{Z} = \tilde{X} - \tilde{Y}$

$$\gamma_i = \alpha_i - \beta_i, \quad i = 1,\dots,n, \quad c_z = c_x - c_y \tag{21a}$$

$$R_z = R_x + R_y \tag{21b}$$

Example 2. Let $x = (x_1, x_2)$ and

$$f(x) = (x_1 - 2x_2)x_1, \quad x_i \in X_i, \quad i = 1,2 \tag{22)}$$

Find the linear interval enclosure (3) corresponding to (22).

We first introduce the G intervals

$\tilde{X}_1 = \alpha_1 V_1 + 0.V_2 + c_1, \quad \tilde{X}_2 = 0.V_1 + \beta_2 V_2 + c_2$

with $\alpha_1 = 1$, $\beta_2 = 2$. Then we compute

$$\tilde{Y} = \tilde{X}_1 - \tilde{X}_2 \tag{23}$$

using (21). The final result is obtained as the product

$$F(X) = \tilde{Z} = \tilde{Y}\tilde{X}_1 = \gamma_1 V_1 + \gamma_2 V_2 + c_z - V_z \tag{24}$$

computed by (20).

The linear form (24) represents equivalently (for the example considered) the linear form (3). Indeed, from (16)

$$V_i = X_i - c_i, \quad i = 1,2 \tag{25}$$

and substituting (25) into (24) we get

$$F(X) = a_1 X_1 + a_2 X_2 + B \tag{26a}$$

with

$$a_1 = \gamma_1, \quad a_2 = \gamma_2, \quad B = c_z - \gamma_1 c_1 - \gamma_2 c_2 + V \tag{26b}$$

B. Factorable functions

The approach suggested in Section 2.2 A will be now extended to arbitrary factorable functions [9]. A function $f: D \subseteq R^n \to R$ is a factorable function (f.f.) if and only if it can be represented by an expression $f(x)$ which is the last element in a finite sequence $(f_i(x))$ of expressions. For the case of $f \in C^1(D)$, the list of admissible expressions is given in [9]. An approach to treating non-differentiable functions is considered in [3], Ch. 14. An alternative idea is suggested in [5], Ch. 6. Here, we shall consider the general situation when $f \in C^1(D)$, $f \in C^n(D)$ or even when some expressions $f_i(x)$ may be discontinuous functions.

Let W denote the set of all building expressions for a given f.f. For our purposes, it is convenient to divide W into two parts W_1 and W_2 such that

$$W = W_1 \cup W_2$$

The set W_1 is made up of expressions which are used to construct a multivariate polynomial function. The set W_2 contains the following three groups of expressions:

(i) the reciprocal value operation

$$f_i(x) = 1/x, \quad x \in X \in R, \quad 0 \notin X \tag{27}$$

(ii) the set Φ containing standard functions Φ_i to be found in high level programming languages, i.e.

$$\Phi = \{\mathrm{sqrt}(\cdot), \exp(.), \ln(\cdot), \sin(\cdot), \mathrm{abs}(\cdot), \dots,\} \tag{28}$$

The set Φ can be enlarged as appropriate. For instance, it may include various discontinuous functions. A typical example of such functions is the unit step function 1(x) defined as:

$$1(x) = \begin{cases} 1, & \text{for} \quad x > 0 \\ 0, & \text{for} \quad x \le 0 \end{cases} \tag{29}$$

(iii) unary functions $f_i: R \to R$ which may include rational and irrational parts.

The main characteristic of all unary functions $f_i \in W_2$ is the fact that they allow easy computation of a corresponding linear interval enclosure (3).

Now we are in a position to present an algorithm for computing (3) for the case of an arbitrary f.f. We assume that the sequence $f_i(x)$ representing the function at hand f has already been chosen. To simplify the presentation, we assume additionally that the first k expressions $f_i \in W_1$ while the remaining $f_i \in W_2$ (in the general case, the appearance of $f_i \in W_1$ and $f_i \in W_2$ in the sequence $(f_i(x))$ may have a more complex pattern).

On account of the results obtained in subsection 2.2 A it is clear that the linear interval enclosure $F_k(X)$ corresponding to the last expression $f_k(x)$ belonging to W_1 is given by the G interval

$$\tilde{F}_k = \sum_{j=1}^{n} \alpha_{kj} X_j + B_k \tag{30}$$

which has been computed recursively using G intervals $\tilde{F}_j$ corresponding to expressions f_j with $j < k$.

Now consider the first function $f_i \in W_2$, i.e. the function f_{k+1}. According to the construction of the sequence $(f_i(x))$

$$f_{k+1}(x) = f_{k+1}(f_k(x)) \tag{31a}$$

$$f_k(x) \in \tilde{F}_k(X) \tag{31b}$$

Since f_{k+1} is a unary function, it can be enclosed by the interval function

$$F_{k+1} = a_{k+1}\tilde{F}_k + B'_{k+1} \tag{32}$$

where a_{k+1} and B_{k+1} are determined in one way or another (depending on whether f_{k+1} is a C^n, C^0 or discontinuous function). Substituting (30) into (32), it is seen that F_{k+1} can be represented as a G interval

$$\tilde{F}_{k+1} = \sum_{j=1}^{n} \alpha_{k+1,j} X_j + B_{k+1} \qquad (33a)$$

with

$$\alpha_{k+1,j} = a_{k+1}\alpha_{kj}, \quad B_{k+1} = a_{k+1}B'_{k+1} + B_k \qquad (33b)$$

Since (31) remains valid if the index k+1 is replaced with $i > k+1$, it is clear that the recursive formula (33) also holds for $i > k$+1. Thus, it has been shown that the factoriable function can be enclosed in X by a G interval

$$\tilde{F} = \sum_{j=1}^{n} \alpha_j X_j + B \qquad (34)$$

whose coefficients α_j and additive term B can be determined in a recursive way using only the binary operations (19), (20), (21) (addition, multiplication and subtracting of two G intervals) and (33) (multiplication of a G interval by a scalar).

Example 3 [9]. Find the interval enclosure for

$$f(x) = x_1^2 / \exp(x_2), \quad X_1 = [1, 1.5], \; X_2 = [2, 2.5] (35)$$

The function $f(x)$ can be defined by the corresponding element $f_3(x)$ from the following sequence

$$f_1(x) = x_1^2 \qquad (36a)$$

$$f_2(x) = 1/\exp(x_2) \qquad (36b)$$

$$f_3(x) = f_1(x) f_2(x_2) \qquad (36c)$$

Applying the above algorithm we first have to compute the enclosure for (36a):

$$F_1(X) = a_1 X_1 + B_1 \qquad (37a)$$

Using Procedure 1 from [7] we have

$$\underline{b}_1 = -a_1^2/4, \quad \overline{b} = -\underline{x}_1 \overline{x}_1 \qquad (37b)$$

In a similar way, we find the enclosure for (36b):

$$F_2(X) = a_2 X_1 + B_2 \qquad (38)$$

where

$$\beta = -1/a_2, \quad \underline{b}_2 = 1/\beta - a_2 \ln\beta, \quad \overline{b}_2 = f_2(\overline{x}_2) - a_2 \overline{x}_2$$

(In (37) and (38), a_1 and a_2 are the respective slopes.) The expressions (37a) and (38) are then represented in the form of two generalized intervals $\tilde{F}_1$ and $\tilde{F}_2$. Finally, the enclosure $F(X)$ of (35) is given by the product $\tilde{F}_1 \tilde{F}_2$, i.e.

$$F(X) = \tilde{F}_1 \tilde{F}_2 = \gamma_1 V_1 + \gamma_2 V_2 + B \qquad (39)$$

Using (39), we have obtained

$$F(X) = [0.0367, \; 0.3045] \qquad (40)$$

The same example was solved in [9] using first- and second-order interval derivatives and slopes. The following results have been obtained there:

$$F_{D1} = [-0.013, \; 0.35] \quad F_{S1} = [0.018, \; 0.32] \qquad (41a)$$

$$F_{D2} = [\;0.014, \; 0.32] \quad F_{S2} = [0.021 \; 0.31] \qquad (41b)$$

Comparison of (40) and (41) shows that for the example considered the present algorithm provides a narrower enclosure than the approach used in [9]

3. The new method

It is based on the following set of procedures to be carried out at each iteration. The interval extensions required in every procedure are implemented using the new linear form (3).

Procedure 1. Let $\overline{\varphi}_0$ be a current upper bound of $\varphi_0(x)$, $x \in X$, where $X \subseteq X^{(0)}$. If

$$\Phi_0(X) > \overline{\varphi}_0 \qquad (42)$$

then X is discarded [1], [3].

Procedure 2. Monotonocity test [1], [3]. If X is strictly admissible (s.a.) [3] and for some i

$$G_i(X) > 0 \text{ or } G_i(X) < 0 \qquad (43)$$

where $G_i(X)$ is the interval extension of $\partial\varphi_0 / \partial x_i$ then X is discarded.

Procedure 3. Nonconvexity test. If X is s.a. and

$$H_{ii}(X) < 0 \qquad (44)$$

for any $i = 1,\dots,n$, then X is discarded [3]. Here $H_{ii}(X)$ is the interval extension of $\partial^2\varphi_0 / \partial x_i^2$.

Procedure 4. Inadmissibility test. If

$$\Phi_i(X) > 0 \qquad (45)$$

for any $i = 1,\dots r$, X is discarded [3].

Procedure 5. If

$$\Phi_i(X) < 0 \qquad (46)$$

then the corresponding constraint is inactive and can be ignored for the current iteration.

Procedure 6. This procedure is an attempt to reduce the current box $X \in R^n$ using separately each of the inequalities

$$L_0(Y) = \sum_{j=1}^{n} \alpha_j^{(0)} y_i + B_0 \le \overline{\varphi}_0 \le 0 \qquad (47a)$$

$$L_i(Y) = \sum_{j=1} \alpha_j^{(i)} y_i + B_i \le \overline{\varphi}_0 \le 0, \quad i = 1,\dots n \qquad (47b)$$

where $L_i(Y)$ is the linear interval form (3) corresponding to function $\varphi_i(x)$, $i = 0,\dots r$.

Procedure 7. This procedure is based on the linearization of the so-called Fritz-John system

$$f_j(x) = u_0 g_0^{(j)}(x) + \sum_{i=1}^{r} u_i g_i^{(j)}(x) = 0, \quad j = 1,\dots,n$$

$$f_j(x) = u_j \varphi_j(x) = 0, \quad j = 1,\dots,r \qquad (48)$$

$$f(m) = \sum_{i=0}^{r} u_i - 1 = 0$$

where e u_i are scalars. This is a nonlinear system of m equations in m unknowns and, in general, $m = n + r + 1$. However, at some iterations m can be reduced by Procedure 2 or/and Procedure 5. Each function $f_j(x)$ in (48) is then linearized in $X \subset R^m$ and the following system is set up

$$Ax = B \qquad (49)$$

where A is constant (real) $m \times n$ matrix while B is an interval vector. System (11) is then solved in an efficient manner by the use of the so-called equationwise constraint propagation [8]

4. Numerical example

We consider the following optimization problem ([3], p [illegible]

Minimize

$$\varphi_0(x) = x_1^6 - 6.3x_1^4 + 12x_1^2 + 6x_1x_2 + 6x_2^2 \qquad (50a)$$

subject to

$$\varphi_1(x) = 1 - 16x_1^2 - 25x_2^2 \le 0$$

$$\varphi_2(x) = 13x_1^3 - 145x_1 + 85x_2 - 400 \le 0 \quad (50b)$$

$$\varphi_3(x) = x_1x_2 - 4 \le 0$$

There are two global solutions:

$$x_1^* = \pm 0.06604. \quad x_2^* = \pm 0.1929 \quad (51)$$

and the global minimum is

$$\varphi_0^* = 0.1990 \quad (52)$$

The problem has been solved with accuracy $\varepsilon = 10^{-4}$ for various starting boxes $X^{(0)}$. The results are quite encouraging. Thus, for $X_1^{(0)} = X_2^{(0)} = [-1,4]$ the method required N_1=107 iterations to find the global solutions (51). (52) and took t=0.53 sec when run on a 166 MHz Pentium computer. As the volume of the starting box $X^{(0)}$ was enlarged 1000000 times, the run time was increased only 3.562 times.

5. Conclusion

In this paper, the general nonlinear programming problem (or its variants) is addressed. A new interval method for the global solution of the optimization problem considered has been suggested. It is based on an alternative interval linearization of the nonlinear functions involved which is updated at each iteration of the computation process. The interval linearization suggested is more general than other known linearization forms since it is capable of enclosing functions that are only continuous or even discontinuous. The present interval linearization is in the form of an affine interval function where only the additive term is an interval which accounts for its better enclosing properties.

In its present form, the global optimization method suggested is based on the use of seven procedures that are implemented through the new interval linearization. More sophisticated computational schemes, including additional procedures, are however possible. If the optimization problem considered involves a system of nonlinear inequality constraints, one such procedure can solve (in the sense of [3]) this inequality system at each iteration of the iterative process.

4. Acknowledgments

This work was supported by the Bulgarian Research Foundation under Contract No TN-554/97.

5. References

[1] Moore, R. Methods and applications of interval analysis. SIAM, Philadelphia, 1979.

[2] Ratscek, H., Rokne, J. New computer methods for global optimization.. Chichester: Horwood, 1988.

[3] Hansen, E. Global optimization using interval analysis. New York, etc.: Marcel Dekker, Inc., 1992.

[4] Kolev, L. Interval methods for circuit analysis. Singapore, New Jersey: World Scient. 1993.

[5] Kearfort, R. Rigorous global search: continuous problems. Kluwer academic problems, Dordrecht, 1996.

[6] Kolev, L. Un efficient interval method for global analysis of nonlinear resistive circuits. Int. J. of Circuit Theory and Appl., vol. 26, 1998, pp. 81-92.

[7] Kolev, L. A new method for global solution of systems of nonlinear equations. Reliable Computing, vol. 4, No 2, 1998, pp. 125-146.

[8] Kolev, L. An improved method for global solution of nonlinear systems. Reliable Computing. vol. 5, No 2, 1999, pp. 103-111.

[9] Zuhe, S., Wolfe, M.A. On interval enclosures using slope arithmetic. Appl. Math. Comput. 39, 1990, pp. 89-105

121

IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS

I: FUNDAMENTAL THEORY AND APPLICATIONS

A PUBLICATION OF THE IEEE CIRCUITS AND SYSTEMS SOCIETY

WWW.IEEE-CAS.ORG

MAY 2000 VOLUME 47 NUMBER 5 ITCAEX (ISSN 1057–7122)

PAPERS

Analog and Mixed Signal Integrated Circuits

Analog Circuits and Filters

Chaos and Bifurcation

Computational Methods

General Circuits and Systems Theory

Multidimensional Signals and Systems

Nonlinear Circuits and Systems

Power Electronics and Systems

TRANSACTIONS BRIEFS

CALLS FOR PAPERS

IEEE TRANSACTIONS ON

CIRCUITS AND SYSTEMS

I: FUNDAMENTAL THEORY AND APPLICATIONS

A PUBLICATION OF THE IEEE CIRCUITS AND SYSTEMS SOCIETY

WWW.IEEE-CAS.ORG

MAY 2000 VOLUME 47 NUMBER 5 ITCAEX (ISSN 1057-7122)

PAPERS

Analog and Mixed Signal Integrated Circuits

Analog Circuits and Filters

Chaos and Bifurcation

Computational Methods

General Circuits and Systems Theory

Multidimensional Signals and Systems

Nonlinear Circuits and Systems

Power Electronics and Systems

TRANSACTIONS BRIEFS

CALLS FOR PAPERS

An Interval Method for Global Nonlinear Analysis

Lubomir Kolev

***Abstract*—In this paper, the problem of finding the set of all real solutions to a system of n nonlinear equations contained in a given n-dimensional box [the global nonlinear analysis (GNA) problem] is considered. A new iterative interval method for solving the GNA problem is suggested. It is based on the following techniques: 1) transformation of the original system into an augmented system of $n' = n + m$ equations of n' variables by introducing m auxiliary variables, the augmented system being of the so-called semiseparable form; 2) enclosure of the nonlinear augmented system at each iteration by a specific linear interval system of size $n' \times n'$; 3) elimination of the auxiliary variables; and 4) solution of the resulting reduced size $n \times n$ linear system, using the so-called constraint propagation approach. The method suggested shows a significant improvement over previous techniques for the numerical examples solved.**

***Index Terms*—Global nonlinear analysis, global solution of nonlinear systems, interval analysis, interval methods.**

I. Introduction

THE problem of finding the set of all real isolated solutions to a system of n nonlinear equations of n variables is one of the fundamental problems in science and technology. In the field of circuit and system analysis and design, typical applications are global analysis of resistive nonlinear circuits, synthesis of linear passive and active circuits, fault diagnosis of linear circuits, and determination of equilibria in neural networks. Other applications are load flow determination in power systems and computer graphics. Solving systems of equations is also an integral part of many algorithms for global optimization.

Over the last few decades, numerous methods have been proposed for tackling the global nonlinear analysis (GNA) problem. They can be categorized into the following two major groups:

i) continuation (homotopy) methods [1]–[4];

ii) interval methods [5]–[13].

The methods of the first group solve the global analysis problem solely in the special case where the nonlinear functions in the system considered are multipolynomials [4]. In the general case of arbitrary functions they only are globally convergent to one or more solutions without guaranteeing localization of all solutions.

Presently, interval methods (methods based on interval analysis techniques [5]–[7]) seem to be the only methods which are capable of infallibly solving the GNA problem for arbitrary functions. The problem can be formulated as follows (cf. [13] and the references therein cited).

The GNA problem. Let $X^{(0)} = (X_1^{(0)}, \ldots, X_n^{(0)})$ be an interval vector (a box) whose components are the intervals $X_i^{(0)} = [\underline{x}_i^{(0)}, \overline{x}_i^{(0)}]$. Given the function $f : D \subset R^n \to R^n$ and $X^{(0)} \subset D$, find the set $S(f, X^{(0)}) = \{x^{(1)}, \ldots, x^{(p)}\}$ of all real isolated solutions (zeros) to the system of equations

$$f(x) = 0 \tag{1a}$$

which are contained in $X^{(0)}$, i.e., when

$$x \in X^{(0)}. \tag{1b}$$

The initial box $X^{(0)}$ is chosen large enough to enclose all solutions to (1a) in R^n. There are recommendations as to how to choose $X^{(0)}$ in the case of nonlinear resistive circuits [14], [15].

At their present stage of development, all interval methods known to date suffer from a serious drawback which severely limits their applicability, namely, their numerical complexity grows too rapidly with the dimension n of the system and the size of the initial search region $X^{(0)}$. Thus, for a system of nine equations (representing an accurate Ebers–Moll model of a transistor) and a relatively small initial region $X^{(0)}$ with $X_i^{(0)} = [0, 10]$, a method suggested in [16] requires billions of functions evaluations in interval form to locate the unique zero of the GNA problem considered and to prove uniqueness computationally.

Several attempts to improve the numerical efficiency of the interval methods have already been made. One is based on the use of the so-called modified interval extensions [17] of the functions involved which have reduced overestimation. An alternative approach is associated with using interval slopes [18]–[21] rather than interval derivatives in evaluating the interval extensions. However, experimental evidence showed that the overall improvement of the interval methods efficiency obtained along these lines is still not satisfying, especially for more complex problems of larger n and initial search region [12], [22].

Recently, a more efficient interval method for global analysis has been developed [13] for the class of nonlinear circuits and systems described by systems of separable form

$$f_i(x) = \sum_{j=1}^{n} f_{ij}(x_j), \qquad i = 1, \ldots, n. \tag{2}$$

In this paper, a new interval method for solving the GNA problem considered is suggested. It is based on the following

Manuscript received April 23, 1998; revised August 10, 1999. This work was supported in part by the Bulgarian Research Foundation under Contract TN-554/97. This paper was recommended by Associate Editor J. Vlach.

The author is with the Faculty of Automatica, Technical University of Sofia, Sofia 1756 Bulgaria.

Publisher Item Identifier S 1057-7122(00)03975-1.

1057-7122/00$10.00 © 2000 IEEE

approach. First, by adding a certain number m of auxiliary variables, the original general form system (1) is transformed into a larger system of the special form

$$f_i(\bar{x}) = \sum_{j=1}^{n'} f_{ij}(x_j) + \sum_{\substack{k=1\\k\neq l}}^{n'} \sum_{l=1}^{n'} \alpha_{kl}^{(i)} x_k x_l,$$
$$i = 1, \ldots, n', \qquad n' = n + m. \quad \text{(3a)}$$

defined in a n'-dimensional box, i.e., when

$$\bar{x} \in X^{(0)} \quad \text{(3b)}$$

(some of the terms $f_{ij}(x_j)$ or/and some of the products $x_k x_l$ may be missing). Expressions (3a) differ from (2) in that now they include additionally the products $x_k x_l$. If all the terms involving products are missing, we get a function in separable form. For this reason, the (3a) will be referred to as a system in semiseparable form. Next, the method [13] treating the separable form case (2) is generalized to cover the augmented size semiseparable form system (3). Finally, two important modifications in the computational scheme of the generalized method are introduced: 1) elimination of the auxiliary variables and 2) improved solution of the resulting reduced size $n \times n$ linear system.

The paper is organized as follows. Sect on II presents the transformation of the general form system (1) to the system of semiseparable form (3). In Section III, the previous method [13] is extended for the case of semiseparable fc.rm system (3). The improvements in the computational scheme of the extended method are presented in Section IV. The overall efficiency of the resulting method is illustrated by several examples. Concluding remarks are given in Section V.

II. Transformation to Semiseparable Form

In this section, it will be shown how a system of general form (1) can be transformed into a system of semiseparable form (3). More specifically, the system's components f_i are assumed to be composed of four arithmetic operations $(+, -, *, /)$, unary operations (sin, exp, log, sqrt, abs, etc.), and the power operation (^).

The transformation of (1) into (3) includes two stages: 1) transforming (1a) into (3a) and 2) transforming (1b) into (3b).

A. Transforming the System of Equations

The approach herein adopted is to transform the general form system (1a) into the semiseparable form (3a). The idea of decomposing a nonlinear function into a set of simpler functions has already been suggested in the literature (e.g., [23] and [24]). The present approach is a simplification of the algorithm from [24] which takes into consideration the specificity of the semiseparable form.

Let f_L and f_R be subexpressions of f_i depending on at least one variable. Consider the following four cases:

$$f = f_L + f_R \quad (4)$$
$$f = f_L \cdot f_R \quad (5)$$
$$f = f_L / f_R \quad (6)$$
$$f = (f_L)^{f_R}. \quad (7)$$

If both f_L and f_R depend on only one and the same variable then f is obviously semiseparable in all the four cases.

If f_L depends on only one single variable and f_R depends on another single variable, then only (4) is semiseparable. The remaining three cases can, however, be easily transformed into semiseparable form. Consider first the product (5). By introducing two auxiliary variables, (5) is transformed into semiseparable form as follows:

$$f = f_L \cdot f_R \rightarrow \begin{cases} f = y_1 \cdot y_2 \\ y_1 = f_L \\ y_2 = f_R. \end{cases} \quad (8)$$

The third case can be easily reduced to the second by replacing $y_2 = f_R$ with $y_2 = 1/f_R$. It should be stressed that this is only possible if the original function $f_R \neq 0$ for all values of its argument. For the last case, the transformation to semiseparable form is as follows:

$$f = (f_L)^{f_R} \rightarrow \begin{cases} f = \exp(y) \\ y = f_R \cdot \log(f_L) \end{cases} \quad (9)$$

where the product in (9) must be transformed using (8). It should be mentioned that (9) is only valid if $f_L > 0$ for all values of its argument.

If both f_L and f_R depend on more than one variable we first introduce auxiliary variables and then apply the above approach. For instance, case (5) is transformed using (8). Now, by representing f_L and f_R in semiseparable form, f can be put into semiseparable form.

In order to make f_L and f_R semiseparable, we perform the above transformation to f_L and f_R, regarding them as f. This process can be implemented as a computer program (see [24] for the more complex case of transformation to separable form).

To illustrate the above approach we shall consider an example.

Example 1: Consider the system

$$\begin{aligned} 1 - x_3 - x_4 + 2x_1(x_3 + x_4) &= 0 \\ 2x_2 x_3 - x_1 &= 0 \\ x_3\left(x_1^2 + x_2^2 - 1\right) &= 0 \\ x_4\left(x_1^2 - x_2\right) &= 0 \end{aligned} \quad \text{(10a)}$$

where $x = (x_1, \ldots, x_4)$ belongs to the initial box $X^{(0)}$ with components

$$X_1^{(0)} = [-1, 1], \quad X_2^{(0)} = [0, 1]$$
$$X_3^{(0)} = [0, 1], \quad X_4^{(0)} = [0, 1]. \quad \text{(10b)}$$

The problem is to transform (10) to an equivalent system of semiseparable form (3).

It is seen that only the first two equations of system (10a) are semiseparable. To transform the last two equations into semiseparable form, we introduce two auxiliary variables

$$x_5 = x_1^2 + x_2^2 - 1 \quad \text{(11a)}$$
$$x_6 = x_1^2 - x_2. \quad \text{(11b)}$$

On account of (10a) and (11), we get

$$
\begin{aligned}
1 - x_3 - x_4 + 2x_1(x_3 + x_4) &= 0\\
2x_2x_3 - x_4 &= 0\\
x_3x_5 &= 0\\
x_4x_6 &= 0\\
x_1^2 + x_2^2 - 1 - x_5 &= 0\\
x_1^2 - x_2 - x_6 &= 0.
\end{aligned}
\tag{12a}
$$

System (12a) is now in semiseparable form. However, while the original system (10a) has four equations of four variables, the equivalent system (12a) has six equations of six variables. For this reason, (12a) will be referred to as the augmented system.

So far, we have transformed (10a) to the semiseparable form system (12a). To complete the solution of Example 1, we need also to enlarge the initial four-dimensional (4-D) box $X^{(0)}$ to a corresponding six-dimensional (6-D) box $\tilde{X}^{(0)}$.

B. Transforming the Initial Box

We first consider the transformation of $X^{(0)}$ in Example 1. With this in mind, we have to compute bounds X_5 and X_6 on the auxiliary variables. Using (10b) and (11), it is easily seen that

$$
\begin{aligned}
X_5^{(0)} &= X_1^2 + X_2^2 - 1 = [-1, 1]\\
X_6^{(0)} &= X_1^2 - X_2 = [-1, 1].
\end{aligned}
\tag{12b}
$$

Based on this example, we now proceed to considering the general case. Let the augmented system of separable form be denoted as

$$f(\tilde{x}) = 0 \tag{13a}$$

where $\tilde{x} = (x_1, \ldots, x_{n'})$ is the augmented vector of variables and $n' = n + m$, with m being the number of auxiliary variables needed to transform (1a) into (3a). For the above example $n = 4$, $m = 2$, $n' = 6$, and $\tilde{x} = (x_1, \ldots, x_6)$. It is necessary to determine the initial box $\tilde{X}^{(0)}$ for $\tilde{x}$ in order to complete (13a) with the condition

$$\tilde{x} \in \tilde{X}^{(0)}. \tag{13b}$$

To do this, we partition the augmented vector $\tilde{x}$ into two parts

$$x = (x_1, \ldots, x_n) \tag{14a}$$

$$x_a = (x_{n+1}, \ldots, x_{n'}) \tag{14b}$$

corresponding to the original and the auxiliary variables, respectively. Thus, the augmented vector can be put in the form

$$\tilde{x} = (x, x_a). \tag{15}$$

In the general case, the auxiliary variables and the original variables are related by the function $f_a\colon R^n \to R^m$, i.e.,

$$x_a = f_a(x) \tag{16}$$

In Example 1, the function f_a is given by (11). We can compute bounds X_a on x_a for any box $X \subseteq X^{(0)}$ by evaluating the interval extension $F_a(X)$ of $f_a(x)$ when $x \in X$, i.e.

$$X_a = F_a(X), \quad X \subseteq X^{(0)} \tag{17}$$

Thus, we can compute the interval vector

$$X_a^{(0)} = F_a\left(X^{(0)}\right) \tag{18}$$

which determines the bounds on the auxiliary variables. Finally, the initial box is given in partitioned form by the interval vector $\tilde{X}^{(0)}$

$$\tilde{X}^{(0)} = \left(X^{(0)}, X_a^{(0)}\right). \tag{19}$$

Thus, it has been shown that the original problem (1) involving general form equations can be transformed into a corresponding augmented system (13). The components f_i in (13a) are in the semiseparable form (3a) and the initial box $\tilde{X}^{(0)}$ related to (13a) can be computed using (18) and (19).

Once the transformation of (1)–(13) is completed, the solution of the original GNA problem can be equated to that of finding all real solutions to (13). Indeed, let $\tilde{x}^* = (x^*, x_a^*)$ be a solution to (13). Clearly, the vector x^* in $\tilde{x}^*$ provides a solution to the original problem (1). Thus, our next objective is to devise an efficient meth:d for solving systems of semiseparable form.

III. Solving Semiseparable Form Systems

In this section, an interval method for solving the semiseparable form system (3) will be presented. It is an extension of a previous method [13] developed for the special case of separable functions. More specifically, the idea to enclose the separable terms in (3a) by an appropriate linear interval function [13] is generalized to encompass the terms representing products of two variables.

To simplify notation, the symbols x, $X \in X^{(0)}$, $X^{(0)}$ will stand for the n'-dimensional vectors $\tilde{x}$, $\tilde{X}$, $\tilde{X}^{(0)}$, respectively, throughout this section.

A. Enclosures for Semiseparable Functions

To maintain completeness, first the linear interval approximation for the separable terms will be briefly presented.

1) Enclosure for $f_{ij}(x_j)$: Let $X = (X_1, \ldots, X_j, \ldots, X_{n'}) \subseteq X^{(0)}$. The linear interval enclosure (approximation) of $f_{ij}(x_j)$ in X_j suggested earlier [13] is in the form

$$L_{ij}(X_j) = B_{ij} + a_{ij}x_j, \qquad x_j \in X_j \tag{20}$$

where B_{ij} is an interval, while a_{ij} is a real number. Both B_{ij} and a_{ij} (which, in fact, depend on X) are determined such that the following inclusion property holds:

$$f_{ij}(x_j) \in B_{ij} + a_{ij}x_j, \qquad x_j \in X_j. \tag{21}$$

[It is on account of (21) that $L_{ij}(X)$ is called a linear interval enclosure (approximation) of $f_{ij}(x_j)$.] A procedure for determining a_{ij} and B_{ij}, which will be called Procedure 1, has been suggested in [13].

Remark 1: In the original paper [13], no restrictions on the functions $f_{ij}(x_j)$ are imposed except for the requirement that they be continuous. The Procedure 1 therein suggested for determining the enclosure (20) is applicable for the case of continuously differentiable (CD) functions and piecewise linear (PWL) functions. It can, however, be easily shown (using simple geometrical considerations as in [13]) that the linear interval enclosure (20) can also be constructed, even in the case of discontinuous functions having bounded discontinuities.

2) Enclosure for $x_k x_l$: To simplify notation, we shall consider the product

$$xy, \quad x \in X, \qquad y \in Y$$

(where X and Y are intervals) rather than the product $x_k x_l$. If x_0 and y_0 are the centers of X and Y, respectively, then

$$\begin{aligned} xy &= (x_0 + u)(y_0 + v) = x_0 y_0 + y_0 u + x_0 v + uv \\ &= -x_0 y_0 + y_0 x + x_0 y + uv. \end{aligned} \tag{22}$$

When $x \in X$ and $y \in Y$, the centered variables $u \in [-R_x, R_x]$ and $v \in [-R_y, R_x]$ where R_x, R_y are the radii of X and Y. Let $R = R_x \cdot R_y$: it follows from (22) that

$$xy \in -x_0 y_0 + y_0 x + x_0 y + [-R, R], \quad x \in X, \quad y \in Y. \tag{23}$$

Thus, the product xy has been enclosed by a linear interval expression since

$$xy \in \alpha x + \beta y + B_{xy} \tag{24a}$$

where $\alpha = y_0, \beta = x_0$ are real numbers while

$$B_{xy} = -x_0 y_0 + [-R, R] \tag{24b}$$

is an interval.

Returning to the original term $\alpha_{kl}^{(i)} x_k x_l$ of the ith function $f_i(x)$ in (3a), it is easily seen that the corresponding linear interval enclosure denoted as $L_{kl}^{(i)}(X_k X_l)$ has the form

$$L_{kl}^{(i)}(X_k, X_l) = \alpha_{ik} x_{ik} + \alpha_{il} x_l + B_{kl}^{(i)} \tag{25a}$$

with

$$\alpha_{ik} = \alpha_{kl}^{(i)} x_l^{(0)}, \quad \alpha_{il} = \alpha_{kl}^{(i)} x_k^{(0)} \tag{25b}$$

$$B_{ik}^{(i)} = \alpha_{kl}^{(i)} B_{kl}^{(i)} \tag{25c}$$

where $x_k^{(0)}, x_l^{(0)}$ are the centers of X_k and X_l, respectively, while $B_{kl}^{(i)}$ is an interval and is computed as in (24b) [with x_0, y_0 and $R = R_x R_y$ replaced by $x_k^{(0)}, x_l^{(0)}$ and $R = R_k R_l$]. On account of (24a)

$$\alpha_{kl}^{(i)} x_k x_l \in \alpha_{ik} x_k + \alpha_{il} x_l + B_{kl}^{(i)} \tag{26a}$$

when

$$x_k \in X_k \quad x_l \in X_l. \tag{26b}$$

The above procedure for determining the slopes α_{ik}, α_{il} and the interval B_{lk} will be referred to as Procedure 2.

B. The Generalized Method

Using the semiseparable representation (3a) and the inclusions (21) and (26), it is seen that the following inclusion is valid

$$f_i(x) \in \sum_{j=1}^{n} a'_{ij} x_j + b_i^I, \quad i = 1, \ldots, n', \quad x \in X \tag{27a}$$

where

$$a'_{ij} = a_{ij} + \alpha_{ij} \tag{27b}$$

$$b_i^I = \sum_j B_{ij} + \sum_l \sum_k B_{lk}^{(i)} = [\underline{b}_i, \bar{b}_i] \tag{27c}$$

Now a real $n' \times n'$ matrix

$$A = \{a'_{ij}\} \tag{28}$$

is introduced and a n'-dimensional interval vector

$$b^I = (b_1^I, \ldots, b_n^I) \tag{29}$$

is formed. Thus, (27a) can be put in vector form

$$f(x) \in Ax + b^I, \quad x \in X. \tag{30}$$

If y is a solution of (3), then $f(y) = 0$ and by (30)

$$0 \in Ay + b^I, \quad y \in X. \tag{31}$$

Now we can state the main result of the section.

Theorem 1: All the solutions y to

$$f(x) = 0, \quad x \in X \tag{32}$$

with $f_i(x)$ given by (3a), which are contained in the n'-dimensional box X are also contained in the solution set $S(X)$ of the system

$$Ax + b = 0, \quad b \in b^I \tag{33}$$

where b is any real vector contained in b^I.

The proof of the present theorem is similar to that of Theorem 1 in [13] and will therefore be omitted.

Remark 2: It should be stressed that A and b^I depend on the box X, i.e., (33) should read

$$A(X) + b = 0, \quad b \in b^I(X). \tag{33'}$$

To simplify notation, here and henceforth the shorter form A and b^I is used.

Formally, Theorem 1 of this paper is an almost verbatim replica of Theorem 1 of the earlier paper [13] where the separable form equations case was considered. In reality, this paper's result is more general since it covers the case of equations in semiseparable form. The most important distinction is the inclusion (27) and, more specifically, the expressions (27b) and (27c) for the slopes a'_{ij} and the intervals b_i^I, respectively.

Based on Theorem 1, the method of [13] is readily extended to cover the semiseparable case. Technically, the only difference is that now the square real matrix A in (28) and interval vector b^I in (29) are $(n+m)$-dimensional while their counterparts in [13] are n-dimensional. For this reason, only several facts, needed in the sequel, will be briefly presented here. Let

$$B = -b^I. \tag{34}$$

The generalized method applicable to system (3) is an iterative method. It is based, essentially, on the following procedure.

Procedure 3: At each iteration k, a current n'-dimensional box $X^{(k)} \subseteq X^{(0)}$ is generated. Using Procedures 1 and 2, the corresponding $A^{(k)}$ and $B^{(k)}$ are formed [according to Remark 2 $A^{(k)}$ and $B^{(k)}$ stand for $A(X^{(k)})$ and $B(X^{(k)})$, respectively]. Using $A^{(k)}$ and $B^{(k)}$, an interval vector $Y^{(k)}$ is then computed (by formulae given in [13]). The iterative procedure is defined as follows:

$$X^{(k+1)} = Y^{(k)} \bigcap X^{(k)}, \quad k \geq 0. \tag{35}$$

The algorithm of the generalized method which is based on Procedure 3 will be referred to as Algorithm 1.

Similarly to other interval methods the present method can be used as a computationally verifiable test for existence of solutions to (3a) in X.

Theorem 2: Let $f : D \subseteq R^n \rightarrow R^n$ be a CD function in the domain D and $X^{(0)} \subset D$. Introduce the interval operator

$$K\left(X^{(k)}\right) = A^{-1}\left(X^{(k)}\right) B\left(X^{(k)}\right), \quad k \geq 0 \tag{36}$$

where $A(X^{(k)})$ and $B(X^{(k)})$ defined as in Procedures 1 and 2, correspond to the current box $X^{(k)}$. Then, if at some kth iteration

$$K\left(X^{(k)}\right) \subseteq X^{(k)} \tag{37}$$

the above inclusion implies the existence of a solution to (3a) in $X^{(k)}$.

The proof of the theorem is given in the Appendix.

The numerical performance of Algorithm 1 has been tested on several systems of equations of the form (3a). It will be illustrated by the following two examples.

Example 2: The problem is to find all real solutions to the system (10a) contained in the box $X^{(0)} \in R^4$ defined by (10b).

According to the approach adopted in Section II of this paper, we have to find all real solutions to the augmented system (12a) which are contained in the enlarged initial box $X^{(0)}$ determined by (10b) and (12b). The latter problem will be solved using Algorithm 1. To do this, we need to generate at each iteration k the corresponding subbox $X^{(k)}$. At this stage, we apply Procedures 1 and 2 for the current box $X^{(k)}$ to obtain the following linear interval system:

$$\begin{aligned} a_{11}x_1 + a_{13}x_3 + a_{14}x_4 &= B_1 \\ a_{22}x_2 + a_{23}x_3 - x_4 &= B_2 \\ a_{33}x_3 + a_{35}x_5 &= B_3 \\ a_{44}x_4 + a_{46}x_6 &= B_4 \\ a_{51}x_1 + a_{52}x_2 - x_5 &= B_5 \\ a_{61}x_1 - x_2 - x_6 &= B_6 \end{aligned} \tag{38}$$

where, for notational simplicity, the dependence of the real numbers a_{ij} and the interval vectors B_i on $X^{(k)}$ is not shown explicitly. Now $Y^{(k)}$ is obtained as the interval solution to (38). According to Procedure 3 $Y^{(k)}$ is to be determined at each kth iteration. Thus, Algorithm 1 reduces, essentially, to repeatedly setting up and solving the linear interval system (38) (for each subbox $X^{(k)}$) until the desired accuracy of solution ε [13] is met.

The augmented system (12), (10b) has been solved with $\varepsilon = 10^{-6}$ and the following two solutions were found

$$\begin{aligned} x^{(1)} = (&-1.000\,000, 0.000\,000, 0.333\,333, 0.000\,000, \\ &0.000\,000, 0.999\,999) \\ x^{(2)} = (&-0.786\,151, 0.618\,034, 0.173\,857, 0.214\,899, \\ &0.000\,000, 0.000\,000) \end{aligned} \tag{39}$$

where each component recorded to six decimal places is the midpoint of a corresponding side of the solution box enclosing a solution. (In actual computation the left end-point of all components $X_i^{(0)}$ of the starting box $X^{(0)}$ was lowered with $\partial = 10^{-7}$ to ensure the location of the first component of $x^{(1)}$.)

Algorithm 1 requires $N_i = 236$ iterations to locate the solutions (39). The solutions of the original nonlinear system (10) are given by the corresponding first four components of $x^{(1)}$ and $x^{(2)}$.

In the example considered, the original system (10a) is related to solving a global minimization problem [20] for which

$$x_5 \leq 0, \quad x_6 \leq 0. \tag{40}$$

On account of (40) the intervals (12b) are reduced to

$$X_5^{(0)} = [-1, 0], \quad X_6^{(0)} = [-1, 0]. \tag{41}$$

The augmented system (12a) has also been solved for the smaller box $X^{(0)}$, determined by (10b) and (41). A unique solution was thus found which is given by $x^{(2)}$. Now the number of iterations N_i needed to locate $x^{(2)}$ within an accuracy $\varepsilon = 10^{-5}$ is $N_i = 24$. The latter example has also been solved using Krawczyk's method [6], [7], [11], [20]. Although the more efficient componentwise version of the method was programmed now 993 iterations were needed to locate the solution within the same accuracy. The corresponding execution times for a Pentium 166-MHz computer are 0.018 and 0.684 s, respectively. The example illustrates two empirically observed facts: 1) the global analysis problem considered can be solved more efficiently by the present method and 2) both methods require approximately the same amount of calculation per iteration.

The following example is related to electric circuit synthesis ([11], p. 276).

Example 3: The problem is to realize the following voltage transfer function:

$$V(s) = \frac{0.186s^2 + 2.474}{0.327s^3 + 2.770s^2 + 4.945 + 4.949} \tag{42}$$

by means of a passive electric circuit ([11, Fig. 6.4]). It is desired to determine the component values G_1, C_1, C_2, C_3, GL and L_1.

It can be checked that these values are solutions of the following design equations system:

$$\begin{aligned} x_1 x_6 &= 2.474 \\ x_1 x_4 &= 0.186 \\ (x_1 + x_5)x_6 &= 4.949 \\ (x_2 + x_3)x_6 + x_1 x_5 &= 4.945 \\ x_1(x_2 + x_3) + x_5(x_2 + x_4) &= 2.770 \\ x_2(x_3 + x_4) + x_3 x_4 &= 0.327 \end{aligned} \tag{43}$$

where $x_1 = G_1$, $x_1 = C_1$, $x_4 = C_2$, $x_3 = C_3$, $x_5 = G_L$, and $x_6 = 1/L_1$. The initial box $X^{(0)}$ is given by

$$\begin{aligned} X_1^{(0)} &= [0.01, 2.0], \quad X_2^{(0)} = [0.01, 0.55] \\ X_3^{(0)} &= [0.01, 2.0], \quad X_4^{(0)} = [0.01, 0.50] \\ X_5^{(0)} &= [0.01, 2.5], \quad X_6^{(0)} = [0.01, 2.00]. \end{aligned} \tag{44}$$

The desired solution accuracy ε was chosen to be 10^{-3}. Using Algorithm 1, three solutions

$$\begin{aligned} x^{(1)} &= (1.515, 0.0837, 1.538, 0.1220, 1.516, 1.632) \\ x^{(2)} &= (1.783, 0.1780, 1.092, 0.1040, 1.784, 1.387) \\ x^{(3)} &= (1.907, 0.3490, 0.656, 0.0974, 1.908, 1.296) \end{aligned} \tag{45}$$

of (43), contained in (44), have been found within the desired accuracy. The number of iterations N_i required is 9169. It is worthwhile noting that Krawczyk's method requires 37 928 iterations to find all the three solutions (45) within the same accuracy.

IV. Improvements

As shown in Section III-B by Example 2, Algorithm 1 of the generalized method reduces, essentially, to setting up and solving the following $n' \times n'$ linear interval system

$$\tilde{A}^{(k)} x = \tilde{B}^{(k)} \tag{46}$$

at each iteration. In Example 2, $\tilde{A}^{(k)}$ and $\tilde{B}^{(k)}$ are given in (38).

In this section, two modifications will be introduced into the computational scheme of Algorithm 1. The first is associated with the elimination of all m auxiliary variables from the linear system (46). Thus, (46) is transformed to a system

$$A^{(k)} y = B^{(k)} \tag{47}$$

of reduced $n \times n$ size. The second modification consists of applying the constraint propagation approach (e.g., [28]) to the reduced system (47). These modifications result in a considerable improvement of the numerical efficiency of Algorithm 1.

A. Eliminating the Auxiliary Variables

The computational efforts needed to solve (46) can be substantially reduced (for large n and m) if the auxiliary variables are eliminated from system (46). This possibility will be shown by way of Example 2. Indeed, from (38)

$$\begin{aligned} x_5 &= a_{51} x_1 + a_{52} x_2 - B_5 \\ x_6 &= a_{61} x_1 - x_2 - B_6. \end{aligned} \tag{48}$$

Substituting (48) into the third and fourth equation of (38), we get

$$\begin{aligned} a_{11} x_1 + a_{13} x_3 + a_{14} x_4 &= B_1 \\ a_{22} x_2 + a_{23} x_3 - x_4 &= B_2 \\ a_{31} x_1 + a_{32} x_2 + a_{33} x_3 &= B_3' \\ a_{41} x_1 + a_{42} x_2 + a_{44} x_4 &= B_4' \end{aligned} \tag{49}$$

where

$$\begin{aligned} a_{31} &= a_{35}, \quad a_{32} = a_5 a_{52} \\ a_{41} &= a_{46} a_{61}, \quad a_{42} = -a_{46} \\ B_3' &= B_3 + a_{35} B_5, \quad B_4' = B_4 + a_{46} B_6. \end{aligned} \tag{50}$$

It is seen that the new system (49) has only four equations of four variables while the augmented linear system (38) has six equations of six variables.

In the general case, by elimination of the auxiliary variables the linear interval system (46) of augmented dimension $n + m$ is reduced to system (47) of dimension n. Solving (47) is a much easier problem than solving (46), especially for larger n and m.

It should be emphasized that the elimination of the auxiliary variables is carried out at each iteration, i.e., for each current box X. Therefore, we have to compute, at each iteration, bounds on the auxiliary variables. This can be done using (17), i.e., by evaluating some interval extension $F_{a,i}(X)$ (or still better, the range $f_{a,i}(X)$) of $f_{a,i}(x)$ when $x \in X$ where $f_{a,i}(x)$ is the ith component of the vector function $f_a(x)$. In general, the interval extension $X_{a,i} = F_{a,i}(X)$ of $f_{a,i}(x)$ in X can be computed using some of the methods available in interval analysis [5]–[8], [18]–[20]. However, Procedure 1 allows an alternative simpler approach to evaluating $X_{a,i}$. Indeed, $f_{a,i}(x)$ are semiseparable functions. Therefore

$$f_{a,i}(x) = \sum_{j=1}^{n} f_{a,ij}(x_j) + \sum_{\substack{k=1 \\ k \neq l}}^{n} \sum_{l=1}^{n} \alpha_{kl}^{(a,i)} x_k x_l. \tag{51}$$

However

$$f_{a,ij}(x) \in \alpha_{ij}^{a} x_j + B_{ij}^{a}. \tag{52}$$

Hence

$$f_{a,i}(x) \in l_i^{a}(x) + B_i^{a} \tag{53}$$

where

$$l_i^{a}(x) = \sum_{j=1}^{n} \alpha_{ij}^{a} x_j + \sum_{\substack{k=1 \\ k \neq l}}^{n} \sum_{l=1}^{n} \alpha_{kl}^{(a,i)} x_k x_l \tag{54a}$$

$$B_i^{a} = \sum_{j} B_{ij}^{a}. \tag{54b}$$

Thus, the interval extension $X_{a,i}$ can be computed in the form

$$X_{a,i} = L_i^{a}(X) + B_i^{a} \tag{55}$$

where $L_i^{a}(X)$ is the interval extension (or, if possible, the range) of $l_i^{a}(x)$ in X.

TABLE I
RESULTS FOR EXAMPLE 4

Algorithm	A1	A2
N_I	258	88
t (sec)	0.514	0.141
N_{cl}	6	3

The algorithm of the present method which implements the elimination of the auxiliary variables will be referred to as Algorithm 2.

Example 4: We consider a well-known network used to realize a third-order maximally flat (Butterworth) function (e.g., [27, Fig. 7]). If we now specify $R = 0.5\ \Omega$ and put $x_1 = C_1$, $x_2 = L_2$, $x_3 = C_3$, the design equations are

$$\begin{aligned} x_1 + 2x_2 + x_3 - 6 &= 0 \\ 2x_1x_2 + x_2x_3 - 6 &= 0 \\ x_1x_2x_3 - 3 &= 0. \end{aligned} \qquad (56)$$

It is known that system (56) has two solutions

$$x^{1} = (1.0, 1.5, 2.0), \quad x^{2} = (3.261, 0.779, 1.181). \qquad (57)$$

The last equation of (56) is not in semiseparable form. It can, however, be easily transformed into such form by introducing an auxiliary variable

$$x_4 = x_2x_3. \qquad (58)$$

The corresponding linear interval system (46) is

$$\begin{aligned} x_1 + 2x_2 + x_3 &= 6 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= B_2 \\ a_{31}x_1 + a_{34}x_4 &= B_3 \\ a_{42}x_2 + a_{43}x_3 + a_{44}x_4 &= B_4. \end{aligned} \qquad (59)$$

The initial $X^{(0)}$ was chosen to have equal components

$$X_i^{(0)} = [0, 4], \quad i = 1, \ldots, 4. \qquad (60)$$

The example considered was solved using Algorithm 1 (by solving the augmented system (59)) and Algorithm 2 (by eliminating x_4 in (59) and solving a system of three equations). Both algorithms have located infallibly the two solutions (57) with $\varepsilon = 10^{-4}$. Data illustrating the improved efficiency of Algorithm 2 as compared to Algorithm 1 are given in Table I.

In Table I, N_I is the number of iterations required to solve the problem considered within the given accuracy, t (in seconds) is the corresponding execution time for a Pentium 166-MHz computer and N_{cl} is the number of cluster boxes (boxes generated additionally to the two solution boxes by the respective algorithm) [13], [26].

For comparison, the same example has been solved by Krawczyk's method also (its improved componentwise version [2]–[4]). It is worthwhile noting that the latter method required $N_I = 1156$ iterations to solve the GNA problem considered.

B. Use of Constraint Propagation

The second modification is related to the application of the constraint propagation approach [28] to solving the reduced system (47). To simplify notation, it is written in the form

$$Ay = B \qquad (61)$$

which actually stands for

$$Ay = b, \quad b \in B. \qquad (62a)$$

Taking into account the fact that y must remain in the current box X, (62a) is to be completed with the condition

$$y \in X. \qquad (62b)$$

The problem is to find an interval solution to (62), that is, an interval vector Y which contains the solution set of (62)

$$S(A, B, X) = \{y\colon Ay = b, b \in B, y \in X\}. \qquad (63)$$

The optimal interval solution Y^* will be the smallest interval solution still containing $S(A, B, X)$.

It is readily seen that each component $Y_i^* = [\underline{y}_i^*, \overline{y}_i^*]$, $i = 1, 2, \ldots, n$ can be determined by solving two linear programming problems

$$\begin{aligned} &y_i = \min \\ &Cy - b = 0, \quad y \in X, \quad b \in B \end{aligned} \qquad (64)$$

and

$$\begin{aligned} &y_i = \max \\ &Cy - b = 0, \quad y \in X, \quad b \in B \end{aligned} \qquad (65)$$

to find the endpoints $\underline{y}_i^*$ and $\overline{y}_i^*$, respectively. Thus, computing Y^* would require the solution of $2n$ linear programming problems. Such an approach to tackling the GNA problem considered seems to be rather costly since Y^* is to be computed at each iteration k. Therefore, a simpler approach will be adopted here which is based on computing a tight interval solution Y in a cheaper manner. This is made possible by resorting to the constraint propagation approach as a preliminary stage in solving (62). Several algorithms implementing the latter approach will be presented now.

Algorithm A3: This is an algorithm that is based on the following procedure involving two stages.

Stage A. For $i = 1$ to n do

$$Y_i = \frac{1}{a_{ii}}\left[B_i - \sum_{\substack{j\neq i\\ j=1}}^{n} a_{ij}X_j\right] \qquad (66a)$$

$$X_i := Y_i \bigcap X_i. \qquad (66b)$$

It is seen that this stage implements the known interval Gauss–Seidel scheme [2]–[4]. Note that (66) is, however, a simpler version since unlike other interval methods now all the coefficients a_{ij} are real numbers rather than intervals.

Stage B. Now one iteration of Procedure 3 is applied to the box X obtained on exit from Stage A.

TABLE II
RESULTS FOR EXAMPLE 5

Algorithm	A2	A3	A4	A5
N_i	3233	2780	1354	473
t (sec)	6.86	5.68	2.97	1.18

Algorithm A4: This is an extended version of the previous algorithm, in which the first stage is modified as follows.

```
Stage A. For $i = 1$ to $n$ do
  For $j = 1$ to $n$ do
```

$$Y_j = \frac{1}{a_{ij}} \left[B_i - \sum_{\substack{k \neq j \\ k=1}}^{n} a_{ik} X_k \right] \tag{67a}$$

$$X_j := Y_j \, \mathrm{I} \, X_j. \tag{67b}$$

(In actual computation, (67a) is implemented in a more efficient manner by first computing

$$S_1 = B_i - \sum_{\substack{k \neq 1 \\ k=1}}^{n} a_{ik} X_k = [\underline{S}_1, \overline{S}_1]. \tag{68}$$

Then

$$\underline{S}_2 = \underline{S}_1 + \underline{a_{i1} X_1} - \underline{a_{i2} X_2} \tag{69a}$$

and

$$\overline{S}_2 = \overline{S}_1 + \overline{a_{i1} X_1} - \overline{a_{i2} X_2}. \tag{69b}$$

(The next sums S_j, $j > 2$ are computed in a similar way.)

Stage B. The same as in algorithm A3.

In the algorithms presented so far, all elements of the real matrix A and the interval vector B are computed at the start of the current iteration and remain unchanged during the iteration. A better, equationwise (row by row) computation of A and B is implemented in the next algorithm.

Algorithm 5: In this algorithm, stage A is modified in the following manner. Initially for $i = 1$, we compute the first row of A and the first element of B using the current box X. We then apply (67) to (hopefully) reduce X to a new box X'. Now X' is renamed X and the second row of A and the second element of B are determined. Now (67) is applied with $i = 2$. This process continues until $i = n$.

Stage B. The same as in algorithm A4.

To illustrate the efficiency of the above algorithms, a numerical example will be considered.

Example 5: The system to be solved is

$$\begin{aligned} x_3\left(6x_1^5 - 25.2x_1^3 + 24x_1 + 6x_2\right) \\ -32x_1x_4 + x_5\left(39x_1^2 - 145\right) + x_2x_6 &= 0 \\ x_3(6x_1 + 12x_2) - 50x_2x_4 + 85x_5 + x_1x_6 &= 0 \\ x_4\left(16x_1^2 + 25x_2^2 - 1\right) &= 0 \\ x_5\left(13x_1^3 - 145x_1 + 85x_2 - 400\right) &= 0 \\ x_6(x_1x_2 - 4) &= 0 \\ x_3 + x_4 + x_5 + x_6 - 1 &= 0. \end{aligned} \tag{70a}$$

The initial box $X^{(0)}$ has the following components:

$$X_1^{(0)} = X_2^{(0)} = [-2, 4], \quad X_i^{(0)} = [0, 1], \quad i = 3, \dots 6. \tag{70b}$$

The GNA problem considered has 9 solutions

$$\begin{aligned} x^{(1)} &= (-1.7475, 0.8738\ 1, 0, 0, 0) \\ x^{(2)} &= (-1.075, 0.5353\ 1, 0, 0, 0) \\ x^{(3)} &= (-0.2398, -0.056\,48\ 0, 5716, 0, 4284\ 0, 0) \\ x^{(4)} &= (0.066\,04, -0.1929\ 0, 8341\ 0, 1659\ 0, 0) \\ x^{(5)} &= (0.2398, 0.056\,48\ 0, 5716, 0, 4284\ 0, 0) \\ x^{(6)} &= (0, 0, 1, 0, 0, 0) \\ x^{(7)} &= (-0.066\,04, 0.1929\ 0, 8341, 0, 1659\ 0, 0) \\ x^{(8)} &= (1.075, -0.5353\ 1, 0, 0, 0) \\ x^{(9)} &= (1.7475, -0.8738\ 1, 0, 0, 0). \end{aligned}$$

They have been located by algorithms A2–A5 with $\varepsilon = 10^{-4}$. No cluster effect has been observed. Data about number of iterations and execution time (for a Pentium 166 MHz) required by each algorithm are given in Table II.

It is seen that algorithm A5 has relatively the best performance. For the same accuracy Krawczyk's method required $N_i = 33\,089$ iterations to solve the GNA problem considered.

V. CONCLUSION

In this paper, the problem of finding (within preset accuracy) the set of all real solutions to a system of nonlinear equations (1) contained in a given box $X^{(0)}$ (the GNA problem) has been considered. The problem formulation is rather general; the nonlinear functions involved need only be bounded and may even be discontinuous.

A new method for solving the GNA problem has been suggested. It comprises the following stages. First, the original system (1) of size n is transformed into a new system (3) of size $n + m$ by introducing m auxiliary variables. The equations of the latter system are in the semiseparable form (3a) which is a sum of the functions $f_{ij}(x_j)$ of a single variable and of products $\alpha_{kl}^{(i)} x_k x_l$. Each term $f_{ij}(x_j)$ or $\alpha_{kl}^{(i)} x_k x_l$ is then enclosed tightly by the linear interval functions (20) or (25), respectively, at each iteration of the computation process. Using these enclosures, an augmented linear interval system (46) of size $n + m$ is obtained. Next, the auxiliary variables are eliminated from (46) and a reduced linear interval system (47) of size n, having a real matrix A and an interval vector right-hand side, is set up and solved at each iteration of the method. This advantageously distinguishes the present method from the other known interval methods where a much more complex linear interval system having an $n \times n$ interval matrix and a real right-hand side vector is to be solved at each iteration. Finally,

(47) is solved in a most efficient manner using the constraint propagation approach as implemented in Algorithm 5.

Experimental data show that as regards computer time and memory volume requirements the present method outperforms the other known interval meth82a ods for solving the GNA problem considered.

There seems to exist a possibility for further improvement of the numerical efficiency of the new method by incorporating into its scheme ideas and techniques from affine arithmetic [29] in order to automate the transformation of the original nonlinear system (1) into the linear interval system (47).

APPENDIX

Proof of Theorem 2: Let y be a solution to (32). Then (32) can be transformed into the fixed point format

$$x = C(x)b(x) = P(x)$$

where $C : X^{(k)} \to R^n$ and $b : X^{(k)} \to R^n$ are to be determined. We choose

$$C(x) = C^{(k)} = A^{-1}\left(X^{(k)}\right),$$

$$b(x) = b \in B\left(X^{(k)}\right).$$

Obviously, for any $x \in X^{(k)}$

$$P(x) \in K\left(X^{(k)}\right).$$

Thus, if (37) holds, then P maps $X^{(k)}$ into itself. Therefore, by the Shrauder fixed point theorem P has a fixed point in $X^{(k)}$ and, hence, (32) has a solution in $X^{(k)}$.

ACKNOWLEDGMENT

The author is grateful to the reviewers for their valuable recommendations and comments.

REFERENCES

[1] L. O. Chua and A. Ushida, "A switching-parameter algorithm for finding multiple solutions of nonlinear resistive networks," *Int. J. Circuit Theory Appl.*, vol. 4, pp. 215–237, 1976.

[2] C. B. Garcia and W. I. Zangwill, *Pathways to Solutions, Fixed Points and Equilibria.* Englewood Cliffs, NJ: Prentice-Hall, 1981.

[3] S. L. Richter and R. A. DeCarlo, "Continuation methods: Theory and applications," *IEEE Trans. Circuits Syst.*, vol. CAS-30, pp. 347–352, 1983.

[4] A. Morgan, *Solving Polynomial Systems Using Continuation for Engineering and Scientific Pohlems.* Englewood Cliffs, NJ: Prentice-Hall, 1987.

[5] R. E. Moore, *Interval Analysis.* Englewood Cliffs, NJ: Prentice-Hall, 1966.

[6] ——, *Methods and Applications of Interval Analysis* Philadelphia, PA, 1979. SIAM.

[7] G. Alefeld and J. Herzberger, *Introduction to Interval Computations.* New York, NY: Academic , 1983.

[8] E. Hansen and R. Greenberg, "An interval Newton method," *Applied Math. Comp.*, vol. 12, pp. 89–98, 1983.

[9] L. Kolev, "Finding all solutions of nonlinear resistive circuit equations via interval analysis," *Int. J. Circuit Theory Appl.*, vol. 12, pp. 75–178, 1984.

[10] L. Kolev and V. Mladenov, "An interval method for finding all operating points of nonlinear resistive circuits," *Int. J. Circuit Theory Appl.*, vol. 18, pp. 257–267, 1990.

[11] L. Kolev, *Interval Methods for Circuit Analysis*, Singapore: World Scientific, 1993.

[12] L. Kolev and V. Mladenov, "Use of interval slopes in implementing an interval method for global nonlinear DC circuits analysis," *Int. J.Circuit Theory Appl.*, vol. 25, pp. 37–42, 1997.

[13] L. V. Kolev, "An efficient interval method for global analysis of nonlinear resistive circuits," *Int. J. Circuit Theory Appl.*, vol. 26, pp. 81–92, 1998.

[14] A. N. Willson, "The no-gain property for networks containing three-terminal elements," *IEEE Trans. Circuits Syst.*, vol. CAS–22, pp. 678–687, 1975.

[15] M. Tadeusiewicz, "A method for finding bounds on all the dc solutions of transistor circuits," *IEEE Trans.Circuits Syst. I*, vol. 39, pp. 557–563, July 1992.

[16] H. Ratschek and J. Rokne, "Experiment using interval analysis for solving a circuit design problem," *J. Global Optimization*, vol. 3, pp. 501–518, 1993.

[17] L. V. Kolev, V. Mladenov, and S. Vladov, "Interval mathematics algorithms for tolerance analysis," *IEEE Trans.Circuits Syst.*, vol. 35, pp. 967–975, 1988.

[18] R. Krawczyk and A. Neumaier, "Interval slopes for rational functions and associated centred forms," *SIAM J. Numer. Anal.*, vol. 22, pp. 604–616, 1985.

[19] S. Zuhe and M. Wolfe, "On interval enclosures using slope arithmetic," *Appl. Math. Comput.*, vol. 39, pp. 89–105, 1990.

[20] E. Hansen, *Global Optimization Using Interval Analysis.* New York: Dekker, Inc., 1992.

[21] L. Kolev, "Use of interval slopes for the irrational part of factorable functions," *Reliable Comput.*, vol. 3, pp. 83–93, 1997.

[22] V. Mladenov and L. Kolev, "Interval methods for solving cellular neural networks (CNN's) equations," in *Proc. The Third Int. Conf. on Electronics, Circuits and Systems (ICECS'96)*, Rodos, Greece, Oct. 13–16, 1996.

[23] H. T. Nguyen, V. Kreinovich, V. Nesterov, and M. Nakamura, "On hardware support for interval computations and for soft computing: Theorems," *IEEE Trans. Fuzzy Syst.*, vol. 5, pp. 108–127, 1997.

[24] K. Yamamura, "An algorithm for representing functions of many variables by superpositions of functions of one variable and addition," *IEEE Trans. Circuits Syst. I*, vol. 43, pp. 338–340, Mar. 1996.

[25] R. Kearfott, C. Hu, and M. Novoa, "A review of preconditioners for the interval Gauss-Seidel method," *Interval Computat.*, vol. 1, pp. 59–85, 1991.

[26] R. B. Kearfott and K. Du, "The cluster problem in global optimization, the univariate case," *Comput. Supp.*, vol. 9, pp. 117–127, 1992.

[27] L. P. Huelsman, "Optimization—A powerful tool for analysis and design," *IEEE Trans. Circuits Syst. I*, vol. 40, pp. 431–439, 1993.

[28] A. B. Babichev, O. B. Kadyrova, T. P. Kashevarova, A. S. Leshchenko, and A. L. Semenov, "UniCalc, a novel approach to solving systems of algebraic equations," *Interval Computat.*, vol. 2, pp. 29–47, 1993.

[29] L. H. de Figueiredo and J. Stolfi, "Adaptive enumeration of implicit surfaces with affine arithmetic," *Comp. Graph. Forum*, vol. 15, pp. 287–296, 1996.

Lubomir Kolev received the Electrical Engineering degree, with distinction, from Moscow Power Institute, Moscow, Russia, in 1966, and the Ph.D. and D.Sc. degrees from the Higher Institute for Mechanical and Electrical Engineering, Sofia, Bulgaria, in 1972 and 1989, respectively.

From 1966 to 1968 he was an Engineer at the Institute for Technical Cybernetics, the Bulgarian Academy of Sciences. Since December 1968 he has been with the Electrical Engineering Department, Faculty of Automatica, the Technical University of Sofia, Sofia, Bulgaria, where he is currently a Full Professor. His research interests are mainly in the areas of nonlinear systems and circuit theory. He is the author of the monograph *Interval Methods for Circuit Analysis* (Singapore: World Scientific, 1993).

ISCAS 2000 - IEEE International Symposium on Circuits and Systems, May 28-31, 2000, Geneva, Switzerland

AN INTERVAL METHOD FOR GLOBAL INEQUALITY-CONSTRAINT OPTIMIZATION PROBLEMS

L. Kolev and D. Penev

Faculty of Automatica
Technical University of Sofia, Sofia 1756, BULGARIA
e-mail lkolev@vmei.acad.bg

ABSTRACT

An interval method is suggested for globally solving optimization problems of the following type: minimize a given objective function subject to both functional inequality constraints and simple bounds on the variables. The present method appeals to a new interval linearization of each nonlinear function and is based, essentially, on two computation techniques: linear programming and constraint propagation. The use of these techniques in the computational scheme of the present method seems to lead to improved performance as compared to other known interval methods of the same class.

1. INTRODUCTION

Interval methods (methods applying interval analysis techniques) are capable of globally solving various optimization problems with mathematical certainty [1]-[5]

In this paper, a new intervai method is suggested for global solution of the following inequ ity-constraint optimization problem

Minimize

$$\varphi_0(x) \qquad (1a)$$

subject to the constraints

$$\varphi_i(x) \le 0, \quad i = 1,2,\dots,r \qquad (1b)$$

$$x \in X^{(0)} \subset R^n \qquad (1c)$$

where x is a n-dimensional vector and $X^{(0)}$ is a given initial search region (a box). The function in (1a) is assumed continuously differentiable in $X^{(0)}$ while these in (1b) can be only continuous.

Various methods have been suggested for solving (1) globally [1]-[5]. Let $f(x)$ denote any function in (1). Known interval methods for solving (1) are all based on the linearization of f in X as follows:

$$F(X) = f(x^c) + \sum_{i=1}^{n} G_i(X)(X_i - x_i^c) \qquad (2)$$

where x_i^c are the components of the center x^c of the box $X = (X_1, \dots, X_n)$ with components X_i and $G_i(X)$ is the interval extension of the derivative $g_i(x) = \partial f / \partial x_i$ or , better, the interval extension of the corresponding slope [3] in X (other more sophisticated extensions of the type (2) have also been suggested and used [1]-[5]). An elaborate algorithm for solving (1) by applying (2) is presented in [3]. It is, however, too complicated for practical purposes. Indeed, it involves as an integral part a procedure for solving a nonlinear system derived on account of the so-called Fritz-John conditions [3]. Since (1c) is in fact the short notation of $2n$ inequalities:

$$x_i - \overline{x}_i \le 0, \quad i = 1,\dots,n \qquad (3a)$$

$$\underline{x}_i - x_i \le 0, \quad i = 1,\dots,n \qquad (3b)$$

where $\underline{x}_i$ and $\overline{x}_i$ are the end-points of X_i, the Fritz-John system will be a system of m equations in m unknowns with $m = 3n + r + 1$. To reduce the amount of computation required, an approach called "peeling of the boundary" which skips the inequalities (3) has been proposed in [5]. However, the simpler problem (1a), (1b) is to be solved 3^n times which is still prohibitively inefficient for larger n.

Recently, a new approach to constructing interval methods for global optimization has been proposed in [6]. It is based on a new type of linear interval linearization of $f(x)$ in X in the form:

$$F(X) = \sum_{i=1}^{n} \alpha_i X_i + B, \quad x_i \in X_i \qquad (4)$$

where α_i are real numbers and only B is an interval.

In this paper, a method for solving (1) is suggested. It appeals to linearization (4) and uses, essentially, the following two techniques: linear programming and constraint propagation. The combined effect of the use of these techniques in the computational scheme of the new method seems to lead to improved performance as compared to other known interval methods of the same class.

2. LINEAR PROGRAMMING APPROACH

In this section, we formulate the linear programming (LP) problem to be incorporated into the computational scheme of the new combined method.

Let $X \subseteq X^{(0)}$ and

$$g(x) = 0, \quad x \in X \qquad (5)$$

where $g(x) = g_1(x), \dots, g_n(x))$ is the gradient of $\varphi_0(x)$. The LP problem to be set up and solved at each iteration of the optimization method is related to solving (5) and will be presented in the form of the following subroutine.

Subroutine 1.

It is assumed that using the approach of [7]-[10] the functions $g_i(x)$ of (5) have been linearized by the form (4).

Step 1. Form the (corresponding to X) real matrix $A = \{-a_{ij}\}$ and the interval vector B and set up the system

$$-Ax + B = 0\,, \quad x \in X \tag{6}$$

"Correct" B (if possible) by computing

$$B' = AX\,, \quad B = B' \cap B \tag{7}$$

Step 2. (Start of the choice of the objective function for the LP problem) Compute

$$y_c = Ax_c\,, \quad d = y_c - b_c$$

(x_c and b_c are the centers of X and B, respectively) and find

$$d_0 = \max\{|d_i|, \quad i = 1,\dots n\}$$

and the corresponding index $l_0 = k$. So

$$d_k = d_0 sign(d_k)\,.$$

If $d_k < 0$ then go to next step, else go to step 4.

Step 3. (Continuation of the choice of the LP objective function and construction of the LP problem). Form the following LP problem

$$f = b_k' = \sum_{i=1}^{n} c_i x_i = \max \tag{8a}$$

$$c_i = a_{ki}\,, \quad i = 1,\dots,n \tag{8b}$$

$$-\sum_{j=1}^{n} a_{ij} x_j + b_i = 0\,, \quad i \neq k\,, \quad i = 1,\dots,n \tag{8c}$$

$$x_i \in X_i \quad , \quad b_i \in B_i \tag{8d}$$

Go to step 5.

Step 4. In this case

$$f = b'' = \sum_{i=1}^{n} c_i x_i = \min \tag{9a}$$

and (9b), (9c) and (9d) are as in (8). However, as is well known, problem (9) can be written equivalently as a max problem (8). Thus, the solution of (9) can be carried out in the same way as in the previous step.

Step 5. (Discarding X). Start solving LP problem (8) (or (9)) using the dual LP method. In this method, f for problem (8) is reached from above, i.e.

$$f_l \geq f\,, \quad l = 0, 1, \dots \tag{10}$$

where f_l is the value of the objective function at the lth LP iteration. Thus, if at some iteration (for $B_k = [\underline{B}_k, \overline{B}_k]$):

$$f_l < \underline{B}_k \tag{11}$$

then obviously the maximum value

$$f < \underline{B}_k \tag{12}$$

This means that the initial system (6) is incompatible. Also, if at some iteration, the second method shows incompatibility of problem (8), the iterations are stopped and the current X is discarded.

Step 6. (Reducing B_k). If (dealing with (8)) we reach the maximum value f, then obviously

$$f = b_k' < \overline{B}_k \tag{13}$$

and we can update (reduce) B_k by putting

$$B_k = [\underline{B}_k, b_k'] \tag{14a}$$

It is easily seen that in case of problem (9) B_k is updated as follows

$$B_k = [b_k'', \overline{B}_k] \tag{14b}$$

Using constraint propagation we can now reduce X as will be shown in the next section.

3. CONSTRAINT PROPAGATION

This technique has already been applied in the context of solving systems of nonlinear equations using linearization (2) [11] or linearization (4) [9].

For the purpose of global optimization we apply the constraint propagation (CP) technique in the following two cases.

A) *In solving system* (5).

In this case, the CP approach is applied when the LP algorithm terminates in Step 6. We start with updating vector B as shown in (14). We then continue with the following subroutine (as in algorithm A6 of [9])

Subroutine 2.

For $i = 1$ to n do, for $j = 1$ to n do

$$Y_j = \frac{1}{a_{ij}}[B_i - \sum_{k \neq j}^{n} a_{ik} X_k] \tag{15a}$$

$$X_j := Y_j \cap X_j \tag{15b}$$

As a result the current box is reduced in size.

B) *In handling inequalities* (1b)

Subroutine 3.

Each inequality (1b) is linearized in the form (4) to get

$$L_i(X) = \sum_{j=1}^{n} a_{ij} x_j + B_i \le 0, \quad x_j \in X_j \tag{16}$$

Now for j=1 to n we solve the inequality

$$a_{ij} Y_j + S_j \le 0 \tag{17a}$$

for Y_j as shown in [3] with

$$S_j = \sum_{\substack{k=1 \\ k \ne j}}^{n} a_{ij} X_j + B_i \tag{17b}$$

For each j, we update

$$X_j := Y_j \cap X_j \tag{17c}$$

and use this updated interval in (17b) when $j > 1$.

This process is repeated for all $i = 1, \dots, r$.

4. ALGORITHM OF THE METHOD

The algorithm of the present iterative method is based on the execution, at each iteration, of several basic procedures.

Let $X \subsetneq X^{(0)}$ be the current box to be analyzed. If X is strictly feasible [3], then the following two procedures are carried out (otherwise the iterative process continues with Procedure 3 to be presented below).

Procedure 1.

(Monotonicity test [1], [3]). If for some i

$$G_i(X) > 0 \ \text{ or } \ G_i(X) < 0 \tag{18}$$

where $G_i(X)$ is the interval extension of $\partial\varphi_0/\partial x_i$ then X is discarded. To narrower $G_i(X)$ and, hence, to improve the effectiveness of the monotonicity test, $G_i(X)$ are computed using optimal poles [4].

Procedure 2.

Let $\bar{\varphi}_0$ denote an upper bound on the global minimum φ_0^*, obtained at a previous iteration (at the first iteration $\bar{\varphi}_0$ is set to $+\infty$). Now we compute $\varphi_0(x^c)$ where x^c is the center of X and update $\bar{\varphi}_0$, i.e.

$$\bar{\varphi}_0 = \varphi_0(x^c), \quad \text{if } \varphi_0(x^c) < \bar{\varphi}_0, \tag{19}$$

otherwise $\bar{\varphi}_0$ remains unchanged.

Procedure 3.

(Extended unfeasibility test)

Let

$$\varphi_0(X) := \varphi_0(X) - \bar{\varphi}_0 \tag{20}$$

If for some i from $0, 1, \dots, r$

$$\Phi_i(X) > 0 \tag{21}$$

the current box X is discarded.

If one of the above three tests (18), (19) or (21) is satisfied, a new box X is retrieved from a queue Q of boxes to be processed and the iterative process continues with checking feasibility. Otherwise, we proceed to the following procedure.

Procedure 4.

We call Subroutine 1 to set up and solve LP problem (8) (or (9)).

Procedure 5.

Call Subroutine 2 to apply constraint propagation (CP) to a system of equalities.

Procedure 6.

Call Subroutine 3 to apply CP to a system of inequalities.

Remark. The remaining steps of the algorithm of the present method such as control of the accuracy, splitting the current box whenever necessary etc. are omitted for space reasons.

5. NUMERICAL EXAMPLES

The examples given below have been solved by the following two methods:

- method M1 which is based on constraint propagation using the mean-value form (2) as linearization of each nonlinear equation;

- the present method (denoted as method M2) which appeals to the alternative linearization (4), linear programming and constraint propagation.

The data about the results include: number of splits (denoted as S), number of evaluations of all nonlinear functions (denoted as F), number of all interval gradients (if using M1) or all interval forms (4) (for M2) (denoted by the same symbol G since form (4) requires approximately the same amount of computation as interval gradients) and computer time T in seconds needed to solve the global optimization problem considered within desired accuracy. The examples were run on a relatively slow 166 MHz PC using an interpreter of EXCEL.

Example 1.

The function to be minimized is [3]

$$\varphi_0(x) = x_1^6 - 6.3x_1^4 + 12x_1^2 + 6x_1 x_2 + 6x_2^2 \tag{22a}$$

subject to the constraints

$$\begin{aligned} \varphi_1(x) &= 1 - 16x_1^2 - 25x_2^2 \le 0 \\ \varphi_2(x) &= 13x_1^3 - 145x_1 + 85x_2 - 400 \le 0 \\ \varphi_3(x) &= x_1 x_2 - 4 \le 0 \end{aligned} \tag{22b}$$

For this example, the absolute accuracy (upper bound minus lower bound on the global minimum) was chosen to be 10^{-5}. We give data about S, F, G and T for two initial boxes whose side are [-2, 4] and [-10^5, 10^5], respectively:

Table 1

	S	F	G	T
M1	1050	8411	8397	7
M2	1050	2115	8437	6
M1	2713	21703	21653	18
M2	2381	4759	19038	14

Example 2.

In this example [12]

$$\varphi_0(x) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 +$$

$$10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7$$

$$\varphi_1(x) = 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 - 127 \le 0$$

$$\varphi_2(x) = 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 - 282 \le 0$$

$$\varphi_3(x) = 23x_1 + x_2^2 + 6x_6^2 - 8x_7 - 196 \le 0$$

$$\varphi_4(x) = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \le 0$$

For this example, the accuracy measure was the relative accuracy (absolute accuracy over absolute value of the mean of upper and lower bounds) and was chosen to be 0.01.

Table 2

	S	F	G	T
IM2	11887	118727	118727	292
IM2	8074	16146	80738	143

The above two examples (as well as others not reported here) confirm the expected effect of enhanced numerical efficiency of the present method as compared with other interval methods of the same class.

6. SUMMARY

A new interval method for globally solving inequality-constrained optimization problems defined as in (1) has been suggested. The method has one major feature which distinguishes it favorably from other known interval methods for global optimization. It consists in the fact that the interval extensions $\Phi_i(X)$ of the functions $\varphi_i(x)$, $i = 0,...,r$, associated with the optimization problem, are determined in a new way, using the recently suggested linear interval form (4). The new form differs from the known form (2) in that the coefficients α_i before the variables x_i are real numbers and only the additive term B is an interval while in (2) all the coefficients are intervals and only the additive term is a real number.

The specific feature of form (4) has permitted the development of two techniques aiming to speed up the computation: linear programming (in solving the associated system of nonlinear equations (5)) and constraint propagation (for handling both (5) and the system of nonlinear inequalities (1b)). Experimental evidence seems to indicate that the new method's performance is better than methods based on the traditional form (2).

An improvement of the numerical efficiency seems possible by generalizing the linear programming approach to the system of nonlinear inequalities associated with the optimization problem considered.

7. REFERENCES

[1] Moore R. *Methods and applications of interval analysis*, SIAM, Philadelphia, 1979.

[2] Ratschek H. and Rokne J. *New computer methods for global optimization*, Chichester: Horwood, 1988.

[3] Hansen E. *Global optimization using interval analysis*, New York, Marcel Dekker, Inc., 1992.

[4] Kolev L. *Interval methods for circuit analysis*, Singapore, World Scientific, 1993.

[5] Kearfort R. *Rigorous global search: continuous problems*, Kluwer academic, Dordrecht, 1996.

[6] Kolev, L. "A new interval approach to global optimization", *Computational Mathimatics and Mathematical Physics*, Vol. 39, No. 5, pp.727-737, 1999.

[7] Kolev, L. "An efficient interval method for global analysis of nonlinear resistive circuits". *Int. J. of Circuit Theory and Appl.*, Vol. 26, pp. 81-92, 1998.

[8] Kolev, L. "A new method for global solution of systems of nonlinear equations". *Reliable Computing*, Vol. 4 No 2, pp. 125-146, 1998.

[9] Kolev, L. "An improved method for global solution of nonlinear systems". *Reliable Computing*, Vol. 5, No 2, pp. 103-111, 1999.

[10] Kolev, L. "Automatic computation of a linear interval enclosures" (to appear in *Reliable Computing*).

[11] Hentenryck, P., and Mcallester, D., and Kapur, D. "Solving polynomial systems using a branch and prune approach", *SIAM J. Numer. Anal.*, vol. 34, No2, pp. 797-827, April 1997.

[12] Hentenryck, P., et al. *NUMERICA - a modeling language for global optimization*, MIT Press, 1997.