Reliable Computing **4:** 125–146, 1998.

A New Method for Global Solution of Systems of Non-Linear Equations

LUBOMIR V. KOLEV
Dept. of Theoretical Electrotechnics, Faculty of Automatica, Technical University of Sofia, 1756 Sofia, Bulgaria, e-mail: LKOLEV@vmei.acad.bg

(Received: 5 May 1997; accepted: 8 October 1997)

Abstract. In this paper the problem of finding the set of all real solutions to a system of n non-linear equations contained in a given n-dimensional box (the global solution problem) is considered. A new method for solving the global solution problem is suggested. It is based on a transformation of the original system into a larger system of separable form. The global solution of the latter system is then found in a most efficient manner by a new interval method which exploits the separabily property. Numerical examples illustrating the efficiency of the method suggested are provided.

1. Introduction

Let $\psi : D \subset R^n \to R^n$ be a continuously differentiable function and let $X^0 = (X_1^0, \ldots, X_n^0) \subset D$ be a given interval vector (a box). The following global solution problem (GS problem) has been considered in a number of publications.

THE GS PROBLEM. *Given ψ and X^0, find the set $S(\psi, X^0) \triangleq (x^{(1)}, \ldots, x^{(p)})$ of all real solutions (zeros) to the system of equations*

$$\psi(x) = 0 \tag{1.1a}$$

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

$$x \in X^{(0)}. \tag{1.1b}$$

Presently, interval methods (methods based on interval analysis techniques) seem to be the only methods which are capable of infallibly solving the GS problem (see [17] and the references therein cited). Indeed, on termination they provide a set of small boxes $X^{(s)}$, $s = 1, \ldots, P$ with $P \geq p$ which contains the set $x^{(s)}$, $s = 1, \ldots, p$. If no clustering is observed, $P = p$ and each $X^{(s)}$ provides guaranteed bounds on the solution $x^{(s)}$ even in the presence of roundoff errors.

However, 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. Thus, for a system of 9 equations and a relatively small initial region X^0 with $X_i^0 = [0, 10]$ a rather sophisticated method suggested in [17] requires billions of function evaluations in interval form to locate the unique zero of the GS problem considered. The authors' comment:

"There were too many function evaluations in order to be proud of reaching the goal and to be really successful" speaks for itself.

The fact that known interval methods are rather inefficient for even moderate values of n is not difficult to explain. There seem to be 4 basic reasons for this.

1. Since known interval methods for solving the GS problem are all one or another form of the interval Newton method some interval extension $J^I(X)$ of the Jacobian $J(x)$ of $\psi(x)$ in X is needed at each iteration of the method; here $X \subseteq X^{(0)}$ denotes the box currently processed. At the initial iterations X is large and $J^I(X)$ is generally rather a crude overestimation of the range $J(X)$ of $J(x)$ in X.

2. All methods of this group are associated with solving a linear interval system

$$A^I(X)(y - x) = b(x) \tag{1.2}$$

with respect to y for each iteration. Here $b(x)$ is a real vector (ignoring for simplicity of exposition the interval arithmetic implementation of the method considered), x is usually the centre of X while $A^I(X)$ is an interval matrix which is either the interval extension $J^I(X)$ itself or is in one way or another related to it. For instance, in methods using preconditioning [3]

$$A^I(X) = BJ^I(X), \quad b(x) = -B\psi(x) \tag{1.3a}$$

where B is some real matrix. In Hansen's method [2]

$$B = [J(x)]^{-1}. \tag{1.3b}$$

Since the exact (optimal) interval solution Y of (1.2) is extremely hard to determine, in practice an approximate interval solution $Y^I \supset Y$ is found which is once again, rather a crude overestimation of Y. Indeed, most often a componentwise Gauss-Seidel procedure is used to compute Y^I and it can be easily seen that the overestimation of Y by Y^I becomes more and more pronounced as n increases. Now let $A(X)$ denote the interval matrix associated with the range $J(X)$, i.e.

$$A(X) = BJ(X).$$

Obviously, this is the narrowest possible interval matrix for the current box which can replace $A^I(X)$ in (1.2). Furthermore, let $\tilde{Y}$ denote the optimal interval solution of the "best" linear interval system

$$A(X)(y - x) = b(x). \tag{1.4}$$

On account of inclusion monotonicity

$$\tilde{Y} \subseteq Y \subseteq Y^I \tag{1.5}$$

and the inclusion is proper and rather pronounced. Hence, initially, most often

$$X \subset Y^I \tag{1.6}$$

and therefore the new box

$$X' = Y^I \cap X \tag{1.7}$$

generated by the method for the next iteration is the same as the box X at the current iteration. In this case X is to be splitted into two subboxes X_L and X_R. One of them is stored into a list L for further processing while the other is renamed X and the iterative process continues on X.

3. The above two factors substantiate the third cause for low efficiency: there are too many splittings at the early stages of the computation process. This, in turn, gives rise to a long queue of boxes X^v stored in L and awaiting processing. In fact, splitting stops and reducing the size of X by (1.7) starts only at that moment when the width of X becomes so small that the first two factors associated with overestimation become insignificant.
4. Overestimation accounts for the ineffective functioning of the exclusion rule

$$Y^I \cap X = \emptyset \tag{1.8}$$

which deletes the current box X from further processing. Similarly to reducing, deleting occurs only at the later iterations when the width of X becomes small enough and the effects of overestimation become negligible.

It is to be noted that less effective reducing and deleting rules lead to a stronger clustering effect.

An attempt to reduce the adverse effect of the first factor due to overestimation is associated with using interval slopes ([6], [11], [12], [20]) rather than interval derivatives in evaluating $J^I(X)$. However, since this leaves unchanged the second factor for low efficiency, further experimental evidence showed that the overall improvement of the interval methods based on the use of interval slopes is still not satisfying, especially for more complex problems of larger n [9], [12].

In this paper, a new method for tackling the GS problem which seems to have a considerably better numerical performance is suggested. It is based on the following approach. First, the original system (1.1) is transformed into a larger system of n' non-linear equations

$$f(x) = 0; \tag{1.9a}$$

$$x \in X^{(0)} \in R^{n'} \tag{1.9b}$$

in such a way that the resulting system (1.9a) is in separable form, i.e.

$$f_i(x) = \sum_{j=1}^{n'} f_{ij}(x_j), \quad i = 1, \ldots, n'. \tag{1.10}$$

This transformation has recently become possible due to a result obtained in [19]. Then a new interval method is applied to system (1.9). The global solution of (1.9) provides the global solution of the original problem (1.1). The new interval method

exploits in a rather efficient manner the separability property (1.10) and reduces, essentially, to solving the following linear interval system

$$A^{(\nu)}y = B^{(\nu)}(X) \tag{1.11}$$

at each iteration ν. Now, unlike (1.2) $A^{(\nu)}$ is a real matrix while $B^{(\nu)}(X)$ is an interval vector. In contrast to (1.2) the optimal interval solution $\tilde{Y}$ to (1.11) is readily obtained:

$$\tilde{Y} = [A^{\nu}]^{-1}B^{\nu}(X). \tag{1.12}$$

This is the main feature of the new interval method which accounts for its computational superiority over the previously known interval methods. Indeed, using $\tilde{Y}$ (instead of Y^I as in the previous methods) results in a faster rate of convergence of the new method.

The paper is organized as follows. Section 2 presents the new method for solving systems of separable form. A numerical example of a system with $n = 10$ is solved. The method is extended to systems of arbitrary form in Section 3. The overall efficiency of the resulting method is illustrated by the example considered in [17]. Concluding remarks are given in the last section.

2. Solving Separable Form Systems

2.1. The New Method

We consider the GS problem related to (1.9) with f satisfying the separability property (1.10). Several methods ([4], [5], [7]–[9], [12]) have been proposed for solving this problem for the special case where $f_{ij}(x_j) = a_{ij}x_j$ is a linear function for $j \neq i$ and $f_{ii}(x_i)$ are assumed continuously differentiable (CD) functions.

A new method for globally solving the separable form system (1.9) will be presented here. In its present statement no restrictions on the functions $f_{ij}(x_j)$ are imposed except for the requirement that they be continuous in $X_j^{(0)}$.

Let X denote the subbox processed at the current iteration. First, a new interval approximation of a component $f_{ij}(x_j)$ in X_j will be suggested. Unlike previous methods where the functions $f_{ij}(x_j)$ are approximated in X_j by enclosures using interval derivatives ([4], [5], [7], [8]) or interval slopes ([9], [12]), the new approximation is chosen in the following form

$$L_{ij}(X_j) = B_{ij} + a_{ij}x_j, \quad x_j \in X_j \tag{2.1}$$

where $B_{ij} = [\underline{b}_{ij}, \overline{b}_{ij}]$ is an interval while a_{ij} is a real number. Both B_{ij} and a_{ij} are to be determined such that the following inclusion property should hold

$$f_{ij}(x_j) \in B_{ij} + a_{ij}x_j, \quad x_j \in X_j. \tag{2.2}$$

A simple and efficient procedure for finding a_{ij}, $\underline{b}_{ij}$, and $\overline{b}_{ij}$ is suggested here for the case of CD functions. It is motivated by elementary geometrical considerations (Figure 1a) and can be readily adapted for the case of functions that are only continuous (Figure 1b).

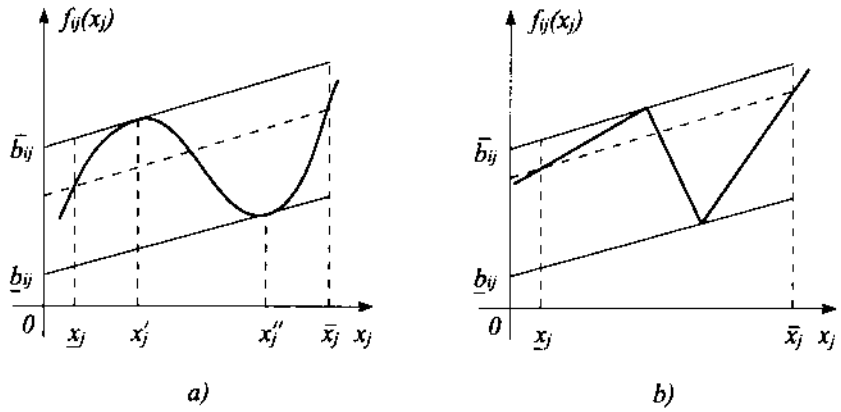

Figure 1. Geometrical illustration of the linear interval approximation of $f_{ij}(x_j)$ in $X_j = [\underline{x}_j, \bar{x}_j]$: a) for the case of a continuously differentiable function, b) for the case of a piecewise-linear functions.

PROCEDURE 1. First, compute

$$\underline{f}_{ij} = f_{ij}(\underline{x}_j), \quad \bar{f}_{ij} = f_{ij}(\bar{x}_j). \tag{2.3}$$

Then a_{ij} is defined as the slope

$$a_{ij} = (\bar{f}_{ij} - \underline{f}_{ij}) / (\bar{x}_j - \underline{x}_j). \tag{2.4}$$

Let

$$l_1(x_j) = \bar{b}_{ij} + a_{ij}x_j \tag{2.5a}$$

be a straight line such that

$$f_{ij}(x_j) \leq l_1(x_i), \quad x_j \in X_j. \tag{2.5b}$$

Thus, l_1 is the lowest possible line of slope a_{ij} bounding f_{ij} from above in X_j (see Figure 1a). Similarly, let the straight line

$$l_2(x_j) = \underline{b}_{ij} + a_{ij}x_j \tag{2.6a}$$

have the property

$$f_{ij}(x_j) \geq l_2(x_j), \quad x_j \in X_j. \tag{2.6b}$$

Thus, l_2 is the highest possible line of slope a_{ij} bounding f_{ij} from below in X_j (see Figure 1a).

The unknown constants $\underline{b}_{ij}$ and $\bar{b}_{ij}$ are determined as follows. Let $p_{ij}(x_j)$ denote the derivative of $f_{ij}(x_j)$, i.e. $p_{ij}(x_j) = \dfrac{\mathrm{d}f_{ij}}{\mathrm{d}x_j}(x_j)$. Then the following functional equation of one variable

$$p_{ij}(x_j) - a_{ij} = 0 \tag{2.7}$$

is solved globally in X_j. Let $x_j^{(k)}$, $k = 1, 2, \ldots, K$, denote the k-th solution of (2.7) in X_j; furthermore, let $x_j^{(0)} = \underline{x}_j$. Now compute the quantities

$$b_{ij}^{(k)} = f_{ij}(x_j^{(k)}) - a_{ij}x_j^{(k)}, \quad k = 0, 1, \ldots, K. \tag{2.8}$$

As is easily seen from geometrical considerations (Figure 1a)

$$\underline{b}_{ij} = \min\{b_{ij}^{(k)},\ k = 0, 1, \ldots, K\} \tag{2.9a}$$

and

$$\overline{b}_{ij} = \max\{b_{ij}^{(k)},\ k = 0, 1, \ldots, K\}. \tag{2.9b}$$

Finally, the interval

$$B_{ij} = [\underline{b}_{ij}, \overline{b}_{ij}] \tag{2.10}$$

is found. Thus, the linear interval approximation (2.1) having the enclosing property (2.2) has been determined.

Remark. For simplicity of notation and presentation, the quantities $\underline{f}_{ij}$, $\overline{f}_{ij}$, a_{ij}, $x_{ij}^{(k)}$, $b_{ij}^{(k)}$, $\underline{b}_{ij}$, and $\overline{b}_{ij}$, are assumed to be real numbers. It should, however, be borne in mind, that in the actual implementation of Procedure 1 they must be computed using interval arithmetic. Therefore, they are in fact intervals although their widths are rather small. Rigorous bounds on all zeros of equation (2.7) in X_j can be computed using an appropriate interval algorithm (e.g., see [1], Ch. 7 and 8).

A simple illustrative example of the procedure for determining $\underline{b}_{ij}$ and $\overline{b}_{ij}$ will be considered now. Assume that $f_{ij}(x_j)$ is a cubic equation in x_j. Then (2.7) is a quadratic equation. Let x_j' and x_j'' denote the corresponding zeros. Several cases are possible depending on whether both zeros belong to X_j or not.

Case A. Both zeros belong to the interior of the interval X_j as shown in Figure 1a. Now computing $b_j^{(k)}$, $k = 0, 1, 2$ by (2.8) we see that in this case

$$\overline{b}_{ij} = b_{ij}^{(1)} = f_{ij}(x_j') - a_{ij}x_j'$$

while

$$\underline{b}_{ij} = b_{ij}^{(2)} = f_{ij}(x_j'') - a_{ij}x_j''.$$

Case B. The first zero belongs to X_j while the second does not. In this case, we need to compute only $b_{ij}^{(0)}$ and $b_{ij}^{(1)}$ since now $k = 0, 1$; thus

$$\overline{b}_{ij} = b_{ij}^{(1)} = f_{ij}(x_j') - a_{ij}x_j'$$
$$\underline{b}_{ij} = b_{ij}^{(0)} = f_{ij}(\underline{x}_j) - a_{ij}\underline{x}_j.$$

Case C. Only the second zero belongs to X_j. It is easily seen that in this case $\overline{b}_{ij} = b_{ij}^{(0)}$ and $\underline{b}_{ij} = b_{ij}^{(2)}$.

Using the Procedure 1 a real matrix is introduced

$$A = \{-a_{ij}\} \tag{2.11}$$

and an interval vector $B = (B_1, \ldots, B_n)$ is formed with

$$B_i = \sum_{j=1}^{n} B_{ij} = [\underline{b}_{ij}, \overline{b}_{ij}]. \tag{2.12}$$

On account of (1.10), (2.1), (2.2), (2.11), and (2.12)

$$f_i(x) \in \sum_{j=1}^{n} a_{ij}x_j + B_i, \quad x \in X, \quad i = 1, \ldots, n \tag{2.13}$$

or in vector form

$$f(x) \in -Ax + B, \quad x \in X. \tag{2.14}$$

If y is a solution of (1.1) in X, then $f(y) = 0$ and by (2.14)

$$0 \in -Ay + B, \quad y \in X. \tag{2.15}$$

Now we can state the main result of the section.

THEOREM 2.1. *All the solutions y to*

$$f(x) = 0 \tag{2.16}$$

contained in X are also contained in the solution set $S(X)$ of the system

$$-Ax + b = 0, \quad b \in B \tag{2.17}$$

where b is any real vector contained in B.

The proof is straightforward on account of (2.14) to (2.17).

Since B is an interval vector the set $S(X)$ is a convex polyhedron. Indeed, (2.17) is in fact a system of n linear equalities and $2n$ two-sided linear inequalities.

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

THEOREM 2.2. *All the solutions y to (2.16) in X are also contained in the intersection*

$$P(X) = S(X) \cap X. \tag{2.18}$$

Since $S(X)$ and X are convex polyhedra it is seen from (2.18) that $P(X)$ is also a convex polyhedron.

COROLLARY 2.1. *If $P(X)$ is empty, i.e. if*

$$S(X) \cap X \neq \emptyset \tag{2.19}$$

the system (2.16) has no solution in X.

Let $H(P,X)$ denote the interval hull of $P(X)$, that is the smallest interval vector (box) containing $P(X)$. Consider the following iteration procedure

$$X^{(k+1)} = H(P, X^{(k)}) \cap X^{(k)}, \quad k \geq 0. \tag{2.20}$$

Procedure (2.20) could be used for designing a method for finding all real solutions to (1.1) in $X^{(0)}$. Such an approach seems to be rather costly since $2n$ linear programming problems are to be solved at each iteration to determine $H(P,X^{(k)})$. Therefore, a simpler and, presumably, more efficient procedure is suggested here.

Let $H(S,X)$ denote the interval hull of $S(X)$. Then it follows from (2.17) that $H(S,X)$ is given by the formula

$$H(S,X) = A^{-1}B. \tag{2.21}$$

Let $C = A^{-1}$ (assuming A invertible) and $Y = H(S,X)$. It follows from (2.21) that the components $Y_i = [\underline{Y}_i, \overline{Y}_i]$ of Y are given by the formulae

$$\underline{Y}_i = \sum_{j=1}^{n} \underline{y}_{ij} \tag{2.22}$$

with

$$\underline{y}_{ij} = \begin{cases} c_{ij}\underline{b}_j, & \text{if } c_{ij} \geq 0; \\ c_{ij}\overline{b}_j, & \text{if } c_{ij} < 0; \end{cases} \tag{2.23}$$

$$\overline{Y}_i = \sum_{j=1}^{n} \overline{y}_{ij} \tag{2.24}$$

with

$$\overline{y}_{ij} = \begin{cases} c_{ij}\overline{b}_j, & \text{if } c_{ij} \geq 0; \\ c_{ij}\underline{b}_j, & \text{if } c_{ij} < 0. \end{cases} \tag{2.25}$$

Now the following iterative procedure can be used as the basis for solving the GS problem considered.

PROCEDURE 2. Let $X^{(k)}$ be a current box. Using Procedure 1 determine $C^{(k)}$ and $B^{(k)}$ corresponding to $X^{(k)}$. By formulae (2.22) to (2.25) compute $Y^{(k)}$. The iterative procedure is then defined as follows

$$X^{(k+1)} = Y^{(k)} \cap X^{(k)}, \quad k \geq 0. \tag{2.26}$$

The procedure may result in three outcomes.

A. The sequence $X^{(k+1)}$ converges to a solution $x^{(s)}$ as k increases. Actually, the iterations are stopped whenever the width of $X^{(k+1)}$ becomes smaller than a constant ε_1 (accuracy with respect to x). Now $x^{(s)}$ is approximated by the centre x^c of $X^{(k+1)}$ and x^c is substituted in (1.9a). If

$$\text{tol} = \max_i [|f_i(x^c)|, \ i = 1, \ldots, n] > \varepsilon_2 \tag{2.27}$$

(ε_2 is the accuracy of x^c with respect to the system of equations) then the iterations are resumed; otherwise x^c is accepted as a solution to (1.9).

B. At some k

$$Y^{(k)} \cap X^{(k)} = \emptyset. \tag{2.28}$$

Since $Y = H(S, X) \supseteq S(X)$ it follows from Corollary 2.1 that system (2.16) has no solution in $X^{(k)}$ if (2.28) becomes valid. In this case $X^{(k)}$ is discarded from further consideration.

C. The sequence $X^{(k+1)}$ converges to a fixed interval (box) X^*. In practice, the procedure is stopped whenever the reduction in the volume $V(X^{(k+1)})$ of the current box $X^{(k+1)}$ as compared to that of the preceding box $X^{(k)}$ is smaller then a constant ε_3, i.e. if $V(X^{(k+1)}) > \varepsilon_3 V(X^{(k)})$. In this case $X^{(k+1)}$ is split along its widest side into two boxes X^L and X^R (left and right). The right box is stored into a list L for further processing. The left box is remained $X^{(0)}$ and the iterative procedure (2.22) to (2.25) is resumed.

Procedure 2 is based on vector operations. Its convergence can be improved if componentwise operations are introduced. Thus, whenever a reduction of a component $X_i^{(k+1)}$ occurs, this will be used immediately for reducing (if possible) the remaining components $X_j^{(k+1)}$, $j = i+1, \ldots, n$.

To introduce this componentwise algorithm we need to modify Procedure 1 and Procedure 2 to Procedures 3 and 4, respectively.

PROCEDURE 3. In this procedure, all the computations in Procedure 1 are carried out for a fixed j and the corresponding X_j. Thus, on exit from Procedure 3 we have the j-th column A_j of A and an interval vector B_j whose components are B_{ij}, $i = 1, \ldots, n$.

For reasons to become clear later, it is expedient to introduce two real matrices B_l and B_u and to store $\underline{b}_{ij}$ and $\overline{b}_{ij}$ in the j-th row of B_l (lower end-points) and B_u (upper end-points), respectively.

To introduce Procedure 4, formula (2.26) is written in componentwise form

$$X_i^{(k+1)} = Y_i^{(k)} \cap X_i^{(k)}. \tag{2.29}$$

Now, according to the idea to get the most out of the reduction of $X_i^{(k+1)}$ we check whether

$$X_i^{(k+1)} \subset X_i^{(k)}. \tag{2.30}$$

If (2.30) holds (in practice if the reduction is greater than some threshold ε_4, i.e. if $w(X_i^{(k+1)}) < \varepsilon_4 w(X_i^{(k)})$) Procedure 3 is immediately called to recompute the corresponding i-th column A_i of A, and the i-th columns B_i^l and B_i^u of B^l and B^u, respectively. Let these updated vectors are denoted $\tilde{A}_i$, $\tilde{B}_i^l$, and $\tilde{B}_i^u$. Similarly let $\tilde{A}^{(k)}$, $\tilde{C}^{(k)}$, and $\tilde{B}^{(k)}$, designate the corresponding updated matrices $A^{(k)}$, $C^{(k)}$ and interval vector $B^{(k)}$. Now, the lower and upper end-points of $\tilde{B}^{(k)}$ are easily evaluated:

$$\underline{\tilde{B}}^{(k)} = \underline{B}^{(k)} - B_i^l + \tilde{B}_i^l; \tag{2.31a}$$

$$\overline{\tilde{B}}^{(k)} = \overline{B}^{(k)} - B_i^u + \tilde{B}_i^u. \tag{2.31b}$$

THEOREM 2.3. *Let $f : D \subseteq R^n \to R^n$ be a continuously differentiable function in the domain D and $X^{(0)} \subset D$. Introduce the interval operator*

$$K(X^{(k)}) = H(S^{(k)}, X^{(k)}) = A^{-1}(X^{(k)})B(X^{(k)}), \quad k \geq 0 \tag{2.34}$$

where $A(X^{(k)})$ and $B(X^{(k)})$, defined in Procedure 1, correspond to the current box $X^{(k)}$. Then, if at some k-th iteration

$$K(X^{(k)}) \subseteq X^{(k)} \tag{2.35}$$

the above inclusion implies the existence of a solution to (2.15) in $X^{(k)}$.

Proof. Let y be a solution to (2.15). Then (2.15) can be transformed into the fixed point format

$$x = C(x)b(x) = P(x) \tag{2.36}$$

where $C : X^{(k)} \to R^n$ and $b : X^{(k)} \to R^n$ are to be determined. We choose

$$C(x) \triangleq C^{(k)} = A^{-1}(X^{(k)}); \tag{2.37a}$$

$$b(x) \triangleq C^{(k)} = b \in B(X). \tag{2.37b}$$

Obviously, for any $x \in X^{(k)}$

$$P(x) \in K(X^{(k)}). \tag{2.38}$$

Thus, if (2.35) 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 $f(x) = 0$ has a solution in $X^{(k)}$. □

The theorem can be extended in a straightforward manner to the componentwise version of the present method as introduced by Procedure 5.

Now we shall consider the convergence rate of the sequence (2.26) towards a solution. We need the following well-known facts from interval analysis. For an interval $[a, b]$ $(a \leq b)$, and an interval vector $X = (X_1, \ldots, X_n)$ define the width of $[a, b]$ and X as follows

$$w([a, b]) = b - a; \tag{2.39a}$$

$$w(X) = \max_i w(X_i). \tag{2.39b}$$

If

$$Y = AX$$

where A is a real matrix with elements a_{ij} then

$$w(Y_i) = \sum_{j=1}^{n} |a_{ij}| w(X_j) \tag{2.40a}$$

and

$$w(Y) \leq \|A\| w(X) \tag{2.40b}$$

where

$$\|A\| = \max_i \sum_{j=1}^{n} |a_{ij}|. \tag{2.40c}$$

Additionally, we shall make use of the following lemma.

LEMMA 2.1. *Let $f_{ij}(x_j)$ be a CD function in $X_j^{(0)}$. Furthermore, let B_{ij} be determined as in Procedure 1. Then for a narrow enough interval $X_j \subset X_j^{(0)}$ (neglecting terms of order higher than 2)*

$$w(B_{ij}) = \beta_{ij} w^2(X_j). \tag{2.41}$$

Proof. For simplicity of notation the subscripts will be temporarily dropped. If X is narrow enough, equation (2.7) will have a unique solution x' in X. With no loss of generality, assume that $f(x)$ is convex in X, i.e. $f''(x) > 0$ in $X = [\underline{x}, \overline{x}]$. In this case

$$\overline{b} = f(\underline{x}) - a\underline{x}, \quad \underline{b} = f(x') - ax'.$$

But $f(\underline{x})$ can be expressed approximately as

$$f(\underline{x}) = f(x') + f'(x')(\underline{x} - x') + \frac{1}{2} f''(x')(\underline{x} - x')^2.$$

Taking into account that $f'(x') = a$ we have

$$w(B) = \overline{b} - \underline{b} = \frac{1}{2} f''(x')(\underline{x} - x')^2.$$

We can relate $x' - \underline{x}$ to $w(X)$ as follows

$$x' - \underline{x} = \alpha(\overline{x} - \underline{x}) = \alpha w(X).$$

Hence

$$w(B) = \frac{1}{2} f''(x') \alpha w^2(X) = \beta w^2(X)$$

which completes the proof. □

Now we are in a position to state the following theorem.

THEOREM 2.4. *Suppose that*

$$K(X^{(k)}) \subset X^{(k)} \tag{2.42}$$

holds for all $k \geq k_0$, i.e. that the iterative procedure (2.26) converges to a solution x^* and the Jacobian $J(x)$ is non-singular in $X^{(k)}$. Then the convergence rate towards x^* is quadratic.

Proof. It follows from (2.21) and (2.40a) that

$$w(Y_i) = \sum_{j=1}^{n} |c_{ij}| w(B_j). \tag{2.43}$$

But, using (2.12) and (2.41)

$$w(B_j) = \sum_{l=1}^{n} \beta_{jl} w^2(X_l). \tag{2.44}$$

Hence

$$w(Y_i) = \sum_{l=1}^{n} R_{il} w^2(X_l) \tag{2.45a}$$

with

$$R_{il} = \sum_{j=1}^{n} |c_{ij}| \beta_{jl}. \tag{2.45b}$$

Now it will be shown that

$$w(Y) \leq \|R\| w^2(X) \tag{2.46a}$$

where

$$R = \{R_{ij}\}. \tag{2.46b}$$

Indeed

$$w(Y) = \max_i w(Y_i) = \max_i \sum_{l=1}^{n} R_{il} w^2(X_l) \leq \max_i \sum_{l=1}^{n} |R_{il}| w^2(X_l). \tag{2.47}$$

Now (2.46a) follows from (2.47), (2.39b), and (2.40c).

Due to the invertibility of $J(x)$, matrix C has bounded elements c_{ij}. As f is a CD function, $f''(x) < \infty$ in $X^{(k)}$ and the coefficients β_{jl} are also bounded. Hence (as seen from (2.45b)) $\|R\| \leq \gamma$ in $X^{(k)}$. Finally,

$$w(Y) \leq \gamma w^2(X) \tag{2.48}$$

which completes the proof of the theorem. □

Now we shall consider the question of uniqueness.

THEOREM 2.5. *If at some iteration k_0 of the iterative procedure (2.26) the condition*

$$K(X^{(k_0)}) \subset X^{(k_0)} \tag{2.49}$$

is satisfied and the real functions $f_{ij}(x_j)$ are all strictly monotone in $X_j^{(k_0)}$, then:

a) there is a unique solution x^* to $f(x) = 0$ in $X^{(k_0)}$;

b) the modified procedure

$$X^{(k+1)} = K(X^{(k)}) = C^{(k_0)}B^{(k)}, \quad k \geq k_0 \tag{2.50}$$

where $C^{(k_0)} = (A^{(k_0)})^{-1}$ *is a constant matrix while* $B^{(k)}$ *is computed as in Procedure 1 (using, however, the same slopes* $a_{ij}^{(k_0)}$ *at each iteration), converges to* x^*.

Proof. Let for simplicity of notation $C = C^{(k_0)}$ and $k_0 = 0$. On account of (2.49) and (2.50)

$$X^{(1)} = CB^{(0)} \subset X^{(0)} \tag{2.51a}$$

so

$$w(X^{(1)}) < w(X^{(0)}). \tag{2.51b}$$

Taking into account (2.51) and the monotonicity of $f_{ij}(x_j)$ it is easily seen that

$$B^{(1)} \subset B^{(0)}. \tag{2.52a}$$

(Indeed, it follows from (2.51a) that $X_j^{(1)} \subset X_j^{(0)}$ for each j. As is readily seen, the above inclusion and the strict monotonicity of $f_{ij}(x_j)$ in $X_j^{(0)}$ imply that $B_{ij}^{(1)} \subset B_{ij}^{(0)}$ for all i and j. These inclusions lead to (2.52a) on account of (2.12).) Hence

$$X^{(2)} = CB^{(1)} \subset X^{(1)}. \tag{2.52b}$$

Thus, we have shown that procedure (2.50) generates a nested sequence $\{X^{(k)}\}$ of decreasing width $w(X^{(k)})$. By Theorem 2.3 x^* exists and is unique because $w(X^{(k)}) \to 0$ as $k \to \infty$. This completes the proof of Theorem 2.5. □

2.3. A Numerical Example

The numerical performance of the present method has been tested on several systems of equations of the form

$$f(x) = \varphi(x) - Hx - s = 0 \tag{2.53a}$$

where

$$\varphi_i(x) = \varphi_i(x_i), \quad i = 1, \ldots, n \tag{2.53b}$$

and H is a constant matrix with n ranging from 2 to 20. In all the cases $\varphi_i(x_i)$ are continuously differentiable functions. The method has been implemented in its vector form.

To illustrate the improved numerical efficiency of the new method an example (considered in [9] and [18]) will be presented here. The system (2.53) is now given by

$$\begin{aligned} &\varphi_i(x_i) = 2.5x_i^3 - 10.5x_i^2 + 11.8x_i, \quad i = 1, \ldots, 10; \\ &H = \{h_{ij}\} \quad \text{with} \quad h_{ij} = -1; \\ &s = (-1, -2, -3, \ldots, -10). \end{aligned} \tag{2.53c}$$

Table 1.

Method	M1	M2	M3	M3A
N_i	524143	116522	146	127

The initial box $X^{(0)}$ is defined by

$$x_i \in [-1, 4], \quad i = 1, \ldots, n.$$

The accuracy ε_1 has been chosen to be 10^{-4}.

Two interval methods were applied in [9] to solve the GS problem associated with (2.53). The first method denoted here as M1 is a variant of a method due to Alefeld and Herzberger ([1], Ch. 22); it is based on the use of interval derivatives. The second method designated as M2 is an improved version which uses interval slopes. This paper's method denoted as M3 has also solved the problem considered and has found within the some accuracy $\varepsilon_1 = 10^{-4}$ all the 9 solutions contained in $X^{(0)}$. However, the data in Table 1, concerning the number of iterations N_i required to solve the GS problem considered reveal that the present method is vastly superior to the other two interval methods as regards computer time.

On account of its fast convergence rate the new method has also improved characteristics as regards memory volume requirements. Indeed, the maximum number of boxes n_m stored during computation reached the value of 3 for method M3 while n_m was manifold higher for M1 and M2. It should also be stressed that no clustering effect has been observed in solving the present example by M3. In contrast, among the two previous methods, even the better method M2 generated decades of clustering boxes, thus requiring much bigger memory volume.

In practice (at the early stage of various design problems) we do not always need to solve the GS problem completely; most often, it suffices to only find the number of solutions p contained in $X^{(0)}$. This simpler problem can be efficiently solved by a modification of the present method denoted M3A. The version M3A is based on Theorem 2.5: The iterative procedure (2.26) is interrupted whenever condition (2.49) is satisfied and is resumed by retrieving a new box from the list L if L is not empty. The corresponding number of iterations for the example considered is given in the last column of Table 1.

3. Solving General Form Systems

3.1. Transformation to Separable Form

In this section, the method presented in the previous section will be extended to systems of general form. More specifically, the system's components $\psi_i(x)$ are assumed to be factorable functions [20], i.e. functions that are composed of four arithmetic operations (+, −, ∗, /), unary operations (sin, exp, log, sqrt, abs, etc.) and the power operation (^).

The approach herein adopted is to transform the general form system (1.1) to the separable form (1.9), (1.10). The theoretical basis for such a transformation is a famous theorem due to Kolmogorov's work [10] published as early as 1957. However, its proof is not constructive and only recently has a simple algorithm been proposed [19] to convert factorable functions into separable functions automatically by computer. To maintain completeness, several basic facts from [19] will be briefly presented here.

Let f_L and f_R be subfunctions of ψ containing at least one variable. Consider the following three cases

$$f = f_L * f_R; \tag{3.1}$$

$$f = f_L / f_R; \tag{3.2}$$

$$f = (f_L)^{f_R}. \tag{3.3}$$

If both f_L and f_R contain only one and the same variable then f is obviously separable in all the three cases.

If f_L contains only one variable and f_R contains only another variable then the functions (3.1) to (3.3) can be easily transformed into separable form as follows. The transformation of (3.1) is:

$$f = f_L * f_R \rightarrow \begin{cases} f = \left((y_1)^2 - (f_L)^2 - (f_R)^2\right) / 2; \\ y_1 = f_L + f_R. \end{cases} \tag{3.4}$$

The second case is reduced to the first by letting

$$f_R := 1 / f_R$$

and applying (3.4).

For the third case the transformation suggested in [19] is:

$$f = (f_L)^{f_R} \rightarrow \begin{cases} f = \exp(y_1); \\ y_1 = f_R * \log(f_L); \\ y_1 = \left((y_2)^2 - (f_R)^2 - \left(\log(f_L)\right)^2\right) / 2; \\ y_2 = f_R + \log(f_L). \end{cases} \tag{3.5}$$

It should be mentioned that (3.5) is valid only if $f_L > 0$ for all values of its argument.

If both f_L and f_R contain more than one variable then we first introduce auxiliary variables and apply the above approach. To illustrate this possibility consider formula (3.1). In this case

$$f = f_L * f_R \rightarrow \begin{cases} f = y_1 * y_2; \\ f = \left((y)^2 - (y_1)^2 - (y_2)^2\right) / 2; \\ y_1 = f_L; \\ y_2 = f_R; \\ y_3 = y_1 + y_2. \end{cases} \tag{3.6}$$

Now, by representing f_L and f_R in separable form, f can be put into separable form.

In order to make f_L and f_R separable, we perform the above transformation to f_L and f_R, regarding them as f. This process has been implemented as a computer program in [19].

To illustrate the above approach we shall consider an example.

3.2. An Illustrative Example

We take up the system considered in [17]

$$\begin{aligned} &\alpha_k(x) = 0, \quad k = 1, \ldots, 4; \\ &\beta_k(x) = 0, \quad k = 1, \ldots, 4; \\ &\gamma(x) = 0 \end{aligned} \tag{3.7a}$$

where $x \in R^9$

$$\begin{aligned} \alpha_k(x) &= (1 - x_1x_2)x_3\{e^{[x_5(g_{1k} - g_{3k}x_7 10^{-3} - g_{5k}x_8 10^{-3})]} - 1\} \\ &\quad - g_{5k} + g_{4k}x_2, \quad k = 1, \ldots, 4; \\ \beta_k(x) &= (1 - x_1x_2)x_4\{e^{[x_6(g_{1k} - g_{2k} - g_{3k}x_7 10^{-3} + g_{4k}x_9 10^{-3})]} - 1\} \\ &\quad - g_{5k}x_1 + g_{4k}, \quad k = 1, \ldots, 4; \\ \gamma(x) &= x_1x_3 - x_2x_4. \end{aligned} \tag{3.7b}$$

The numerical constants g_{ij} are given in [17]. As in [17], the problem is to establish computationally that (3.7) has a unique solution in a given initial box $X^{(0)}$ with sides

$$X_i^{(0)} = [\underline{x}_i^0, \bar{x}_i^0], \quad i = 1, \ldots, n_0 \tag{3.7c}$$

where $n_0 = 9$.

To reduce (3.7) to separable form we first introduce the auxiliary variables

$$\begin{aligned} &x_{9+k} = e^{x_5(g_{1k} - g_{3k}x_7 10^{-3} - g_{5k}x_8 10^{-3})}, \\ &x_{13+k} = e^{x_6(g_{1k} - g_{2k} - g_{3k}x_7 10^{-3} + g_{4k}x_9 10^{-3})}, \\ &k = 1, \ldots, 4; \end{aligned} \tag{3.8a}$$

$$x_{18} = x_1x_2, \quad x_{19} = (1 - x_{18})x_3, \quad x_{20} = (1 - x_{18})x_4. \tag{3.8b}$$

Using (3.7) and (3.8) we obtain the following system of 20 equations:

$$
\begin{aligned}
&x_{19}(x_{9+k} - 1) - g_{5k} + g_{4k}x_2 = 0, \quad k = 1, \ldots, 4;\\
&\ln x_{9+k} - x_5(g_{1k} - 10^{-3}g_{3k}x_7 - 10^{-3}g_{5k}x_8) = 0, \quad k = 1, \ldots, 4;\\
&x_{20}(x_{13+k} - 1) + g_{4k} - g_{5k}x_1 = 0, \quad k = 1, \ldots, 4;\\
&\ln x_{13+k} + x_6(-g_{1k} + g_{2k} + 10^{-3}g_{3k}x_7 - 10^{-3}g_{4k}x_9) = 0, \quad k = 1, \ldots, 4;\\
&x_1x_3 - x_2x_4 = 0;\\
&x_1x_2 - x_{18} = 0;\\
&(1 - x_{18})x_3 - x_{19} = 0;\\
&(1 - x_{18})x_4 - x_{20} = 0.
\end{aligned}
\tag{3.9}
$$

Since equations (3.9) contain the products x_1x_3, x_2x_4, $x_{19}(x_{9+k} - 1)$, $x_6(g_{1k} - g_{3k} \cdot 10^{-3}x_7 - g_{5k} \cdot 10^{-3}x_9)$, etc., this form will be called semiseparable.

One way to get a system of separable form is to eliminate the products in (3.9), using (3.4) and (3.6). Thus, a final system of separable form consisting of a total of 43 equations will be obtained.

An alternative approach to reach separability is suggested here. It is based on the following idea. Consider the product

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

where X and Y are intervals. If x_0 and y_0 are the centres of X and Y, respectively, then

$$xy = (x_0 + u)(y_0 + v) = x_0y_0 + y_0u + x_0v + uv = -x_0y_0 + y_0x + x_0y + uv. \tag{3.10}$$

When $x \in X$ and $y \in Y$, the centred variables $u \in R_x$ and $v \in R_y$ where R_x, R_y are the radii of X and Y. Let $R = R_xR_y$; it follows from (3.10) that

$$xy \in -x_0y_0 + y_0x + x_0y + [-R, R], \quad x \in X, \quad y \in Y. \tag{3.11}$$

Thus, the product xy has been enclosed by an interval expression in separable form, i.e.

$$xy \in \alpha x + \beta y + B_{xy} \tag{3.12}$$

where $B_{xy} = -x_0y_0 + [-R, R]$ is an interval.

The above approach is readily extended to products of the form

$$x\Big(\alpha_0 + \sum_j \alpha_j x_j\Big).$$

An appealing feature of the alternative approach is the fact that it converts the semiseparable form into separable form without introducing new variables and equations. Indeed, using this new approach the original system (3.7) has been transformed into a separable form system (2.17) of only 20 equations (rather than to 43 equations if the standard approach of [19] were applied).

Two algorithms have been elaborated to solve the GS problem associated with (3.7). They are based on the version M3A of the present method (vector form of the

method using the uniqueness test from Section 2.2). To present these algorithms we need to distinguish between the original variables $x_{0,i}$, $1 \leq i \leq 9$ and the auxiliary variables $x_{a,i}$, $10 \leq i \leq 20$. So we introduce the real vectors

$$x_0 = (x_{0,1}, \ldots, x_{0,9});$$
$$x_a = (x_{a,10}, \ldots, x_{a,20});$$
$$x = (x_0, x_a)$$

and the interval vectors X_0, X_a, and X. Now (3.8) can be written in the form

$$x_a = f_a(x_0). \tag{3.13}$$

Hence

$$X_a = f_a(X_0). \tag{3.14}$$

where $f_a(X_0)$ denotes the range of X_0 under f_a. Since each original variable occurs only once in (3.8), $f_a(X_0)$ can be computed by a single interval computation according to a well-known theorem from interval analysis.

ALGORITHM A1.

Step 0 *(Initialization).* Using the initial box $X_0^{(0)}$ given by (3.7c) and relation (3.14), the corresponding initial auxiliary vector $X_a^{(0)}$ is computed. Thus, the initial vector

$$X^{(0)} = (X_0^{(0)}, X_a^{(0)})$$

is formed.

Step 1. The version M3A is applied to the box $X^{(0)}$.

The second algorithm is a modification of A1; therefore, only the relevant differences will be noted.

ALGORITHM A2.

The modifications associated with this algorithm are, essentially, related to the case where the current box X is reduced in size.

At the current i-th iteration, X is transformed by Procedure 1 to yield vector Y. Now a new vector X^1 is obtained

$$X^1 = Y \cap X.$$

It is partitioned into two parts: X_0^1 and X_a^1 corresponding to the original and auxiliary variables. Using X_0^1 and (3.14) we obtain

$$\tilde{X}_a^1 = f_a(X_0^1).$$

Next we find the intersection

$$Z_a^1 = \tilde{X}_a^1 \cap X_a^1$$

Table 2.

Algorithm	A1	A2
N_i	183	89

to form the vector

$$X^1 = (X_0^1, Z_n^1)$$

corresponding to the new iteration $i + 1$. Finally, X^1 is renamed X and the iteration process continues as in A1.

It is also to be noted that unlike A1 the width w of the current box X^1 (needed to assess the accuracy by comparison with ε_1) is determined on the basis of the original vector X_0^1. Similarly, only X_0^1 is used in assessing the reduction of the boxes.

The system (3.7) is known to have a solution which is approximately

$$x^s \approx (0.9, 0.45, 1, 2, 8, 8, 5, 12).$$

The initial box $X_0^{(0)}$ for the original variables was chosen to be centred at x^S, that is, each component $X_{i,0}^{(0)}$ was expressed in the form

$$X_{i,0}^{(0)} = x_i^s + [-r, r]. \tag{3.15}$$

Table 2 provides data about the number of iterations needed by A1 and A2 to establish uniqueness of the solution x^s in $X^{(0)}$ when $r = 0.05$.

It is worthwhile mentioning that the sophisticated version of the interval Newton method from [17] (using elaborate subdivision strategies) establishes the existence and uniqueness of the solution to (3.7) in the box (keeping for simplicity only 5 decimal places)

$$X_I = \begin{Bmatrix} 0.89902 & 0.90098 \\ 0.44902 & 0.45098 \\ 0.99902 & 1.00098 \\ 1.99902 & 2.00098 \\ 7.99902 & 8.00098 \\ 7.99902 & 8.00098 \\ 4.99902 & 5.00098 \\ 0.99902 & 1.00098 \\ 1.99902 & 2.00098 \end{Bmatrix}.$$

Using X_I, we can compute its width $w' = 0.00196$, or approximately 0.002. It is seen that with $r = 0.05$ the width $w = 0.1$ of the box $X^{(0)}$ is approximately 500 times larger in comparison to that of the box X_I.

4. Conclusion

In this paper, the problem of finding (within preset accuracy) the set of all real solutions to a system of nonlinear equations (1.1) contained in a given box $X^{(0)}$

(the GS problem) has been considered. The only assumption about the nonlinear functions involved is that they are continuous in $X^{(0)}$.

A new method for solving the GS problem has been suggested. It is based on a transformation of the original system (1.1) into a new system (1.9) of separable form (1.10). The latter system is then solved globally by a new interval method which exploits the separability property (1.10). More specifically, each function $f_{ij}(x_j)$ from (1.10) is approximated dynamically by a linear interval function $L_{ij}(X_j)$ defined by (2.1). Thus, at each iteration of the method a linear interval system (1.11) with a real matrix A is solved. This advantageously distinguishes the present method from the other known interval methods where a much more complex linear interval system having an interval matrix and a real right-hand side vector is to be solved.

Experimental data show that as regards computer time and memory volume requirements the present method exceeds considerably the other known methods for solving the global solution problem considered. It should also be mentioned that so far no clustering has been observed.

There seem to exist several possibilities for further improvement of the numerical efficiency of the new method. Since the equivalent separable form system (1.9) is, generally, much larger than the original system but has, at the same time, a rather sparse structure, one approach should appeal to sparse matrix techniques in implementing the componentwise version of the method. Work is presently in progress to accomplish this scheme. Using such an approach it would be interesting to compare the relative efficiency of the two possible conversions to separability: the semiseparable form suggested and employed in this paper or the standard separability form from [19]. Another possibility is to incorporate into the present method ideas from the constraint propagation method. Preliminary limited experimental results indicate that such an approach seems to be rather promising. Finally, it remains to investigate the convergence properties of the alternative iterative procedure (2.20) which is based on linear programming implementation; it may turn out that such an approach could lead to a method of improved overall numerical efficiency.

Acknowledgements

This work was supported by the Bulgarian Research Foundation under Contract No TN-554/95.

The author is grateful to one of the referees for his valuable questions and remarks.

References

1. Alefeld, G. and Herzberger, J.: *Introduction to Interval Computations*, Academic Press, New York, 1983.
2. Hansen, E. and Greenberg, R.: An Interval Newton Method, *Appled Math. and Comp.* **12** (1983), pp. 89–98.

3. Kearfott, R., Hu, C., and Novoa, M.: A Review of Preconditioners for the Interval Gauss-Seidel Method, *Interval Computations* 1 (1991), pp. 59-85.
4. Kolev, L.: Finding All Solutions of Non-Linear Resistive Circuit Equations via Interval Analysis, *Int. J. Cir. Theor. Appl.* **12** (1984), pp. 75–178.
5. Kolev, L.: *Interval Methods for Circuit Analysis*, World Scientific, Singapore, New Jersey, 1993.
6. Kolev, L.: Use of Interval Slopes for the Irrational Part of Factorable Functions, *Reliable Computing* **3** (1) (1997), pp. 83–93.
7. Kolev, L. and Mladenov, V.: An Interval Method for Finding All Operating Points of Non-Linear Resistive Circuits, *Int. J. Cir. Theor. Appl.* **18** (1990), pp. 257–267.
8. Kolev, L. and Mladenov, V.: An Interval Method for Global Non-Linear DC Circuits Analysis, *Int. J. Cir. Theor. Appl.* **22** (1994), pp. 233–241.
9. Kolev, L. and Mladenov, V.: Use of Interval Slopes in Implementing an Interval Method for Global Non-Linear DC Circuits Analysis, *Int. J. Cir. Theor. Appl.* **25** (1997), pp. 37–42.
10. Kolmogorov, A.: On the Representation of Continuous Functions of Many Variables by Superposition of Continuous Functions of One Variable and Addition, *Dokl. Akad. Nauk SSSR*, **114** (1957), pp. 953–956 (in Russian; English transl.: *American Mathematical Biophysics* **5** (1963), pp. 55–59).
11. Krawczyk, R. and Neumaier, A.: Interval Slopes for Rational Functions and Associated Centred Forms, *SIAM J. Numer. Anal.* **22** (1985), pp. 604–616.
12. Mladenov, V. and Kolev, L.: Interval Methods for Solving Cellular Neural Networks (CNNs) Equations, in: *The Third Int. Conf. on Electronics, Circuits and Systems (ICECS'96)*, Rodos, Greece, 13–16 October, 1996.
13. Moore, R.: A Test for Existence of Solutions to Nonlinear Systems, *SIAM J. Numer. Anal.* **14** (4) (1977), pp. 611–615.
14. Moore, R. and Qi, L.: A Successive Interval Test for Nonlinear Systems, *SIAM J. Numer. Anal.* **19** (4) (1982), pp. 845–850.
15. Pandian, M.: A Convergence Test and Componentwise Error Estimates for Newton Type Methods, *SIAM J. Numer. Anal.* **22** (4) (1985), pp. 779–791.
16. Qi, L.: A Note on the Moore Test for Nonlinear Systems, *SIAM J. Numer. Anal.* **19** (4) (1982), pp. 851–857.
17. Ratschek, H. and Rokne, J.: Experiment Using Interval Analysis for Solving a Circuit Design Problem, *J. of Global Optimization* **3** (1993), pp. 501–518.
18. Yamamura K.: Finding All Solutions of Piecewise-Linear Resistive Circuits Using Simple Sign Tests, *IEEE Trans. Circuits Syst.* **1, 40** (1993), pp. 546–551.
19. Yamamura K.: An Algorithm for Representing Functions of Many Variables by Superpositions of Functions of One Variable and Addition, *IEEE Trans. Circuits Syst.* **1, 43** (3) (1996), pp. 338–340.
20. Zuhe, S. and Wolfe, M.: On Interval Enclosures Using Slope Arithmetic, *Appl. Math. Comput.* **39** (1990), pp. 89–105.