Title	Interval Analysis Using Linear Programming
Author(s)	YAMAMURA, Kiyotaka
Citation	数理解析研究所講究録 (1997), 990: 235-243
Issue Date	1997-04
URL	http://hdl.handle.net/2433/61082
Right	
Type	Departmental Bulletin Paper
Textversion	publisher

数理解析研究所講究録
990 巻 1997 年 235-243

線形計画法を用いた区間解析

山 村 清 隆

群馬大学工学部情報工学科
群馬県桐生市天神町 1–5–1

あらまし　区間解析は非線形方程式のすべての解を求める代表的な方法として知られているが，問題の次元の増加とともに計算時間が指数関数的に増大する欠点をもつ．区間解析の計算効率を改善するためには，与えられた領域に解が存在しないことを判定する強力なテストを開発する必要がある．本論文では，区間解析に線形計画法を導入することにより，線形項の多い非線形方程式に対しては，そのすべての解を非常に効率よく求められることを示す．本手法の基本的な考えは次の通りである．まず与えられた領域に対し，区間拡張を用いて非線形関数を長方形（あるいは多次元の直方体）で囲み，その領域内のすべての解を含むような実行可能領域をもつ線形計画問題を定式化する．そのような実行可能領域の存在・非存在を単体法の Phase I で確認することにより，非線形方程式の解の非存在を判定することができる．さらに Phase II を利用して実行可能領域を含む最小の直方体を求めることにより，同じ解を含むより小さな領域を得ることができる．このテストは従来の区間解析で用いられているテストよりも遥かに強力で，区間解析の計算効率を飛躍的に改善することができる．

Interval Analysis Using Linear Programming

Kiyotaka YAMAMURA

Department of Computer Science, Gunma University, Kiryu 376, Japan
E-mail: yamamura@msc.cs.gunma-u.ac.jp Tel: 0277–30–1812 Fax: 0277–30–1814

Abstract　A new computational test is proposed for nonexistence of a solution to a system of nonlinear equations in a convex polyhedral region X. The basic idea proposed here is to formulate a linear programming problem whose feasible region contains all solutions in X. Therefore, if the feasible region is empty (which can be easily checked by Phase I of the simplex method), then the system of nonlinear equations has no solution in X. The linear programming problem is formulated by surrounding the component nonlinear functions by rectangles using interval extensions. This test is much more powerful than the conventional test if the system of nonlinear equations consists of many linear terms and relatively a small number of nonlinear terms. By introducing the proposed test to interval analysis, all solutions of nonlinear equations can be found very efficiently.

1. Introduction

This paper deals with the problem of finding all solutions of a system of nonlinear equations:

$$\begin{array}{rcl} f_1(x_1,x_2,\cdots,x_n) & = & 0 \\ f_2(x_1,x_2,\cdots,x_n) & = & 0 \\ \vdots & & \\ f_n(x_1,x_2,\cdots,x_n) & = & 0 \end{array} \qquad (1)$$

contained in a bounded rectangular region D in R^n, where $f_1, f_2, \cdots, f_n$ are real-valued nonlinear functions. In vector notation the system (1) will be written as $f(x) = 0$.

As a computational method to find all solutions of nonlinear equations, interval analysis is well-known, and various algorithms based on interval computation have been developed [1],[2],[5]–[19],[23]–[28],[30]. Using the interval analysis, all solutions of (1) contained in $D \subset R^n$ can be found with mathematical certainty. However, the computation time of the interval analysis tends to grow exponentially with the dimension n. Even for small problems, the interval analysis often requires enormous computation time if the nonlinearity of the problems is very large or the problems are ill-conditioned.

In order to improve the computational efficiency of the interval analysis, it is necessary to develop a powerful method for excluding interval vectors (n-dimensional rectangules) containing no solution very effectively. In this paper, we propose a new computational test for nonexistence of a solution to the system of nonlinear equations (1) in a region X. The basic idea proposed here is to formulate a linear programming problem whose feasible region contains all solutions in X. Hence, if the feasible region is empty (which can be easily checked by Phase I of the simplex method), then X contains no solution, and we can exclude it from further consideration. The linear programming problem is formulated by surrounding the component nonlinear functions by rectangles or suitable convex polygons. This test is much more powerful than the conventional test if the system of nonlinear equations consists of many linear terms and relatively a small number of nonlinear terms. By numerical examples, it is shown that all solutions can be found very efficiently by using the proposed test.

Most of the theoretical results of this paper have been presented in [36]–[38]. This work is an extension of the ideas in [21] and [40] to finding all solutions of nonlinear equations.

2. Interval Analysis

In this section, we summarize the basic procedure of the interval analysis briefly.

Intervals will be denoted by capital letters. An n-dimensional interval vector with components $X_i = [a_i, b_i]$ $(i = 1, 2, \cdots, n)$ is denoted by

$$X = (X_1, X_2, \cdots, X_n)^T. \qquad (2)$$

Geometrically, X is an n-dimensional rectangle.

In the interval analysis, the following procedure is performed recursively beginning with the initial region $X = D$. At each level, we analyze the region X. If there is no solution of (1) in X, then we exclude it from further consideration. If there is a unique solution of (1) in X, then we compute it by some iterative method. In the field of interval analysis, computationally verifiable sufficient conditions for nonexistence, existence and uniqueness of a solution in X have been developed. If these conditions are not satisfied and the existence or nonexistence of a solution in X cannot be checked, then bisect X in some appropriately chosen coordinate direction to form two new rectangles; we then continue the above procedure with one of these rectangles, and put the other one on a stack for later consideration.

The nonexistence of a solution in X can be checked by using interval extensions. If in $f(x_1, x_2, \cdots, x_n)$ the variables x_i are replaced by intervals X_i and the arithmetic operations are replaced by the corresponding interval operations (for example, $x_1 + x_2$ is replaced by $X_1 + X_2 = [a_1 + a_2, b_1 + b_2]$), then an interval-valued vector function $F(X)$ is obtained which is called the interval extension of $f(x)$. It is known that $F(X)$ contains the range of $f(x)$ over X. Hence, if

$$0 \notin F(X) \qquad (3)$$

holds, then there is no solution of (1) in X. This is the computationally verifiable sufficient condition for nonexistence of a solution to (1) in X, which is used as the test for nonexistence in the conventional interval analysis †.

Geometrically, (3) implies that there exists an i such that the $(n-1)$-dimensional solution surface satisfying $f_i(x) = 0$ does not exist in X. In other words, if all solution surfaces of $f_i(x) = 0$ $(i = 1, 2, \cdots, n)$ exist in X, then $0 \in F(X)$ holds. However, although all solution surfaces exist in X, a solution does not exist unless they intersect at the same point. Hence, $0 \in F(X)$ is merely a necessary condition for existence of solutions in X. In practical applications, $0 \in F(X)$ often holds although there is no solution in X, especially when the rectangle X is large or $f(x)$ is ill-conditioned. Thus, (3) is not necessarily a powerful test for excluding

†As another computationally verifiable sufficient condition for nonexistence of a solution to (1) in X, $K(X) \cap X = \phi$ is known where $K(X)$ is the Krawczyk operator.

regions. In order to improve the computational efficiency of the interval analysis, it is very important to develop a powerful test which checks the nonexistence of a solution in X.

3. A New Test Using Linear Programming

3.1 Basic Idea

In practical problems, the system of nonlinear equations often consists of many linear terms and relatively a small number of nonlinear terms. The test proposed in this paper is suited to such systems. In this section, we first consider the case where (1) can be represented as

$$\sum_{j\in J_i} g_{ij}(x_j) + \sum_{j=1}^{n} h_{ij}x_j - s_i = 0, \quad i = 1, 2, \cdots, n \tag{4}$$

where $g_{ij}(x_j)$ is a nonlinear function of one variable, h_{ij} and s_i $(i, j = 1, 2, \cdots, n)$ are constants, and J_i is a subset of $\{1, 2, \cdots, n\}$. Assume that $\sum_{i=1}^{n} |J_i|$ is not a so large number, where $|J_i|$ denotes the cardinality of the set J_i. The case where nonseparable functions of more than one variables are contained will be considered later. As a typical example of the system of nonlinear equations of the form (4), nonlinear circuit equations in hybrid representation [4], [35] is known, where $J_i = \{i\}$ holds for all i.

Let the range of $g_{ij}(x_j)$ $(i = 1, 2, \cdots, n,\ j \in J_i)$ over $[a_j, b_j]$ be $[c_{ij}, d_{ij}]$. Here, the range may be the exact range (if possible) or the interval extension containing the exact range. Then, we introduce auxiliary variables y_{ij} $(i = 1, 2, \cdots, n,\ j \in J_i)$ and put $y_{ij} = g_{ij}(x_j)$. If $a_j \leqq x_j \leqq b_j$, then $c_{ij} \leqq y_{ij} \leqq d_{ij}$.

Now we consider the following linear programming (LP) problem:

$$\begin{aligned} &\max \ \text{(arbitrary function)} \\ &\text{subject to} \\ &\sum_{j\in J_i} y_{ij} + \sum_{j=1}^{n} h_{ij}x_j - s_i = 0, && i = 1, 2, \cdots, n \\ &a_i \leqq x_i \leqq b_i, && i = 1, 2, \cdots, n \\ &c_{ij} \leqq y_{ij} \leqq d_{ij}, && i = 1, 2, \cdots, n, \\ & && j \in J_i. \end{aligned} \tag{5}$$

Geometrically, the inequality constraints in (5) implies that the component nonlinear functions $g_{ij}(x_j)$ are surrounded by rectangles as shown in Fig. 1.

Evidently, all solutions of (4) which exist in $X = ([a_1, b_1], \cdots, [a_n, b_n])^T$ satisfy the constraints in (5) if we put $y_{ij} = g_{ij}(x_j)$. Namely, the feasible region of the LP problem (5) is a convex polyhedron containing all solutions of (4) in X. Hence, if the feasible region is empty, then we can conclude that there is no solution of (4) in X.

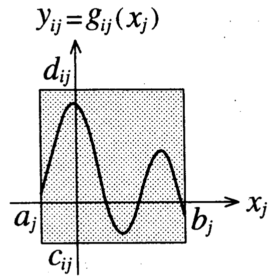

Fig. 1 In the proposed test, nonlinear functions are surrounded by rectangles.

The emptiness or nonemptiness of the feasible region of (5) can be checked by Phase I of the simplex method †. This is the new computational test for nonexistence of a solution to (4) in X. This test is very simple because we just apply Phase I of the simplex method to (5). Since there are many good softwares of the simplex method, the implementation of the proposed test is very easy. For simplicity, we will refer to the proposed test as the LP test.

Note that if the feasible region is empty, then any system of nonlinear equations of the form (4) where the range of $g_{ij}(x_j)$ $(i = 1, 2, \cdots, n,\ j \in J_i)$ over $[a_j, b_j]$ is contained in $[c_{ij}, d_{ij}]$ does not have a solution in X.

As shown in the following example, the LP test is more powerful (often much more powerful) than the conventional test (3). This is because the structure of (4) is fully exploited in the new test.

Example: Consider a system of nonlinear equations:

$$f_1(x_1, x_2) = x_1 - x_2 = 0$$
$$f_2(x_1, x_2) = (x_1 - 1)^2 - x_2 = 0.$$

Let $X = ([-10, 0], [-10, 10])^T$. Using the exact range, we obtain $F(X) = ([-20, 10], [-9, 131])^T$. Since $0 \in F(X)$, the conventional test (3) cannot exclude this region. However, since the feasible region satisfying

†This idea is an extension of the ideas in [21] and [40] to finding all solutions of nonlinear equations. In [21], the concept of polyhedral circuit (i.e., circuit with resistive elements whose characteristics are polyhedra) is introduced and the emptiness or nonemptiness of the solution domain of the polyhedral circuit is checked by Phase I of the simplex method. In [40], an LP problem similar to (5) is formulated for finding all solutions of piecewise-linear circuits.

$$x_1 - x_2 = 0$$
$$y_1 - x_2 = 0$$
$$-10 \leqq x_1 \leqq 0$$
$$-10 \leqq x_2 \leqq 10$$
$$1 \leqq y_1 \leqq 121$$

is empty, the LP test can exclude this region. □

Now let us examine the size of the tableau in the LP test. In the implementation of the simplex method to (5), we apply the variable transformation $\bar{x}_i = x_i - a_i$ and $\bar{y}_{ij} = y_{ij} - c_{ij}$ so that the LP problem becomes the form with nonnegativity constraints:

max (arbitrary function)
subject to

$$\begin{aligned} &\sum_{j \in J_i} \bar{y}_{ij} + \sum_{j=1}^{n} h_{ij}\bar{x}_j - \bar{s}_i = 0, && i = 1, 2, \cdots, n \\ &\bar{x}_i \leqq b_i - a_i, && i = 1, 2, \cdots, n \\ &\bar{y}_{ij} \leqq d_{ij} - c_{ij}, && i = 1, 2, \cdots, n,\ j \in J_i \\ &\bar{x}_i \geqq 0,\ \ \bar{y}_{ij} \geqq 0, && i = 1, 2, \cdots, n,\ j \in J_i. \end{aligned} \tag{6}$$

This LP problem has n equality constraints and $\sum_{i=1}^{n}(l_i + 1)$ inequality constraints (excluding the nonnegativity constraints) where $l_i = |J_i|$. Introducing the slack variables λ_i and μ_{ij} $(i = 1, 2, \cdots, n,\ j \in J_i)$, (6) is transformed into a standard form:

max (arbitrary function)
subject to

$$\begin{aligned} &\sum_{j \in J_i} \bar{y}_{ij} + \sum_{j=1}^{n} h_{ij}\bar{x}_j - \bar{s}_i = 0, && i = 1, 2, \cdots, n \\ &\bar{x}_i + \lambda_i = b_i - a_i, && i = 1, 2, \cdots, n \\ &\bar{y}_{ij} + \mu_{ij} = d_{ij} - c_{ij}, && i = 1, 2, \cdots, n,\ j \in J_i \\ &\bar{x}_i \geqq 0,\ \ \bar{y}_{ij} \geqq 0,\ \ \lambda_i \geqq 0,\ \ \mu_{ij} \geqq 0, & \\ & && i = 1, 2, \cdots, n,\ j \in J_i. \end{aligned}$$

In Phase I, we introduce artificial variables to obtain an initial basic feasible solution. Since $b_i - a_i > 0$ and $d_{ij} - c_{ij} > 0$ $(i = 1, 2, \cdots, n,\ j \in J_i)$ hold, we only need to introduce n artificial variables for the first n equality constraints. Hence, the size of the tableau is $\{\sum_{i=1}^{n}(l_i + 2) + 1\} \times \{\sum_{i=1}^{n}(l_i + 1) + 1\}$.

Remark 1: The LP test can be applied to the case where X is a convex polyhedron other than a rectangle. In that case, we replace $a_i \leqq x_i \leqq b_i$ $(i = 1, 2, \cdots, n)$ in (5) by the inequalities representing the region X. □

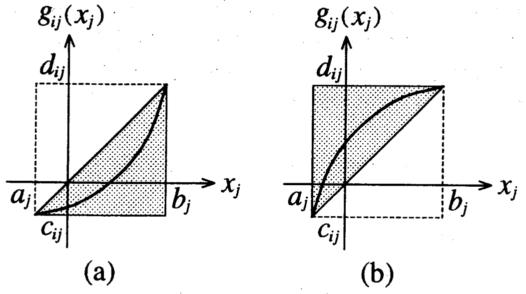

Fig. 2 Nonlinear functions surrounded by right-angled triangles.

Remark 2: In the LP test, the component nonlinear functions $g_{ij}(x_j)$ are surrounded by rectangles as shown in Fig. 1. However, if the component nonlinear functions have some favorable properties such as monotonicity or convexity, then we may surround them by suitable convex polygons instead of the rectangles. For example, if the functions $g_{ij}(x_j)$ are monotone and convex, then it is efficient to surround them by right-angled triangles whose two sides are parallel to the x_j and y_{ij} axes as shown in Fig. 2. In the case of Fig. 2(a), we apply the variable transformation $\bar{x}_j = b_j - x_j$ and $\bar{y}_{ij} = y_{ij} - c_{ij}$, and in the case of Fig. 2(b), we apply the variable transformation $\bar{x}_j = x_j - a_j$ and $\bar{y}_{ij} = d_{ij} - y_{ij}$. Then, we can represent the right-angled triangle by one inequality constraint:

$$\bar{y}_{ij} \leqq -\frac{d_{ij} - c_{ij}}{b_j - a_j}\bar{x}_j + (d_{ij} - c_{ij}) \tag{7}$$

and the nonnegativity constraints $\bar{x}_j \geqq 0$ and $\bar{y}_{ij} \geqq 0$. Hence, the number of inequality constraints decreases compared with (6). Moreover, since triangles are smaller than rectangles, the LP test becomes more powerful by using right-angled triangles. However, if we use triangles other than right-angled triangles, then the computation time often becomes larger because the number of inequality constraints increases. □

3.2 Application to More General Cases

The LP test can be applied to the case where the system of nonlinear equations (1) contains nonseparable functions of more than one variables. For example, assume that (1) contains a nonseparable function of two variables $g(x_1, x_2)$. In this case, we first compute the range of $g(x_1, x_2)$ over X by interval operations. Let the range be $[c, d]$. Then, we introduce an auxiliary variable $y = g(x_1, x_2)$ and formulate the LP problem similar to (5) where

$g(x_1, x_2)$ is replaced by y and the inequality constraint $c \leqq y \leqq d$ is added. That is, we surround the two-dimensional function surface of $y = g(x_1, x_2)$ by a three-dimensional rectangle. Such extension is also possible to more general cases; if (1) contains a nonseparable function of m variables, then we surround the m-dimensional function surface by an $(m+1)$-dimensional rectangle. Then, we can formulate the LP problem similar to (5).

Example: Let $X = ([1,2],[1,2])^T$ and consider a system of nonlinear equations:

$$f_1(x_1, x_2) = x_1 - x_2 = 0$$

$$f_2(x_1, x_2) = x_1 x_2 - 4x_1 + 2x_2 - 3 = 0.$$

Then, the LP problem we consider is

$$\begin{aligned} &\max \ \text{(arbitrary function)} \\ &\text{subject to} \\ &\quad x_1 - x_2 = 0 \\ &\quad y - 4x_1 + 2x_2 - 3 = 0 \\ &\quad 1 \leqq x_1 \leqq 2 \\ &\quad 1 \leqq x_2 \leqq 2 \\ &\quad 1 \leqq y \leqq 4. \end{aligned}$$

In this problem, the feasible region is empty. □

Remark 3: It is known that any nonseparable function of many variables can be represented by a set of separable functions, i.e., additions of functions of one variable [33],[34],[39]. For example, $\exp(2x_1 + x_1 x_2 + 3x_3^2)$ can be represented as

$$\begin{aligned} &\exp(2x_1 + x_1 x_2 + 3x_3^2) \\ &\quad \rightarrow \quad \exp(z_1) \\ &\qquad\quad z_1 = 2x_1 + (z_2^2 - x_1^2 - x_2^2)/2 + 3x_3^2 \\ &\qquad\quad z_2 = x_1 + x_2 \end{aligned}$$

where z_1 and z_2 are auxiliary variables. Algorithms for representing nonseparable functions by separable functions are proposed in [33],[34]. If we represent the nonseparable functions in (1) by separable functions, then we can perform the LP test using two-dimensional rectangles only. □

3.3 Finding Smaller Regions Containing the Same Solutions Using Phase II

In the LP test, the emptiness or nonemptiness of the feasible region is checked by Phase I of the simplex method. If it is not empty, then we find a basic feasible solution (an extreme point of the feasible region) at the end of the Phase I. Then, we may solve the following $2n$ LP problems:

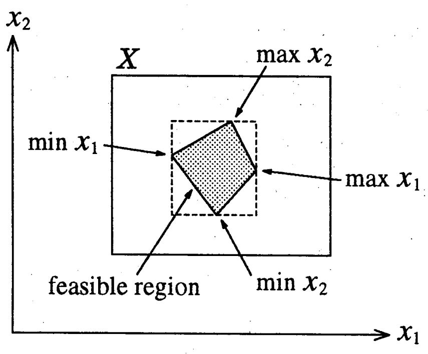

Fig. 3 The smallest rectangle containing the feasible region can be obtained by Phase II.

$$\begin{aligned} &\max / \min \ x_k \\ &\text{subject to} \\ &\sum_{j \in J_i} y_{ij} + \sum_{j=1}^{n} h_{ij} x_j - s_i = 0, && i = 1, 2, \cdots, n \\ &a_i \leqq x_i \leqq b_i, && i = 1, 2, \cdots, n \\ &c_{ij} \leqq y_{ij} \leqq d_{ij}, && i = 1, 2, \cdots, n, \\ & && j \in J_i \end{aligned}$$

for $k = 1, 2, \cdots, n$. That is, we maximize and minimize x_k for all $k = 1, 2, \cdots, n$ under the same constraints as those in (5). Since we have already obtained the basic feasible solution by Phase I, these LP problems can be solved by Phase II only. As shown in Fig. 3.3, the optimal solutions of these problems give the upper bounds and the lower bounds of the feasible region in each coordinate direction. Hence, we can obtain the smallest rectangle which contains the feasible region (the rectangle described by dashed lines in Fig. 3.3). Since all solutions in X are contained in this rectangle, we can replace X by this smaller region for improving the computational efficiency. Thus, the LP test can be used not only to check the nonexistence of a solution in X but also to find a smaller region containing the same solutions.

4. Numerical Examples

We introduced the LP test to the well-known Krawczyk-Moore algorithm [12]–[15] and implemented the new algorithm (where the LP test is performed after the test (3)) using the programming language C on a Sun Ultra SPARC model 170. In the programming, we used the program of the simplex method written in [22]. We did not introduce the technique described in Section 3.3. We have applied the algorithm to many examples and have confirmed the large effectiveness of the LP test. In this section, we show some examples.

Example 1: We first consider a system of n non-

Table 1 Result of computation (Example 1).

n	S	Conventional algorithm N	T (s)	Proposed algorithm N	T (s)
8	7	158,535	25	2,937	3
10	9	2,064,767	541	5,053	6
12	9	26,157,153	10,590	7,877	16
14	5	∞	∞	8,507	27
16	9	∞	∞	12,715	67
18	7	∞	∞	14,539	90
20	9	∞	∞	19,219	165
22	7	∞	∞	23,405	267
24	7	∞	∞	26,005	389
26	9	∞	∞	33,843	649
28	5	∞	∞	38,645	932
30	9	∞	∞	44,301	1,314
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
40	9	∞	∞	75,123	4,069
50	11	∞	∞	115,149	12,674

linear equations:

$$g(x_i) + x_1 + x_2 + \cdots + x_n - i = 0, \quad i = 1, 2, \cdots, n$$

where

$$g(x_i) = 2.5x_i^3 - 10.5x_i^2 + 11.8x_i$$

which describes a nonlinear resistive circuit containing n tunnel diodes [31],[32],[35]. We considered the initial region $([-10, 10], \cdots, [-10, 10])^T$ and applied the conventional Krawczyk-Moore algorithm and the proposed algorithm for various n. Table 1 shows the result of computation, where S denotes the number of solutions obtained by the algorithms, N denotes the number of analyzed regions, T denotes the computation time, and ∞ denotes that it could not be computed in practical computation time. As seen from the table, the proposed algorithm is much more efficient than the conventional Krawczyk-Moore algorithm (especially when n is large).

Example 2: We next consider a system of n nonlinear equations [3]:

$$x_i - \frac{1}{2n}\left(\sum_{j=1}^{n} x_j^3 + i\right) = 0, \qquad i = 1, 2, \cdots, n.$$

The initial region is $([-10, 10], \cdots, [-10, 10])^T$. We applied the conventional Krawczyk-Moore algorithm and the proposed algorithm to this problem for various n. Table 2 shows the result of computation. As seen from the table, the Krawczyk-Moore algorithm becomes much more efficient by introducing the LP test. It is also seen that the computation time of the proposed algorithm does not grow so explosively compared with that of the Krawczyk-Moore algorithm.

Example 3: We next consider a system of 10 equations:

Table 2 Result of computation (Example 2).

n	S	Conventional algorithm N	T (s)	Proposed algorithm N	T (s)
8	3	54,903	16	1,477	1
10	3	400,071	210	2,289	3
12	3	2,984,875	2,411	3,427	7
14	3	22,707,547	26,428	4,561	14
16	3	∞	∞	6,369	30
18	3	∞	∞	7,469	48
20	3	∞	∞	9,245	85
22	3	∞	∞	11,263	137
24	3	∞	∞	13,581	212
26	3	∞	∞	15,317	306
28	3	∞	∞	17,843	433
30	3	∞	∞	20,641	617

$$\sum_{j=1}^{10} x_j + x_i - (n+1) = 0, \qquad i = 1, 2, \cdots, 9$$

$$\prod_{j=1}^{10} x_j - 1 = 0$$

which is known as Brown's almost linear system [3],[8]. The initial region is $([-10, 10], \cdots, [-10, 10])^T$. There are two solutions within the region.

Wen we applied the Krawczyk-Moore algorithm, the computation time was 375,804 seconds (more than four days). However, when we applied the proposed algorithm, all solutions were found in only 131 seconds.

Example 4: We next consider a system of n nonlinear equations:

$$x_{i-1} - 2x_i + x_{i+1} + h^2 \exp(x_i) = 0, \quad i = 1, 2, \cdots, n$$

where $x_0 = x_{n+1} = 0$ and $h = 1/(n+1)$. This system comes from a nonlinear two-point boundary value problem termed the Bratu problem [17]. Since the exponential function can be surrounded by a right-angled triangle, we also considered the LP test using right-angled triangles described in Remark 3.2. We considered the initial region $([-10, 10], \cdots, [-10, 10])^T$ and applied the Krawczyk-Moore algorithm and the proposed algorithms (i.e., the LP test algorithm using rectangles and that using right-angled triangles) for various n. Table 3 shows the result of computation. It is seen that the Bratu problem could be solved much more efficiently by introducing the LP test. It is also seen that the LP test becomes more powerful by using right-angled triangles.

Example 5: Next, we consider the transistor-diode circuit shown in Fig. 4 [29]. Using the Ebers-Moll model [4], this circuit is described by a system of nonlinear equations in 15 variables of the form (4) where $J_i = \{i\}$. In this system, all nonlinear

Table 3 Result of computation (Example 4).

n	S	Conventional algorithm N	Conventional algorithm T (s)	Using rectangles N	Using rectangles T (s)	Using triangles N	Using triangles T (s)
10	2	67,377	13	3,725	4	2,281	2
12	2	317,897	95	7,049	11	3,821	5
14	2	2,009,599	842	9,709	23	4,721	9
16	2	9,695,185	5,681	21,483	83	9,179	27
18	2	47,683,439	36,957	23,129	118	10,877	44
20	2	∞	∞	33,051	238	15,313	80
22	2	∞	∞	84,493	825	21,693	158
24	2	∞	∞	62,193	791	23,787	219
26	2	∞	∞	85,447	1,457	30,979	397
28	2	∞	∞	117,683	2,466	42,545	672
30	2	∞	∞	139,529	3,709	51,983	1,016

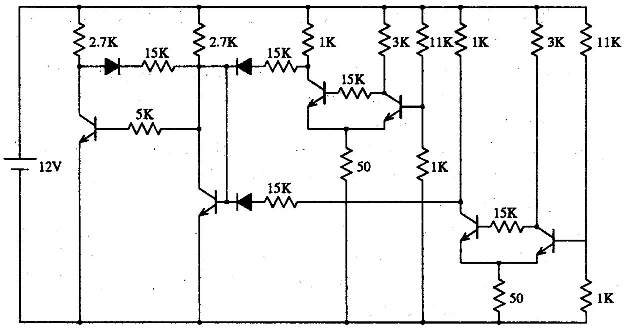

Fig. 4 Transistor-diode circuit (Example 5).

Table 4 Result of computation (Example 5).

	Regions	Pivotings	T (s)
Rectangles	36,737	697,487	131
Right-angled triangles	29,789	504,437	69
Triangles with tangents	21,503	805,991	200

functions are the exponential functions. The initial region we consider is $([-20, 0.5], \cdots, [-20, 0.5])^T$. There are 11 solutions within the region.

We solved this system by the proposed algorithm using rectangles, that using right-angled triangles, and that using triangles whose two sides are the tangents of $g_{ij}(x_j)$ at (a_j, c_{ij}) and (b_j, d_{ij}) [37]. Table 4 shows the result of computation. It is seen that the LP test using right-angled triangles is the most efficient. This is because the number of inequality constraints decreases by using right-angled triangles, and right-angled triangles are smaller than rectangles. However, if we use triangles other than right-angled triangles, the computation time often becomes larger although the number of analyzed regions decreases. This is because the number of inequality constraints increases by using triangles other than right-angled triangles.

Example 6: Finally, we consider the transistor circuit shown in Fig. 5 [32], [39], [40] and the transistor-diode circuit shown in Fig. 6. The numbers of variables are 8 and 22, respectively.

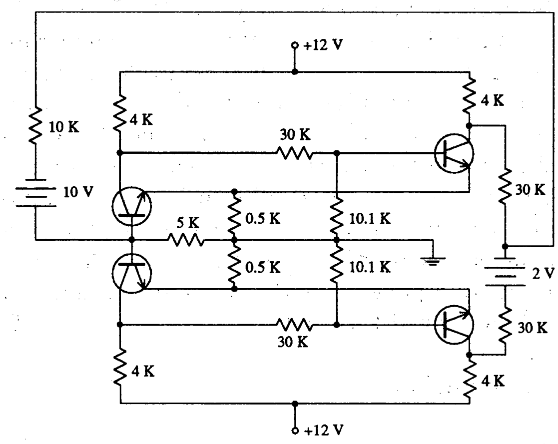

Fig. 5 Transistor circuit (Example 6).

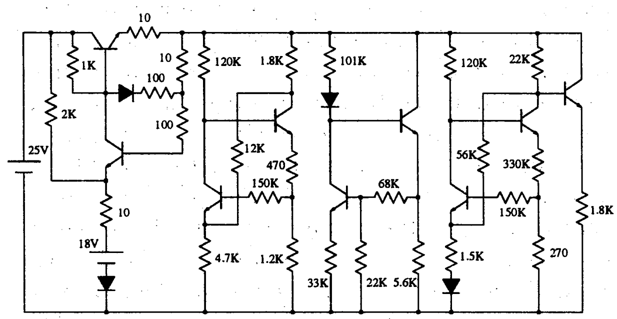

Fig. 6 Transistor-diode circuit (Example 6).

Table 5 Result of computation (Example 6, $n = 8$).

	Regions	Pivotings	T (s)
Rectangles	2,149	28,714	1.8
Right-angled triangles	1,913	21,093	1.1
Triangles with tangents	1,839	34,650	2.7

Table 6 Result of computation (Example 6, $n = 22$).

	Regions	Pivotings	T (s)
Rectangles	2,103	115,367	34
Right-angled triangles	1,971	100,209	20
Triangles with tangents	1,711	143,896	65

The initial region we consider is $([-20, 0.5], \cdots,$

$[-20, 0.5])^T$. The numbers of solutions within the region are 9 and 1, respectively.

We solved this system by the proposed algorithm using rectangles, that using right-angled triangles, and that using triangles whose two sides are the tangents of $g_{ij}(x_j)$ at (a_j, c_{ij}) and (b_j, d_{ij}). Table 5 and Table 6 show the result of computation. In both cases, the LP test using right-angled triangles is the most efficient.

Remark 4: Recently, Prof. Shin'ichi Oishi of Waseda University, Tokyo, Japan, succeeded to find all solutions of the multiphase equilibrium problem of polymer solution (which is a very important problem in polymer chemistry) by the Krawczyk-Moore algorithm using the LP test [20]. This problem is known to be a very difficult problem whose all solutions had not been found with mathematical certainty. In fact, this problem could not be solved by the conventional Krawczyk-Moore algorithm †. In [20], it is reported that all solutions of the multiphase equilibrium problem could be found with guaranteed accuracy by using the algorithm proposed in this paper and the numerical validation system developed there [7]. □

5. Conclusion

In this paper, a new computational test has been proposed for nonexistence of a solution to a system of nonlinear equations in a convex polyhedral region X. The basic idea proposed here is to formulate a linear programming problem whose feasible region contains all solutions in X (such a problem can be formulated by surrounding the component nonlinear functions by rectangles) and then check the emptiness or nonemptiness of the feasible region by the simplex method. The proposed test is very powerful if the the system of nonlinear equations consists of many linear terms and relatively a small number of nonlinear terms. Moreover, it can be easily implemented because there are many publicly available softwares of the simplex method. Using the proposed test, we can find all solutions of nonlinear equations very efficiently.

Acknowledgement

The authors would like to thank Prof. Kazuo Horiuchi and Prof. Shin'ichi Oishi of Waseda University for helpful discussions. They are also grateful to Prof. Götz E. Alefeld of Universität Karlsruhe for kind advices. This work was supported in part by the Japanese Ministry of Education.

†In this problem, the size of the regions on which the Krawczyk operator becomes contractive is less than 10^{-17}. Hence, if we apply the Krawczyk-like interval algorithm, then the computation time and the memory space become impracticably large unless regions containing no solution are excluded very effectively.

References

[1] G. Alefeld and J. Herzberger, Introduction to Interval Computations, Academic Press, New York, 1983.

[2] G. Alefeld and J. Herzberger, "A quadratically convergent Krawczyk-like algorithm," SIAM J. Numer. Anal., 20, pp. 210–219, 1983.

[3] E. Allgower and K. Georg, "Simplicial and continuation methods for approximating fixed points and solutions to systems of equations", SIAM Rev., 22, pp. 28–85, 1980.

[4] L. O. Chua and P. M. Lin, Computer-Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques, Prentice-Hall, Englewood Cliffs, New Jersey, 1975.

[5] E. R. Hansen, "A globally convergent interval method for computing and bounding real roots," BIT, 18, pp. 415–424, 1978.

[6] E. R. Hansen and S. Sengupta, "Bounding solutions of systems of equations using interval analysis," BIT, 21, pp. 203–211, 1981.

[7] M. Kashiwagi and S. Oishi, "Numerical validation method for nonlinear equations using interval analysis and rational arithmetic," IEICE Trans., J77-A, pp. 1372–1382, 1994.

[8] R. B. Kearfott, "Some tests of generalized bisection," ACM Trans. Math. Software, 13, pp. 197–220, 1987.

[9] R. B. Kearfott, Preconditioners for the interval Gauss-Seidel method," SIAM J. Numer. Anal., 27, pp. 804–822, 1990.

[10] R. B. Kearfott, "Interval arithmetic techniques in the computational solution of nonlinear systems of equations: Introduction, examples, and comparisons," in Computational Solution of Nonlinear Systems of Equations (Lectures in Applied Mathematics, Vol. 26), E. L. Allgower and K. Georg, eds., American Mathematical Society, Providence, RI, pp. 337–357, 1990.

[11] R. B. Kearfott and V. Kreinovich, Applications of interval computations, Kluwer Academic Publishers, Dordrecht, 1996.

[12] R. Krawczyk, "Newton-algorithmen zur bestimmung von nullstellen mit fehlerschranken," Computing, 4, pp. 187–201, 1969.

[13] R. E. Moore, "A test for existence of solutions to nonlinear systems," SIAM J. Numer. Anal., 14, pp. 611–615, 1977.

[14] R. E. Moore, Methods and Applications of Interval Analysis, SIAM Studies in Applied Mathematics, Philadelphia, 1979.

[15] R. E. Moore and S. T. Jones, "Safe starting regions for iterative methods," SIAM J. Numer. Anal., 14, pp. 1051–1065, 1977.

[16] R. E. Moore and L. Qi, "A successive interval test for nonlinear systems," SIAM J. Numer. Anal., 19, pp. 845–850, 1982.

[17] J. J. Moré, "A collection of nonlinear model problems," in Computational Solution of Nonlinear Systems of Equations (Lectures in Applied Mathematics, Vol. 26), E. L. Allgower and K. Georg, eds., American Mathematical Society, Providence, RI, pp. 723–762, 1990.

[18] A. Neumaier, "Interval iteration for zeros of systems of equations," BIT, 25, pp. 256–273, 1985.

[19] A. Neumaier, Interval Methods for Systems of Equations, Cambridge University Press, Cambridge, Eng-

land, 1990.

[20] S. Oishi, "A difficult problem in numerical analysis with guaranteed accuracy is just solved," J. IEICE, 79, pp. 693–695, 1996.

[21] S. Pastore and A. Premoli, "Polyhedral elements: A new algorithm for capturing all the equilibrium points of piecewise-linear circuits," IEEE Trans. Circuits and Systems I, 40, pp. 124–132, 1993.

[22] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, Numerical Recipes in C, The Art of Scientific Computing, Second Edition, Cambridge University Press, New York, 1992.

[23] L. Qi, "A note on the Moore test for nonlinear systems," SIAM J. Numer. Anal., 19, pp. 851–857, 1982.

[24] L. B. Rall, "A comparison of the existence theorems of Kantorovich and Moore," SIAM J. Numer. Anal., 17, pp. 148–161, 1980.

[25] H. Schwandt, "Accelerating Krawczyk-like interval algorithms for the solution of nonlinear systems of equations by using second derivatives," Computing, 35, pp. 355–367, 1985.

[26] H. Schwandt, "Krawczyk-like algorithms for the solution of systems of nonlinear equations," SIAM J. Numer. Anal., 22, pp. 792–810, 1985.

[27] J. M. Shearer and M. A. Wolfe, "An improved form of the Krawczyk-Moore algorithm," Applied Math. Comp., 17, pp. 229–239, 1985.

[28] J. M. Shearer and M. A. Wolfe, "Some computable existence, uniqueness, and convergence tests for nonlinear systems," SIAM J. Numer. Anal., 22, pp. 1200–1207, 1985.

[29] M. Tadeusiewicz and K. Glowienka, "A contraction algorithm for finding all the DC solutions of piecewise-linear circuits," J. Circuits, Systems, and Computers, 4, pp. 319–336, 1994.

[30] M. A. Wolfe, "A modification of Krawczyk's algorithm," SIAM J. Numer. Anal., 17, pp. 376–379, 1980.

[31] K. Yamamura, "Simple algorithms for tracing solution curves," IEEE Trans. Circuits and Systems I, 40, pp. 537–541, 1993.

[32] K. Yamamura, "Finding all solutions of piecewise-linear resistive circuits using simple sign tests," IEEE Trans. Circuits and Systems I, 40, pp. 546–551, 1993.

[33] K. Yamamura, "An algorithm for representing functions of many variables by superpositions of functions of one variable and addition," IEEE Trans. Circuits and Systems I, 43, pp. 338–340, 1996.

[34] K. Yamamura, "An algorithm for representing nonseparable functions by separable functions," IEICE Trans. Fundamentals, E79-A, pp. 1051–1059, 1996.

[35] K. Yamamura and K. Horiuchi, "A globally and quadratically convergent algorithm for solving nonlinear resistive networks," IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, 9, pp. 487–499, 1990.

[36] K. Yamamura, H. Kawata, A. Tokue, and T. Sekiguchi, "Algorithms for finding stable operating points of nonlinear resistive circuits," IEICE Technical Report, NLP95-61, pp. 17–23, Oct. 1995.

[37] K. Yamamura, A. Tokue, and H. Kawata, "Interval analysis using linear programming," IEICE Technical Report, CAS96-6, pp. 37–43, June 1996.

[38] K. Yamamura, A. Tokue, and H. Kawata, "Interval analysis using linear programming," in Proc. Int. Symp. Nonlinear Theory and its Applications, Kochi, Japan, pp. 49–52, Oct. 1996.

[39] K. Yamamura and M. Ochiai, "An efficient algorithm for finding all solutions of piecewise-linear resistive circuits," IEEE Trans. Circuits and Systems I, 39, pp. 213–221, 1992.

[40] K. Yamamura and T. Ohshima, "Finding all solutions of piecewise-linear resistive circuits using linear programming," Proc. 1995 Int. Symp. Nonlinear Theory and its Applications, 2, pp. 775–780, 1995.