2004 International Symposium on Nonlinear
Theory and its Applications (NOLTA2004)
Fukuoka, Japan, Nov. 29 - Dec. 3, 2004

An Efficient Algorithm for Finding All Solutions of Systems of Nonlinear Equations

Kiyotaka Yamamura and Akinori Machida

Chuo University, Tokyo, 112-8551 Japan Email: yamamura@elect.chuo-u.ac.jp

Abstract—An efficient algorithm is proposed for finding all solutions of systems of nonlinear equations. This algorithm is based on interval analysis, the dual simplex method, the contraction method, and a special technique which makes the algorithm not require large memory space. By numerical examples, it is shown that the proposed algorithm could find all solutions of a system of 2 000 nonlinear equations in acceptable computation time.

1 Introduction

Finding all solutions of nonlinear equations is an important problem which is widely encountered in science and engineering. In this paper, we discuss the problem of finding all solutions of a system of n nonlinear equations:

$$\begin{aligned} f_1(x_1, x_2, \ldots, x_n) &= 0 \\ f_2(x_1, x_2, \ldots, x_n) &= 0 \\ &\vdots \\ f_n(x_1, x_2, \ldots, x_n) &= 0 \end{aligned} \quad (1)$$

contained in a bounded rectangular region D in R^n, where $f_1, f_2, \ldots, f_n$ are real-valued nonlinear functions.

As a computational method to find all solutions of nonlinear equations, interval analysis based techniques are well-known, and various algorithms based on interval computation have been developed [1]–[4]. Using the interval algorithms, all solutions of (1) contained in $D \subset R^n$ can be found with mathematical certainty. However, the computation time of the interval algorithms tends to grow exponentially with the dimension n. One of the difficulties of these algorithms is that the number of boxes [1] to be analyzed is extremely large for large scale problems. Therefore, it is necessary to develop a powerful test for nonexistence of a solution in a given box so that we can exclude many boxes containing no solution at an early stage of the algorithm.

In [3], a powerful computational test was proposed for nonexistence of a solution to the system of nonlinear equations (1) in a given box $X \subseteq D$. This test is termed the LP test. The basic idea of this test is to formulate a linear programming (LP) problem whose feasible region contains all solutions in X. Hence, if the feasible region is empty (which can be easily checked by the simplex method), then X contains no solution, and we can exclude it from further consideration. The LP problem was much improved by introducing the dual simplex method [4],[5]. Using the dual simplex method, the LP test becomes not only powerful but also efficient and requires only a few pivotings per box. In [4], this improved LP test is introduced to the Krawczyk-Moore algorithm [1], which succeeded in finding all solutions to systems of nonlinear equations with $n = 200$. At the present time, the algorithm proposed in [4] is one of the most efficient algorithms for finding all solutions of nonlinear equations.

In this paper, we extend the algorithm in [4], and propose an efficient algorithm for finding all solutions of systems of nonlinear equations.

2 Basic Algorithm

In this section, we first summarize the basic procedures of interval algorithms [1].

An n-dimensional interval vector with components $[a_i, b_i]$ $(i = 1, 2, \ldots, n)$ is denoted by

$$X = ([a_1, b_1], [a_2, b_2], \ldots, [a_n, b_n])^T. \quad (2)$$

Geometrically, X is an n-dimensional box.

In interval algorithms, the following procedure is performed recursively, beginning with the initial box $X = D$. At each level, we analyze the box 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 neither existence nor nonexistence of a solution in X can be proved, then bisect X in some appropriately chosen coordinate direction to form two new boxes; we then continue the above procedure with one of these boxes, and put the other one on a stack for later consideration. Thus, provided the number of solutions of (1)

[1] An n-dimensional rectangular region with the sides parallel to the coordinate axes will be called a box.

contained in $D \subset R^n$ is finite, we can find them all with mathematical certainty.

Next, we summarize the powerful nonexistence test proposed in [3] and [4].

For the simplicity of discussion, in this paper we assume that (1) can be represented as

$$\sum_{j\in J_i} g_{ij}(x_j) + \sum_{j=1}^{n} h_{ij}x_j - s_i = 0, \quad i = 1,2,\ldots,n \quad (3)$$

where $g_{ij}(x_j)$ is a nonlinear function of one variable, h_{ij} and s_i $(i,j = 1,2,\ldots,n)$ are constants, and J_i is a subset of $\{1,2,\ldots,n\}$. Assume that $\sum_{i=1}^{n} |J_i|$ is not large, where $|J_i|$ denotes the cardinality of the set J_i. Note that the discussion in this paper is easily extended to more general systems of nonlinear equations; as for details, see [3].

Let the interval extension of $g_{ij}(x_j)$ over $[a_j, b_j]$ be $[c_{ij}, d_{ij}]$. Then, we introduce auxiliary variables y_{ij} and put $y_{ij} = g_{ij}(x_j)$. If $a_j \le x_j \le b_j$, then $c_{ij} \le y_{ij} \le d_{ij}$.

Now we replace each nonlinear function $g_{ij}(x_j)$ in (3) by the auxiliary variable y_{ij} and the linear inequality $c_{ij} \le y_{ij} \le d_{ij}$, and consider the LP problem:

$$\begin{aligned}
&\max \text{ (arbitrary constant)}\\
&\text{subject to}\\
&\quad \sum_{j\in J_i} y_{ij} + \sum_{j=1}^{n} h_{ij}x_j - s_i = 0, && i = 1,2,\ldots,n\\
&\quad a_i \le x_i \le b_i, && i = 1,2,\ldots,n\\
&\quad c_{ij} \le y_{ij} \le d_{ij}, && i = 1,2,\ldots,n,\ j \in J_i.
\end{aligned} \quad (4)$$

Then, we apply the simplex method to (4).

Evidently, all solutions of (3) which exist in X satisfy the constraints in (4) if we put $y_{ij} = g_{ij}(x_j)$. Hence, if the feasible region of the LP problem (4) is empty, then we can conclude that there is no solution of (3) in X.

The emptiness or nonemptiness of the feasible region of (4) can be checked by the simplex method. If the simplex method terminates with the information that the feasible region is empty, then there is no solution of (3) in X, and we can exclude X from further consideration. This test is called the LP test. It has been shown that if we use directed roundings, then the LP test gives correct results (in the sense that boxes containing solutions are never discarded) [6],[7].

By introducing the LP test to the interval algorithms (such as the Krawczyk-Moore algorithm), all solutions of (3) can be found very efficiently. In [3], this algorithm solves a system of nonlinear equations with $n = 60$ in practical computation time, although the original Krawczyk-Moore algorithm can solve the system only for $n \le 12$.

In [4] and [5], it is shown that the LP test can be performed with a few pivotings (often no pivoting) per box by using the dual simplex method from the second box. In [4], this improved LP test algorithm succeeds to find all solutions of systems of nonlinear equations with $n = 200$.

3 Proposed Algorithm

Although the LP test proposed in [4] requires only a few pivotings per box, it has to be performed on many boxes. In order to decrease the number of boxes on which the LP test is performed, we introduce the contraction method proposed in [2] which contracts a box X to a smaller box $\bar{X}$ containing the same solutions [2].

In the proposed algorithm, we use the algorithm in [4] as the base and perform the contraction method several times after the LP test is performed on a box X. Then, we bisect the reduced box $\bar{X}$ and repeat the same procedure on the sub-boxes.

However, there is one problem in the above algorithm when it is applied to large scale systems. Namely, since the algorithm has the structure of a binary tree, it requires very large memory space. In other words, tableaus of the dual simplex method have to be copied and reserved at each node of the binary tree. Moreover, the time needed for copying tableaus occupies a large part of the total computation time of the algorithm in [4]. Hence, we show a technique which makes the algorithm not require large memory space and not require copying tableaus.

In the implementation of the simplex method to (4), we apply the variable transformation $\bar{x}_i = x_i - a_i$ and $\bar{y}_{ij} = y_{ij} - c_{ij}$, and introduce the slack variables $\bar{\lambda}_i$ and $\bar{\mu}_{ij}$ $(i = 1,2,\ldots,n,\ j \in J_i)$ so that the LP problem is transformed into a standard form:

$$\begin{aligned}
&\max \text{ (arbitrary constant)}\\
&\text{subject to}\\
&\quad \sum_{j\in J_i} \bar{y}_{ij} + \sum_{j=1}^{n} h_{ij}\bar{x}_j - \bar{s}_i = 0, && i = 1,2,\ldots,n\\
&\quad \bar{x}_i + \bar{\lambda}_i = b_i - a_i, && i = 1,2,\ldots,n\\
&\quad \bar{y}_{ij} + \bar{\mu}_{ij} = d_{ij} - c_{ij}, && i = 1,2,\ldots,n,\ j \in J_i\\
&\quad \bar{x}_i \ge 0,\ \ \bar{\lambda}_i \ge 0, && i = 1,2,\ldots,n,\\
&\quad \bar{y}_{ij} \ge 0,\ \bar{\mu}_{ij} \ge 0, && i = 1,2,\ldots,n,\ j \in J_i.
\end{aligned} \quad (5)$$

Then, we construct the initial tableau.

We explain the proposed idea using Figs. 1 and 2. Consider that we have performed the LP test on a box X in Fig. 1 and have obtained an optimal tableau

[2]In [2], an algorithm for finding all solutions of nonlinear equations is proposed using the contraction method. However, the effectiveness of this algorithm to large scale problems is not clear because in the numerical experiments of [2], the algorithm was applied only to small systems with $n \le 10$.

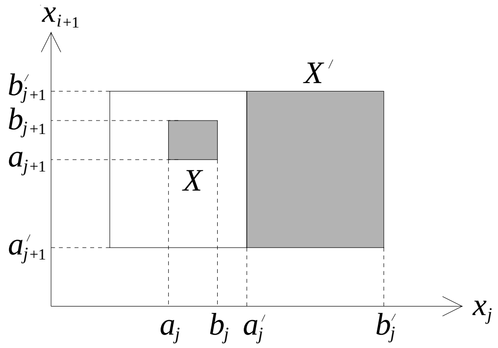

Fig. 1 After X, the LP test is performed on X'.

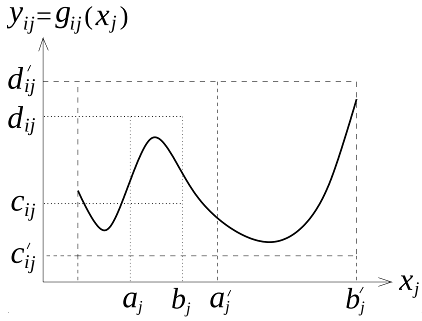

Fig. 2 Illustration of the interval extensions of $g_i(x_i)$ over $[a_i, b_i]$ and $[a'_i, b'_i]$.

for (5). Here, the term *optimal* implies that the optimality condition is satisfied in the *auxiliary objective function low*. Then, consider that we next perform the LP test on a box X'. Let $[c_{ij}, d_{ij}]$ and $[c'_{ij}, d'_{ij}]$ be the interval extensions of $g_{ij}(x_j)$ over $[a_j, b_j]$ and $[a'_j, b'_j]$, respectively, as shown in Fig. 2. In the LP test for X', we similarly introduce auxiliary variables y_{ij} $(i = 1, 2, \ldots, n,\ j \in J_i)$ and consider the LP problem:

$$
\begin{aligned}
&\max\ \text{(arbitrary constant)}\\
&\text{subject to}\\
&\quad \sum_{j\in J_i} y_{ij} + \sum_{j=1}^{n} h_{ij}x_j - s_i = 0, && i = 1, 2, \ldots, n\\
&\quad a'_i \le x_i \le b'_i, && i = 1, 2, \ldots, n\\
&\quad c'_{ij} \le y_{ij} \le d'_{ij}, && i = 1, 2, \ldots, n,\ j \in J_i.
\end{aligned}
\tag{6}
$$

Applying the variable transformation $\tilde{x}_i = x_i - a'_i$ and $\tilde{y}_{ij} = y_{ij} - c'_{ij}$, and introducing the slack variables, (6) is transformed into a standard form:

$$
\begin{aligned}
&\max\ \text{(arbitrary constant)}\\
&\text{subject to}\\
&\quad \sum_{j\in J_i} \tilde{y}_{ij} + \sum_{j=1}^{n} h_{ij}\tilde{x}_j - \tilde{s}_i = 0, && i = 1, 2, \ldots, n\\
&\quad \tilde{x}_i + \tilde{\lambda}_i = b'_i - a'_i, && i = 1, 2, \ldots, n\\
&\quad \tilde{y}_{ij} + \tilde{\mu}_{ij} = d'_{ij} - c'_{ij}, && i = 1, 2, \ldots, n,\ j \in J_i\\
&\quad \tilde{x}_i \ge 0,\ \ \tilde{\lambda}_i \ge 0, && i = 1, 2, \ldots, n,\\
&\quad \tilde{y}_{ij} \ge 0,\ \tilde{\mu}_{ij} \ge 0, && i = 1, 2, \ldots, n,\ j \in J_i.
\end{aligned}
\tag{7}
$$

From Fig. 2, it is clear that $\tilde{x}_i = \bar{x}_i - (a'_i - a_i)$, $\tilde{\lambda}_i = \bar{\lambda}_i - (b_i - b'_i)$, $\tilde{y}_{ij} = \bar{y}_{ij} - (c'_{ij} - c_{ij})$, and $\tilde{\mu}_{ij} = \bar{\mu}_{ij} - (d_{ij} - d'_{ij})$ $(i = 1, 2, \ldots, n,\ j \in J_i)$ hold. Substituting these relations to the previous optimal tableau for (5), the optimal tableau for (7) is easily obtained, which differs from the previous tableau only in the constant column.

Of course, this tableau may not be feasible (i.e., all elements in the constant column may not be non-negative), but always dual feasible because the optimality condition is satisfied. Hence, starting from this tableau, we can perform the dual simplex method and check the existence of the feasible region of (6). Thus, the LP test using the dual simplex method can be performed without copying (reserving) the tableau at each node.

In most cases, this dual simplex method requires only a few pivotings. It often requires no pivoting; namely, if the dual feasible tableau is feasible (i.e., all elements in the constant column are non-negative) or the tableau indicates that the feasible region is empty, then the dual simplex method terminates with no pivoting. Hence, the average number of pivotings per box becomes very small.

This technique also improves the computational efficiency substantially, because as stated before, the time needed for copying tableaus occupies a large part of the total computation time in the conventional algorithm [4].

4 Numerical Examples

We introduced the proposed techniques to the well-known Krawczyk-Moore algorithm [1] and implemented the new algorithm using the programming language C (double precision) on a Sun Blade 2000 (UltraSPARC-III Cu 1.2GHz). In this section, we show some numerical examples.

Example 1: Consider a system of n nonlinear equations:

$$
g(x_i) + x_1 + x_2 + \cdots + x_n - i = 0, \quad i = 1, 2, \ldots, n
$$
$$
\text{where} \quad g(x) = 2.5x^3 - 10.5x^2 + 11.8x
$$

which describes a nonlinear resistive circuit containing n tunnel diodes [2]–[5]. The initial region is $D = ([-10, 10], \ldots, [-10, 10])^T$. Note that the conventional Krawczyk-Moore algorithm could solve this

Table 1 Comparison of computation time (Example 1).

n	S	Ref.[3] T (s)	Ref.[4] T (s)	Proposed T (s)
100	9	71 306	1 060	2
200	13	∞	26 748	21
300	11	∞	∞	65
400	9	∞	∞	124
500	13	∞	∞	333
600	11	∞	∞	543
700	9	∞	∞	632
800	11	∞	∞	1 788
900	19	∞	∞	3 704
1 000	17	∞	∞	5 706
1 100	9	∞	∞	4 387
1 200	9	∞	∞	6 377
1 300	21	∞	∞	22 123
1 400	9	∞	∞	10 741
1 500	13	∞	∞	22 501
1 600	23	∞	∞	58 157
1 700	11	∞	∞	33 638
1 800	9	∞	∞	30 745
1 900	9	∞	∞	41 906
2 000	9	∞	∞	48 805
2 100	11	∞	∞	77 179
2 200	23	∞	∞	173 785
2 300	15	∞	∞	169 243
2 400	9	∞	∞	85 771
2 500	9	∞	∞	136 934

system for $n = 12$ in about three hours, and for $n = 14$ in about 44 h in [3].

Table 1 compares the computation time of the algorithm in [3], that in [4], and the proposed algorithm, where S denotes the number of solutions obtained by the algorithms, T (s) denotes the computation time, and ∞ denotes that it could not be computed in practical computation time or memory over occured. As seen from the table, the proposed algorithm could solve this system for $n = 1\,000$ in about 1.5 h, and for $n = 2\,000$ in about 18 h.

Fig. 3 illustrates the growth of the computation time when n increases. It is seen that the computation time of the proposed algorithm grows exponentially, but not very explosively. It is also seen that the computation time depends largely on the number of solutions.

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

$$x_{i-1} - 2x_i + x_{i+1} + h^2 x^3 = 0, \qquad i = 1, 2, \ldots, n$$

where $x_0 = x_{n+1} = 0$ and $h = 1/(n+1)$. This system comes from a boundary value problem of a nonlinear ordinary differential equation. The initial region is $D = ([0,5], \ldots, [0,5])^T$. Note that the LP test algorithm proposed in [3] could solve this system for $n = 60$ in about 16 h, and the improved LP test algorithm proposed in [4] could solve this system for $n = 120$ in about 25 h. However, the proposed algorithm could solve this system for $n = 1\,000$ in about 5.5 h.

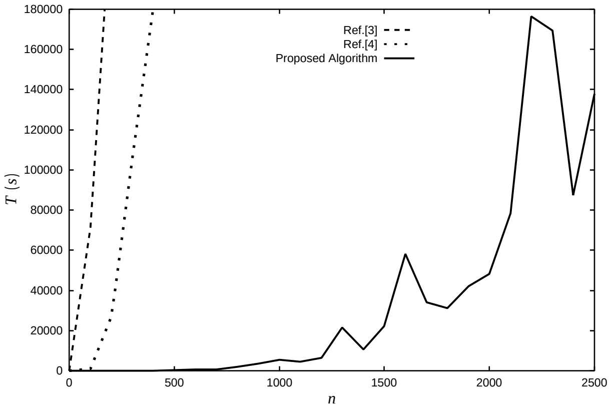

Fig. 3 Computation time of the proposed algorithms.

Example 3: Finally, we solved the transistor circuits shown in Figs. 3–5 of [5]. Then, we found 9, 3, and 11 solutions in 0.08, 0.06, and 1.55 s, respectively. It is seen that all solutions were found in little computation time.

References

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

[2] L. V. Kolev, "An efficient interval method for global analysis of non-linear resistive circuits," *Int. J. Circuit Theory Appl.*, vol. 26, pp. 81–92, Jan. 1998.

[3] K. Yamamura, "Interval solution of nonlinear equations using linear programming," in *Proc. Int. Symp. Circuits Syst.*, Hong Kong, June 1997, pp. 837–840.

[4] K. Yamamura and T. Fujioka, "Finding all solutions of systems of nonlinear equations using the dual simplex method," in *Proc. Int. Symp. Nonlinear Theory and its Applications*, Zao, Japan, Oct. 2001, pp. 219–222.

[5] K. Yamamura and S. Tanaka, "Finding all solutions of piecewise-linear resistive circuits using the dual simplex method," *Int. J. Circuit Theory Appl.*, vol. 30, pp. 567–586, Nov. 2002.

[6] M. Kashiwagi, "Interval arithmetic with linear programming—extension of Yamamura's idea," in *Proc. Int. Symp. Nonlinear Theory and its Applications*, Kochi, Japan, Oct. 1996, pp. 61–64.

[7] M. Kashiwagi, "Simplex method for calculating optimal value with guaranteed accuracy," in *Proc. Int. Symp. Nonlinear Theory and its Applications*, Hawaii, Nov./Dec. 1997, pp. 317–320