An Efficient Algorithm for Finding All Solutions
of Piecewise-Linear Resistive Circuits

群馬大学工学部情報工学科

山村清隆　(Kiyotaka Yamamura)

Abstract   This paper presents an efficient algorithm for finding all solutions of piecewise-linear resistive circuits. First, a technique is proposed which dramatically reduces the number of function evaluations needed in the piecewise-linear modeling process. Then a simple and very efficient sign test is proposed which markedly reduces the number of linear simultaneous equations to be solved for finding all solutions. An effective technique is also introduced which makes the sign test further more efficient. All of our techniques make use of the separability of the nonlinear mappings. Some numarical examples are given, in which all solutions are computed very rapidly. Our algorithm is simple, efficient, and can be easily programmed.

I. INTRODUCTION

Let us first consider nonlinear circuit analysis. A nonlinear resistive circuit can be described by a set of nonlinear equations

$$f_1(x_1,x_2,\cdots,x_n)=0$$
$$f_2(x_1,x_2,\cdots,x_n)=0$$
$$\vdots$$
$$f_n(x_1,x_2,\cdots,x_n)=0 \tag{1}$$

or

$$f(x)=0, \tag{2}$$

where f is a continuous nonlinear mapping from R^n into itself and x is a point in R^n. In the piecewise-linear analysis of nonlinear circuits, the operating region of the nonlinear circuit element is divided into a finite number of subregions, and the nonlinear mapping f is approximated by a piecewise-linear mapping F which is linear on each subregion. Then the system of piecewise-linear equations

$$F(x)=0 \tag{3}$$

can be expressed by the following set of linear simultaneous equations

$$A(m)x+w(m)=0, \qquad m=1,2,\cdots,L, \tag{4}$$

where $A(m)$ is an nxn matrix, $w(m)$ is an n-vector and L is the total number of regions on which F is linear.

Equation (3) or (4) has extensively been studied by many authors. To find all solutions to (3), we generally solve L linear simultaneous equations in (4) and decide whether each solution lies in the assumed region.

Related to this subject, Fujisawa, Kuh and Ohtsuki [1] showed that if the LU-factorization of $A(m)$ is known, the LU-factorization of the matrix which differs from $A(m)$ by rank one can be performed through about $n^2/3$ multiplications. Chua [2] gave a method which calculates the inverse of a matrix which differs from $A(m)$ by one diagonal element through $O(n^2)$ multiplications when $A(m)^{-1}$ is known. Chua and Ying [3] gave an efficient method to find all solutions of (4). In their method, the number of linear simultaneous equations to be solved is reduced by an appropriate sign test, which gives a necessary and sufficient condition on the existence of a solution in each region. Although this method is one of the most efficient methods, the total number of multiplications required exceeds $O(Ln^2)$. Moreover, the number of performance of the sign test exceeds L in the worst case, and the sign test itself is not a simple procedure. Huang and Liu [4] proposed a method which is more efficient than Chua and Ying's method by several times. Nishi [5] proposed a method which requires $O(Ln)$ multiplications to find all solutions for large n. His method seems to be best, but it will be more efficient if the number of regions which can have solutions is reduced by a simple procedure.

In this paper, we present an efficient algorithm for finding all solutions of piecewise-linear resistive circuits. First, a technique is proposed which substantially reduces the number of function evaluations needed in the piecewise-linear modeling process. Then a simple and very efficient sign test is proposed which markedly reduces the number of linear simultaneous equations to be solved for finding all solutions. An effective technique is also introduced which improves the computational efficiency of the sign test. All of our techniques make use of the separability of the nonlinear mapping f. Our algorithm is simple and can be easily programmed, but the effectiveness is very large.

II. PIECEWISE-LINEAR MODELING OF SEPARABLE MAPPINGS

In this paper, we assume that the nonlinear mapping f is separable, that is, there exist $f^i : R^1 \rightarrow R^n$ ($i=1,\cdots,n$) such that

$$f(x) = \sum_{i=1}^{n} f^i(x_i).$$ (5)

Notice that f^i depends on component x_i only. Nonlinear circuit equations have separable mappings if the circuits contain separable elements such as Ebers-Moll transistors or two-terminal resistors. It has been shown that the nonseparable mappings in Shichman-Hodges models of MOSFET's can be transformed into separable mappings by introducing some auxiliary variables

[6]. Therefore most practical resistive circuits are allowed.

It is known that piecewise-linear modeling of separable mappings can be performed on a rectangular subdivision [7]. We first divide the solution domain into rectangular regions whose boundaries are parallel to the co-ordinate axes. In this section, we use two vectors:

$$\mathbf{a}=(a_1,a_2,\cdots,a_n)^T \quad \text{and} \quad \mathbf{b}=(b_1,b_2,\cdots,b_n)^T \qquad (6)$$

to designate a typical rectangle:

$$R=\{\mathbf{x}\in\mathbf{R}^n \,|\, a_i\leq x_i\leq b_i, \quad i=1,2,\cdots,n\}. \qquad (7)$$

Rectangle R has 2^n vertices of the form $\mathbf{v}=(v_1,\cdots,v_n)^T$, where each v_i is either a_i or b_i. Also, any point $\mathbf{x}=(x_1,\cdots,x_n)^T\in R$ can be expressed as

$$x_i=(1-\lambda_i)a_i+\lambda_i b_i, \quad 0\leq\lambda_i\leq 1, \quad i=1,2,\cdots,n \qquad (8)$$

for some unique $\lambda=(\lambda_1,\cdots,\lambda_n)^T$. Given a separable f, let us now define a linear approximation F of f on a rectangle R by

$$F(\mathbf{x})=\sum_{i=1}^{n}[(1-\lambda_i)f^i(a_i)+\lambda_i f^i(b_i)]. \qquad (9)$$

Evidently, F is linear on R. Also, for any vertex $\mathbf{v}\in R$

$$F(\mathbf{v})=f(\mathbf{v}). \qquad (10)$$

Therefore we have created a linear approximation F of f on the rectangle R which equals f on the 2^n vertices of the rectangle. Performing such linearization on each rectangle, we obtain a piecewise-linear approximation F of f on the rectangular subdivision.

The system of linear equations

$$F(\mathbf{x})=0 \qquad (11)$$

can be rewritten as follows:

$$\sum_{i=1}^{n}\lambda_i(f^i(b_i)-f^i(a_i))+\sum_{i=1}^{n}f^i(a_i)=0, \qquad (12)$$

or, in matrix form,

$$A\lambda+\mathbf{w}=0 \qquad (13)$$

where

$$A=[f^1(b_1)-f^1(a_1),\cdots,f^n(b_n)-f^n(a_n)]$$

$$\mathbf{w}=\sum_{i=1}^{n}f^i(a_i). \qquad (14)$$

If the solution of (13) satisfies

$$0\leq\lambda_i\leq 1, \quad i=1,2,\cdots,n, \qquad (15)$$

then the solution

$$x_i = (1-\lambda_i)a_i + \lambda_i b_i, \qquad i=1,2,\cdots,n \qquad (16)$$

lies in the rectangle R.

In piecewise-linear analysis of nonlinear circuits, a system of nonlinear equations is generally given first and then it is approximated by piecewise-linear equations. In this modeling process, it is necessary to determine the function values at all break points. In this section, we shall propose an efficient technique which dramatically reduces the number of function evaluations in the modeling process.

We first give a two-dimensional example to illustrate geometrically the basic idea. Fig.1 shows that the domain of f is partitioned into a rectangular lattice with edges parallel to the co-ordinate axis. Assume that we have already evaluated the function values at 7 vertices a^1, $a^2,\cdots,$ a^7. That is, $f^i(a_i^k)$ (i=1,2, k=1,2,$\cdots$,7) have been computed. Then, since all edges of the rectangular lattice are parallel to the co-ordinate axis, it is easily seen that the function values at all of the vertices can be obtained by simply combining $f^i(a_i^k)$ (i=1,2, k=1,2,$\cdots$,7). For example, the function value at a^{36} is given by $f^1(a_1^3) + f^2(a_2^6)$.

This technique can be extended to the n-dimensional case. Assume that f^i is approximated by k_i line segments. Then, the function values at all of the vertices can be obtained by combining $f^i(a_i^k)$ (i=1,2,$\cdots$,n, k=1,2,$\cdots$, k_i+1). Although there are $(k_1+1)(k_2+1)\cdots(k_n+1)$ vertices in the n-dimensional domain, the number of evaluations of f required in the modeling process is less than $\max\{k_i\}+1$, which is a very small number compared with the total number of vertices $(k_1+1)(k_2+1)\cdots(k_n+1)$. Moreover, once the function values $f^i(a_i^k)$ (i=1,2,$\cdots$,n, k=1,2,$\cdots$,k_i+1) are computed, then the function value at an arbitrary vertex can be easily derived when it is necessary.
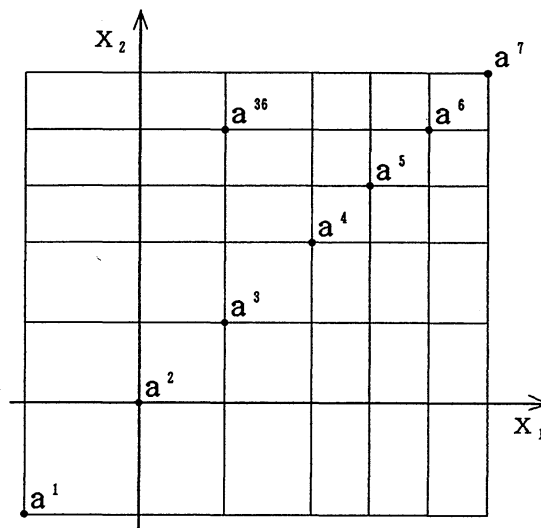

Fig.1 Partitions in the domain of f.

III. SIMPLE SIGN TEST

To find all solutions of (3), we generally solve L linear simultaneous equations in (4) and decide whether each solution lies in the assumed region, where $L=k_1 k_2 \cdots k_n$. Chua and Ying's algorithm requires at least $O(Ln^2)$ multiplications, and Nishi's algorithm requires $O(Ln)$ multiplications. Since L is generally very large, a simple test is required to eliminate regions which do not contain solutions.

Perhaps the simplest test is checking the signs of the functions at the boundary vertices [8],[9]. Let us consider an n-dimensional rectangle R with 2^n vertices. Since F is linear, the following necessary condition is easily verified.

Solution validation test:

If $f_j(v)$ $(=F_j(v))$ has the same signs at all of the 2^n vertices of R for some j, then R contains no solution of (3). □

Therefore we have the followign sign test, which is much simpler than Chua and Ying's test. For any vertex **v** of R, we first evaluate $f_1(v)$ and check the sign. Then we seek another vertex which has a different sign from that of **v**. If there is no such vertex, then (3) does not have a solution in R. If there exists a vertex with a different sign, then we continue the same procedure for f_2, $f_3, \cdots, f_n$, successively. This test allows us to discard a region once the sign test fails for any one of $\{f_j\}$. The idea of checking the signs of the functions at all of the vertices was proposed first by Ushida and Nakamura [8]. However, this test requires 2^n function evaluations per one region in the worst case, and is not so efficient.

Utilizing the separability of **f**, this sign test can be made markedly more efficient. As noted previously, each vertex v_i is either a_i or b_i, and from (5),

$$f(v) = \sum_{i=1}^{n} f^i(v_i).$$ (17)

Therefore it is clear that

$$\max_{v \in R} \{f_j(v)\} = \sum_{i=1}^{n} [\max\{f_j^i(a_i), f_j^i(b_i)\}]$$

$$\min_{v \in R} \{f_j(v)\} = \sum_{i=1}^{n} [\min\{f_j^i(a_i), f_j^i(b_i)\}].$$ (18)

Suppose that $\max\{f_j(v)\}$ and $\min\{f_j(v)\}$ have different signs for $j=1,2,\cdots,n$. In other words, assume that

$$\sum_{i=1}^{n} [\max\{f_j^i(a_i), f_j^i(b_i)\}] \geq 0$$

$$\sum_{i=1}^{n} [\min\{f_j^i(a_i), f_j^i(b_i)\}] \leq 0, \qquad j=1,2,\cdots,n \qquad (19)$$

holds. Then, it is clear that there exist two vertices with different signs, and there may be a solution in R. If (19) does not hold for some j, then the signs of $f_j(v)$ are all the same at 2^n vertices of R, and (3) does not have a solution in R.

In this sign test, the number of function evaluations is only two per one region. If the function values $f^i(a_i^k)$ ($i=1,2,\cdots,n$, $k=1,2,\cdots,k_i+1$) are already computed in the modeling process, then no further function evaluation is required and the test (19) can be performed by additions and comparisons only. It is also worth noting here that this sign test does not require multiplications.

After the sign test, we solve linear simultaneous equations to regions which passed the test. Although our test does not give a necessary and sufficient condition on the existence of a solution as that of Chua and Ying, it is very simple and efficient. As seen in Section VI, most of the regions are discarded by our sign test in many practical applications.

Remark 1: In [8] and [9], an efficient way of performing the sign test is proposed. In general, circuit equations have sparse Jacobians and most of f_j contain only a few variables. Therefore, we should not perform the sign test in the order $f_1, f_2, \cdots, f_n$; instead, we perform the test in the order that functions which contain fewer variables first. For example, consider a system of nonlinear equations

$$f_1(x_1, x_2, x_3, x_4, x_5) = 0$$
$$f_2(x_1, x_2) = 0$$
$$f_3(x_2, x_3, x_4) = 0$$
$$f_4(x_4) = 0$$
$$f_5(x_1, x_3, x_4, x_5) = 0.$$

For this problem, the sign test should be performed in the order f_4, f_2, f_3, f_5, f_1. It is easily seen that the determination of $\max\{f_4(v)\}$ and $\min\{f_4(v)\}$ in (18) requires fewer additions and comparisons than that of $\max\{f_1(v)\}$ and $\min\{f_1(v)\}$, because f_4 contains only one variable. Also, if $\max\{f_4(v)\}$ and $\min\{f_4(v)\}$ have the same signs, the sign tests to f_2, f_3, f_5 and f_1 are not necessary. Therefore, the test will terminate in less computation time in this order. $\qquad \square$

IV. IMPROVING THE EFFICIENCY OF THE SIGN TEST

The sign test proposed in Section III requires at least $2(n-1)$ additions and $n+2$ comparisons per one region. In the worst case, the number of additions is $2n(n-1)$ and the number of comparisons is $n(n+2)$ per one region. (This is the case when the Jacobian matrix of f is full. When the Jacobian matrix is sparse, the number in the worst case is much
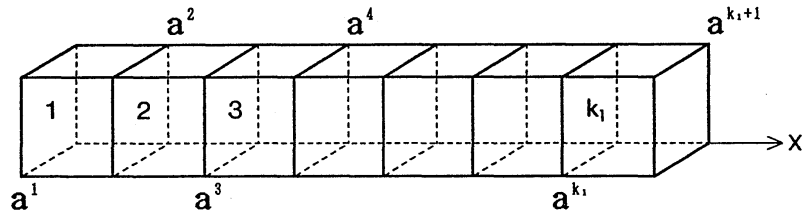
Fig.2 Rectangles arranged in x_1 direction.

smaller.) Although the sign test itself is a simple procedure, the total computation time grows exponentially with the dimension n, because there are $k_1 k_2 \cdots k_n$ regions to be tested. In this section, we propose an effective technique for improving the computational efficiency of the sign test.

Suppose that n-dimensional rectangles are arranged in x_1 direction like Fig.2. Then we first compute $f(a^1)$ and $f(a^2)$ on rectangle 1 and determine p_j and q_j by

$$p_j = \sum_{i=2}^{n} [\max\{f_j^i(a_i^1), f_j^i(a_i^2)\}]$$

$$q_j = \sum_{i=2}^{n} [\min\{f_j^i(a_i^1), f_j^i(a_i^2)\}], \qquad j=1,2,\cdots,n. \tag{20}$$

Then, for $k=1$ to k_1, perform the sign test

$$\max\{f_j^1(a_1^k), f_j^1(a_1^{k+1})\} \geq -p_j$$

$$\min\{f_j^1(a_1^k), f_j^1(a_1^{k+1})\} \leq -q_j, \qquad j=1,2,\cdots,n \tag{21}$$

on rectangle k. Since rectangles are arranged in x_1 direction, it is easily seen that (19) and (21) are equivalent for all k. Therefore if (21) holds, then rectangle k passes the test, and if (21) does not hold for some j, rectangle k contains no solution.

Using this technique, the number of additions becomes zero and the number of comparisons becomes at most 3n (at least 3) from the second rectangle. Hence, the total number of computations is largely reduced.

When n-dimensional rectangles are arranged in all directions, we repeat the above procedure in x_i direction (instead of x_1 direction) in which the number of regions k_i is largest. By choosing such a direction, the number of additions becomes zero and the number of comparisons becomes at most 3n (at least 3) in most of the rectangles.

There is another effective way of choosing a direction of the sign test. That is, we select a variable x_i which is contained in the minimum number of functions, and we perform the sign test in x_i direction. If f_j does not contain x_i, the left-hand side of (21) becomes zero. Therefore, if $p_j \geq 0$ and $q_j \leq 0$ hold, (21) necessarily holds for $k=1,2,\cdots,k_i$, and we can skip the sign test to f_j in this line. Also, if $p_j < 0$ or $q_j > 0$ holds, then

the sign tests in this line are not necessary because f_j does not satisfy
(21) on all rectangles. Hence, if we choose x_i which is not contained in
most of the functions, the number of comparisons in the worst case is
largely reduced. This technique is especially effective when the Jacobian
matrix is sparse.

From the above discussion, we have the following solution validation
test.

New solution validation test:

If there exists a subscript j such that f_j does not contain x_i and
$p_j<0$ or $q_j>0$ holds, then all rectangles in this line do not contain
solutions. Here, x_i is the direction of the sign test. $\square$

When we compute p_j and q_j $(j=1,2,\cdots,n)$ at the beginning of each line,
we check the signs of p_j and q_j if f_j does not contain x_i. If $p_j<0$ or $q_j>0$
holds for some j, we terminate the computation in this line and move to the
next line. This technique reduces the number of regions to be tested, and
makes it possible to apply our algorithm to large scale systems with sparse
Jacobians.

V. ALGORITHM FOR FINDING ALL SOLUTIONS

We now summarize our discussions in the previous sections and state
the algorithm.

Assume a system of nonlinear equations $f(x)=0$ is given.

Step 1 (modeling process)

Divide the solution domain into $k_1k_2\cdots k_n$ rectangular regions, where k_i
denote the number of line segments which approximates f^i. Compute the
function values $f^i(a_i^k)$ $(i=1,2,\cdots,n,\ k=1,2,\cdots,k_i+1)$.

Step 2 (sign test)

Perform the sign test (19) on all regions using the technique proposed
in Section IV. Discard regions which do not pass the test.

Step 3 (computation of solutions)

Construct linear simultaneous equations (12) for regions which passed
the test and solve them.

Remark 2: It is possile to perform Chua and Ying's sign test on regions
which passed the sign test in Step 2. $\square$

VI. NUMERICAL EXAMPLES

We have programmed the proposed algorithm using the C programming
language on a Sun S-4/1 computer. The program became very clear and short
by using the recursive functions.

Example 1

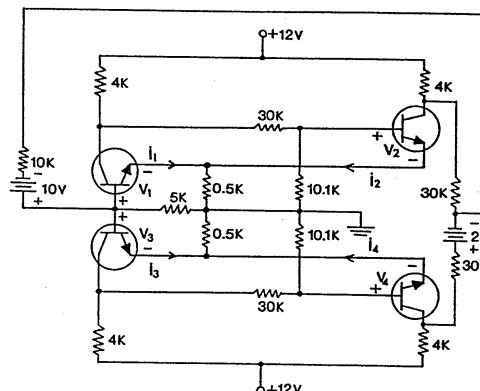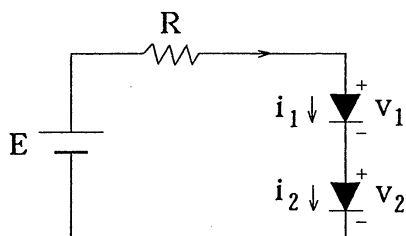Consider the two-tunnel diodes circuit shown in Fig.3 [3],[8]. The

Fig.3 Two-tunnel diode Circuit.   Fig.4 Four-transistor multi-state circuit.

circuit equations are given by

$$E-Rg_1(v_1)-(v_1+v_2)=0$$
$$g_1(v_1)-g_2(v_2)=0$$

where E=30.0, R=13.3 and

$$g_1(v_1)=2.5v_1^3-10.5v_1^2+11.8v_1$$
$$g_2(v_2)=0.43v_2^3-2.69v_2^2+4.56v_2.$$

The operating region we consider is $([0,4],[0,4])^T$. In order to make the number of regions very large, we first divide each [0,4] into 1,000 segments. Thus, there are $1,000^2=1,000,000$ regions. As noted in Section II, function values at all of the vertices can be obtained through 1,001 function evaluations. In this example, we did not use the technique proposed in the last part of Section IV.

Most of the regions were eliminated by the sign test, and only 47 regions passed the test. Then we solved linear simultaneous equations to these regions, and obtained 9 solutions as listed in Table I.

The total computation time was 10.0 seconds, most of which was occupied by the sign test. This is because the sign test was performed 1,000,000 times, while the function evaluation was performed 1,001 times and the linear simultaneous equation was solved 47 times. Although the number of regions is very large, the computation time is very small.

Next, we divided each [0,4] into 10 segments. Then the number of regions was $10^2=100$. In this case, we could also obtain 9 solutions. Of course, the accuracy of these solutions were much lower than that of the solutions in Table I. However, the computation time was very small; only 0.002 seconds.

Example 2

Consider the four-transistor multi-state circuit shown in Fig.4 [3],[8]. The circuit equations are given by

TABLE I		
SOLUTIONS OF EXAMPLE 1		
	v_1	v_2
1	0.19979	3.75421
2	0.21986	1.67295
3	0.22827	0.82863
4	1.66638	0.73935
5	1.70266	1.80903
6	1.77551	3.70718
7	2.22472	3.69304
8	2.27759	1.85749
9	2.30522	0.70557

TABLE II				
SOLUTIONS OF EXAMPLE 2				
	v_1	v_2	v_3	v_4
1	-1.14858	0.36582	0.37725	-4.78518
2	-0.60453	0.36582	-0.60453	0.36582
3	-0.74419	0.36582	0.31449	0.35944
4	0.37507	-4.31655	0.32797	0.35559
5	0.37725	-4.78518	-1.14858	0.36582
6	0.37200	-3.65900	0.37200	-3.65900
7	0.31449	0.35944	-0.74419	0.36582
8	0.32797	0.35559	0.37507	-4.31655
9	0.31861	0.35827	0.31861	0.35827

$$4.36634v_2+6103.168g(v_1)+2863.168g(v_2)-12=0$$
$$5.4v_1+v_3+3580g(v_1)+6220g(v_2)+700g(v_3)+500g(v_4)-22=0$$
$$4.36634v_4+6103.168g(v_3)+2863.168g(v_4)-12=0$$
$$v_1+5.4v_3+700g(v_1)+500g(v_2)+3580g(v_3)+6220g(v_4)-22=0$$

where

$$g(v)=10^{-9}[exp(40v)-1].$$

The operating region we consider is $([-10, 0.45], [-10, 0.45], [-10, 0.45], [-10, 0.45])^T$, and in piecewise-linear modeling, each $[-10, 0.45]$ is divided into 10 segments $[-10, 0], [0, 0.05], [0.05, 0.1], \cdots, [0.35, 0.4]$ and $[0.4, 0.45]$. Thus, there are $10^4=10,000$ regions. Using the technique proposed in Section II, function values at all of the vertices can be obtained through only 11 function evaluations.

When we performed the sign test, 256 regions passed the test, and by solving 256 linear simultaneous equations, we obtained 9 solutions as shown in Table II. The total computation time was 0.2 seconds. Comparing the results with those of the existing methods, the effectiveness of our algorithm is clear.

Next, we divided each $[-10, 0.45]$ into 100 segments. Thus, there are $100^4=100,000,000$ regions. When we performed the sign test, only 161 regions out of 100,000,000 regions passed the test, and by solving 161 linear simultaneous equations, we obtained 9 solutions with higher accuracy. It can be seen that the number of linear simultaneous equations to be solved is decreased although the number of regions to be tested becomes 10,000 times larger.

VII. CONCLUSION

In this paper, we proposed an efficient algorithm which utilizes the separability of the mapping f for finding all solutions of piecewise-linear resistive circuits. The sign test proposed in Sections III and IV is very simple and requires little additional work, but the effectiveness is very

large. The technique proposed in Section II is also effective for decreasing the number of function evaluations in the modeling process.

The proposed algorithm can be made further more efficient by reducing the number of regions to be tested. This can be achieved by exploiting the partial monotonicity of the nonlinear mapping **f**. Details will be discussed in the succeeding paper.

ACKNOWLEDGMENT

REFERENCES

[1] T.Fujisawa, E.S.Kuh, and T.Ohtsuki, "A sparse matrix method for analysis of piecewise-linear resistive networks," IEEE Trans. Circuit Theory, vol.CT-19, pp.571-584, Nov.1972.

[2] L.O.Chua, "Efficient computer algorithms for piecewise-linear analysis of resistive nonlinear networks," IEEE Trans. Circuit Theory, vol.CT-18, pp.73-85, Jan.1971.

[3] L.O.Chua and R.L.P.Ying, "Finding all solutions of piecewise-linear circuits," Int. J. Cir. Theor. Appl., vol.10, pp.201-229, 1982.

[4] Q.Huang and R.Liu, "A simple algorithm for finding all solutions of piecewise-linear networks," IEEE Trans. Circuits Syst., vol.36, pp.600-609, April 1989.

[5] T.Nishi, "An efficient method to find all solutions of piecewise-linear resistive circuits," in Proc. IEEE Int. Symp. on Circuits and Systems, Portland, Oregon, pp.2052-2055, May 1989.

[6] K. Fukuyama, K. Yamamura, and K. Horiuchi, "Application of the rectangular algorithm to non-separable systems of nonlinear equations," Trans. IEICE, vol.J72-A, pp.625-627, (in Japanese), March 1989.

[7] K.Yamamura and K.Horiuchi, "A globally and quadratically convergent algorithm for solving nonlinear resistive networks," IEEE Trans. Comput.-Aided Des. Integrated Cirsuits Syst., vol.9, pp.487-499, May 1990.

[8] A.Ushida and T.Nakamura, "Interval analysis of nonlinear resistive circuits," in Proc. 1989 Joint Technical Conference on Circuits/Systems, Computers and Communications, Sapporo, Japan, pp.499-505, June 1989.

[9] A.Ushida, T.Nakamura, and K.Yamamura, "Finding all solutions of piecewise-linear resistive circuits with Kevorkian decomposition technique," Natl. Conv. IEICE, Tokyo, Japan, 38, (in Japanese), March 1990.