CIRCUIT THEORY AND APPLICATIONS, VOL. 12, 175–178 (1984)

LETTER TO THE EDITOR

FINDING ALL SOLUTIONS OF NON-LINEAR RESISTIVE CIRCUIT EQUATIONS VIA INTERVAL ANALYSIS

L. V. KOLEV

Department of Control, Institute for Mechanical and Electrotechnical Engineering, Darvenitsa, Sofia, Bulgaria

INTRODUCTION

The problem of finding the set of all d.c. solutions of non-linear electronic circuits has recently received a great deal of attention among circuit theoreticians in view of its numerous applications. This problem has two versions depending on whether the non-linear resistors are modelled by piecewise-linear functions (PLF problem) or by continuously differentiable functions (CDF problem).

There exist several algorithms[1-3] for solving the former problem in the case where the resistive circuit equations are written in the known hybrid-representation form.

Various methods for handling the CDF problem have been suggested. The most effective algorithms searching for distinct solutions involve numerically integrating some associated system of non-linear ordinary differential equations along a solution curve.[4,5] The CDF problem can be solved only if the solution curve of the circuit investigated consists of one single branch.[4-6] Although random searches will uncover additional solutions for the general case of a multibranched solution curve,[5] these algorithms all share the serious shortcoming that they cannot guarantee the attainment of all solutions.

In this letter a new approach for solving the CDF problem is proposed which appeals to interval analysis techniques.[7,8] An efficient algorithm based on extended interval arithmetic[11] is presented for the general case where the non-linear resistive circuit is described by the equation

$$f(y)=0 \tag{1}$$

where $f: R^n \to R^n$ is a C^1 function. The algorithm presented obtains within prescribed accuracy all the real solution of (1) contained in a bounded region D.

Ordinary interval arithmetic was first introduced for the analysis of linear electrical circuits in Reference 9.

The reader is assumed to be familiar with the basic concepts of interval analysis. Any relevant notions not defined here are discussed in References 7 and 8.

THE INTERVAL ANALYSIS ALGORITHM

The algorithm herein presented is based on an interval version of Newton's method exposed in References 10 and 11.

The initial region D is represented by an interval vector X^0 (a box). As the algorithm proceeds the initial box X^0 is dynamically subdivided into sub-boxes. Let X denote such a sub-box. The following notations are needed in the sequel:

x is the midpoint of X

$J(X)$ is the interval extension of the Jacobian J of (1) with elements $J_{ij} = \partial f_i / \partial x_j$

J_c is a real matrix whose element in position (i, j) is the midpoint of the corresponding interval element $J_{ij}(X)$ of $J(X)$

B is an approximate inverse of J_c.

0098–9886/84/020175–04$01.00

Received 13 October 1983

The interval element $J_{ij}(X)$ is defined as follows[10]

$$J_{ij}(X) = J_{ij}(X_1, \ldots, X_j, x_{j+1}, \ldots, x_n) \tag{2}$$

where X_k ($k = 1, \ldots, j$) is the kth component of X, and x_m ($m = j+1, \ldots, n$) is the corresponding element of x. If the circuit considered does not contain coupled resistors, then (2) is simplified to

$$J_{ij}(X) = J_{ij}(X_j) \tag{2'}$$

The real vector $g = Bf(x)$ and the interval matrix $P = BJ(X)$ are not computed.

According to References 10 and 11 a new set X' with components X_i' is obtained as follows:

$$Y_i = x_i - P_{ii}^{-1}\left[g_i + \sum_{j=1}^{i-1} P_{ij}(X_j' - x_j) + \sum_{j=i+1}^{n} P_{ij}(X_j - x_j)\right] \tag{3}$$

$$X_i' = Y_i \cap X_i \tag{4}$$

As each new component Y_i ($i = 1, \ldots, n$) is computed, it is immediately intersected with X_i so that the newest result X_i' is used in finding $Y_{i+1}, \ldots, Y_n$.

Since the interval P_{ii} may contain zero for one or more values of i, extended interval arithmetic is used to compute Y_i from (3). The computational details related to this case are given in Reference 11.

When $0 \in P_{ii}$ it is possible for X_i' to be composed of two disjoint intervals. If this is the case (say) for X_j' the single interval X_j is used instead of X_j' when computing Y_i as recommended in Reference 11.

If X_j' is composed of two intervals X_i^L and X_i^R for one or more values of i we only retain (for simplicity[10,11]) the index i_0 and the corresponding intervals $X_{i_0}^L$ and $X_{i_0}^R$ with the largest gap between them. Thus the new set X' will be composed of only two boxes: one whose i_0th component is $X_{i_0}^L$ and one whose i_0th component is $X_{i_0}^R$. The other components of the two boxes are the same as those of X for all $i \neq i_0$.

When each X_i' is a single interval ($i = 1, \ldots, n$) then the new set X' represents one single box. If X' is smaller than X the above procedure is applied to X' (renamed X). If $X \subseteq X'$, the current box X is divided in half (along its largest component) and each sub-box is processed separately.

If the intersection X_i' is empty for at least one i, the current box X cannot contain a solution, so X is deleted.

Whenever a current box is divided into two boxes we put one of these new boxes in a list L to be processed and work on the other. Subsequent boxes may also have to be subdivided, thus adding to the list L of boxes yet to be processed. So the number of boxes in L tends to grow initially. Eventually, however, the boxes become small and, more often, a box is entirely eliminated. Thus the number of boxes in the list L finally decreases to the number of all real solutions of (1) contained in the initial box X^0; if (1) has no solution in X^0 the list L becomes finally empty.

According to the interval analysis approach[10,11] it is guaranteed that if the width w of a box from L at the final stage of the algorithms is small enough, it contains a solution of (1). The width w of a box with components $X_i = [\underline{X}_i, \bar{X}_i]$ is defined to be

$$w = \max_i [(\underline{X}_i - \bar{X}_i), i = 1, \ldots, n] \tag{5}$$

The algorithm is terminated when the width w of each box in L becomes less than a prescribed accuracy ε. Then each solution is assumed to be the midpoint of the corresponding box.

THE STEPS OF THE ALGORITHM

Initially the list L of boxes to be processed consists of a single box X^0. The subsequent steps are to be done in the following order except as indicated by branching.

(1) Let $X = X^0$, $l = 1$ and $\nu = 0$ (l is the length of the list L, and ν denotes the number of solutions found so far).

(2) Compute x, $J(X)$, J_c, B, g and P corresponding to X. If J_c happens to be singular and hence B does not exist, go to step 8.

(3) Put $i=1$.

(4) Calculate the interval X_i' using (3) and (4). If X_i' is empty, eliminate X from the list L and reset $l=l-1$. If the new value $l=0$ go to step 12; otherwise choose the most recent box from L, rename it X and go to step 11.

(5) If X_i' consists of a single interval put $i=i+1$. If $i \leq n$ go to step 4; otherwise go to step 7.

(6) If X_i' consists of two subintervals X_i^L and X_i^R, form the gap

$$\text{gap}_i = \underline{X}_i^R - \overline{X}_i^L$$

where $\underline{X}_i^R$ is the lower end of the right-hand subinterval X_i^R and $\overline{X}_i^L$ is the upper end of the left-hand subinterval X_i^L. Store the index i as i_0 and the corresponding intervals $X_{i_0}^L$ and $X_{i_0}^R$ if $\text{gap}_{i_0} > \text{gap}_{i_0-1}$; put $X_i'=X_i$. Set $i=i+1$ and go to step 4 if $i \leq n$; otherwise skip to step 9.

(7) If $X \subseteq X'$ go to step 8. Otherwise put $X=X'$ and skip to step 11.

(8) Find the index j of the largest component X_j of X. Divide X_j into two subintervals $X_j^L=[\underline{X}_j, (\underline{X}_j+\overline{X}_j)/2]$ and $X_j^R=[(\underline{X}_j+\overline{X}_j)/2, \overline{X}_j]$. Form the boxes $X^L=(X_1,\ldots,X_j^L,\ldots,X_n)$ and $X^R=(X_1,\ldots,X_j^R,\ldots,X_n)$. Go to step 10.

(9) Form the boxes $X^L=(X_1,\ldots,X_{i_0}^L,\ldots,X_n)$ and $X^R=(X_1,\ldots,X_{i_0}^R,\ldots,X_n)$.

(10) Add X^R to the list L. Put $l=l+1$ and $X=X^L$.

(11) Compute w for the current X using (5). If $w>\varepsilon$ go to step 2. If $w \leq \varepsilon$, set $\nu=\nu+1$. Print ν, X and its midpoint. Delete X from L and put $l=l-1$.

If $l=0$ the algorithm is terminated and ν is the number of solutions contained in X^0; if $l>0$ choose the most recent box from L, rename it X and go to step 2.

(12) In this case $\nu=0$ and hence there is no d.c. solution of the non-linear circuit considered in the initial region X^0.

Remark 1

According to step 7 of the algorithm we go to step 11 whenever X' is smaller than X. However, if the reduction of the size of X' with regard to X is negligible this would result in slow convergence so that it is better, in this case, to go to step 8.

Remark 2

In order to ensure that the initial box X^0 contains all the d.c. solutions of the circuit considered one can choose X^0 as large as possible. However, this approach will result in greater computer time. Therefore, it is expedient to find limits on each variable with the help of the no-gain property[12] or any other techniques.

AN ILLUSTRATIVE EXAMPLE

To illustrate the efficiency of the present algorithm we shall take up the example 2 from Reference 5. The circuit considered contains two tunnel diodes, a linear resistor and a voltage source. Its equations are:

$$f_1(x_1,x_2)=30-13\cdot3(2\cdot5x_1^3-10\cdot5x_1^2+11\cdot8x_1)-x_1-x_2=0$$

$$f_2(x_1,x_2)=2\cdot5x_1^3-10\cdot5x_1^2+11\cdot8x_1-0\cdot43x_2^3+2\cdot69x_2^2-4\cdot56x_2=0$$

The real Jacobian matrix J has the following elements J_{ij}:

$$J_{11}(x_1,x_2)=-13\cdot3(7\cdot5x_1^2-21x_1+11\cdot8)-1$$

$$J_{12}(x_1,x_2)=-1$$

$$J_{21}(x_1,x_2)=7\cdot5x_1^2-21x_1+11\cdot8$$

$$J_{22}(x_1,x_2)=-1\cdot29x_2^2+5\cdot38x_2-4\cdot56$$

The interval elements $J_{ij}(X)$ are determined by formula (2′):

$$J_{11}(X) = -13{\cdot}3[X_1(7{\cdot}5X_1 - 21) + (11{\cdot}8) - 1]$$

$$J_{12}(X) = -1$$

$$J_{21}(X) = X_1(7{\cdot}5X_1 - 21) + 11{\cdot}8$$

$$J_{22}(X) = X_2(-1{\cdot}29X_2 + 5{\cdot}38) - 4{\cdot}56$$

Note that we have used the so called nested form[8] for the polynomials involved in $J_{ij}(X)$ since this form assures, in general, narrower intervals than the natural interval extension[8] of $J_{ij}(x_1, x_2)$.

We chose $X_1^0 = X_2^0 = [0, 4]$ for the components of X^0 and $\varepsilon = 10^{-5}$ for the accuracy.

A computer program implementing the algorithm was written in FORTRAN-4. Using this program all nine operating points of the circuit were found infallibly within the prescribed accuracy.

To assess the computational efficiency of the present algorithm, the switching-parameter algorithm of Reference 5 was also programmed. It was applied to the example considered using the same four starting points as in Reference 5. It was observed that whereas the second and fourth solution curves pass through all 9 operating points the first and third solution curves pass through only 5 operating points. Moreover, the present algorithm requires far less computer time.

As regards the computer memory requirements of the present algorithm, the bulk of the needed memory volume is determined by a two-dimensional array $V = l_m \times 2n$, where l_m is the maximum length of the list L and n is the number of non-linear equations. It should be borne in mind that l_m may be a very large number for high-dimensional problems.

CONCLUDING REMARKS

A new algorithm for finding all d.c. solutions of non-linear electronic circuits has been presented. The non-linear functions involved in the circuit equations are assumed to be continuously differentiable. The algorithm makes use of interval analysis techniques and, more specifically, is based on an interval version of Newton's method proposed in References 10 and 11.

Unlike other known algorithms the present algorithm guarantees that all solutions contained in an initial bounded hyperrectangular region will be found within a prescribed accuracy. At the same time it requires comparatively lesser computational efforts for low- and medium-size problems. For high-dimensional problems, however, the memory requirements of the algorithm in its present form may be excessive.

REFERENCES

1. L. O. Chua, *Introduction to Non-linear Network Theory*, McGraw-Hill, New York, 1969.
2. L. O. Chua, 'Efficient computer algorithm for piecewise-linear analysis of resistive non-linear networks', *IEEE Trans. Circuit Theory*, **CT-18**, 73–75 (1971).
3. L. O. Chua and L. P. Ying, 'Finding all solutions of piecewise-linear circuits', *Int. J. Cir. Theor. Appl.*, **10**, 201–229 (1982).
4. K. S. Chao, D. K. Lin and C. T. Pan, 'A systematic search method for obtaining multiple solutions of simultaneous non-linear equations', *IEEE Trans. Circuits and Systems*, **CAS-22**, 748–753 (1976).
5. L. O. Chua and A. Ushida, 'A switching-parameter algorithm for finding multiple solutions of non-linear resistive circuits', *Int. J. Cir. Theor. Appl.*, **4**, 215–239 (1976).
6. C. W. Yun and K. S. Chao, 'Simple solution curves of non-linear resistive networks', *Int. J. Cir. Theor. Appl.*, **11**, 47–55 (1983).
7. R. E. Moore, *Interval Analysis*, Prentice-Hall, Englewood Cliffs, 1966.
8. R. E. Moore, *Methods and Applications of Interval Analysis*, SIAM, Philadelphia, 1979.
9. S. Skelboe, 'True worst-case analysis of linear electrical circuits by interval arithmetic', *Report IT 11*, Institute of Circuit Theory and Telecommunications, Technical University of Denmark, Lyngby, 1977.
10. E. Hansen, 'Global optimization using interval analysis—the multi-dimensional case', *Num. Math.*, **34**, 247–270 (1980).
11. E. Hansen and S. Sengupta, 'Bounding solutions of equations using interval analysis', *BIT*, **21**, 203–211 (1981).
12. L. O. Chua, Y. F. Lam and K. A. Stromsoe, 'Qualitative properties of resistive networks containing multi-terminal non-linear elements', *Proc. 1976 IEEE Int. Symp. Circuits and Systems*, Tech. Univ. Munich, Germany, April 27–29, 680–683 (1976).