A SPARSE MATRIX METHOD FOR ANALYSIS OF PIECEWISE-LINEAR RESISTIVE NETWORKS

by

T. Fujisawa, E. S. Kuh and T. Ohtsuki

Memorandum No. ERL-M316

27 December 1971

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

A SPARSE MATRIX METHOD FOR ANALYSIS OF PIECEWISE-LINEAR RESISTIVE NETWORKS

by

T. Fujisawa, E. S. Kuh and T. Ohtsuki

Memorandum No. ERL-M316

27 December 1971

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

A SPARSE MATRIX METHOD FOR ANALYSIS OF PIECEWISE-LINEAR RESISTIVE NETWORKS

by

T. Fujisawa, E. S. Kuh and T. Ohtsuki

Memorandum No. ERL-M316

27 December 1971

A SPARSE MATRIX METHOD FOR ANALYSIS OF PIECEWISE-LINEAR RESISTIVE NETWORKS

T. Fujisawa,[†] E. S. Kuh and T. Ohtsuki[‡]

Department of Electrical Engineering and Computer Sciences
and the Electronics Research Laboratory
University of California, Berkeley, California 94720

ABSTRACT

This paper deals with nonlinear resistive networks which can be characterized by the equation $\underset{\sim}{f}(\underset{\sim}{x}) = \underset{\sim}{y}$ where $\underset{\sim}{f}(\cdot)$ is a continuous, piecewise-linear mapping of R^n into itself. $\underset{\sim}{x}$ is a point in R^n and represents a set of chosen network variables, and $\underset{\sim}{y}$ is an arbitrary point in R^n and represents the input to the network. New theorems on the existence of solutions together with a convergent method for obtaining at least one of the solutions are given. The second part of the paper is concerned with an efficient computational algorithm which is especially suited for analysis of large piecewise-linear networks. The effectiveness of the method in terms of the number of computation and data handling and storage is demonstrated.

Research sponsored by the National Science Foundation, Grant GK-10656X1, The Naval Electronic Systems Command, Contract N00039-71-C-0255, and the Joint Services Electronics Program, Contract F44620-71-C-0087.

[†]T. Fujisawa is on leave from Osaka University, Toyonaka, Osaka, Japan.

[‡]T. Ohtsuki is on leave from the Nippon Electric Co., Kawasaki, Japan.

A Sparse Matrix Method for Analysis of Piecewise-Linear Resistive Networks

T. Fujisawa, E. S. Kuh and T. Ohtsuki

1. Introduction

The problem of analysis and design of large-scale nonlinear resistive networks is becoming of widespread interest. One encounters such a problem not only in electric or electronic circuits but also in hydraulic networks, structural analysis, numerical integration and economic modeling [1-5]. Because of the size of equations usually involved, it is crucial to devise an efficient computational method for finding the solutions. Recently, a number of theoretical results have been found concerning the question of the existence and uniqueness of solutions [6-10]. It turns out that these results not only provide a basic understanding on inherent properties of networks which possess a unique solution but also are of paramount importance in obtaining convergent computational algorithms.

In this paper we shall confine our study to piecewise-linear resistive networks. The paper can be divided into two parts. In the first part (Sections 2 and 3) we extend the theory of piecewise-linear resistive networks as developed by Fujisawa and Kuh in Reference [10]. New theorems on the existence of solutions (not necessarily unique) are presented and a convergent method of finding at least one of the solutions is given. With these results, the applicability of the piecewise-linear

method is enlarged to include not only all networks with unique solution but also a broad class of networks which possess multiple solutions.

The second part of this paper (Sections 4 and 5) is concerned with an efficient computational algorithm which is especially suited for analysis of large piecewise-linear networks. The problem is essentially solving linear equations $\underset{\sim}{A}\beta = \alpha$ for successive $\underset{\sim}{A}$'s. However, there exists a special relation between two successive $\underset{\sim}{A}$'s, which is imposed by a key property of continuous, piecewise-linear functions developed in Section 3. For readers who are primarily interested in the new algorithm, it is not necessary to understand fully the first part of the paper.

The solution method which is based on the idea developed by Bennett [11] is presented in Section 4. The method depends on the conventional Gaussian elimination but takes advantage of the key property of continuous, piecewise-linear functions; consequently, it is more efficient than either of the two existing methods, namely: (i) using Gaussian elimination at each step and (ii) finding the inverse of the modified matrix at each step. The proof of the new formulas together with a quick review of the Gaussian elimination method is given in Appendix 2. In Section 5, we demonstrate the effectiveness of the method when the matrix is sparse. The data structure for the computer program is discussed in some length. An example which illustrates the advantage of the method is given.

2. Existence of solutions.

Piecewise-linear resistive networks are assumed to be characterized by equations of the form:

$$\underset{\sim}{f}(\underset{\sim}{x}) = \underset{\sim}{y} \tag{1}$$

where $\underset{\sim}{f}$ is a continuous, piecewise-linear function which maps R^n into itself. $\underset{\sim}{x}$ is a point in R^n and represents a set of chosen network variables, and $\underset{\sim}{y}$ is an arbitrary point in R^n and represents the input to the network.

For a continuous, piecewise-linear function $\underset{\sim}{f}$, the whole space R^n is divided into a finite number of polyhedral regions by a finite number of boundary hyperplanes. A typical boundary hyperplane can be characterized by the following equation:

$$\underset{\sim}{r}^T\underset{\sim}{x} = \text{const.} \tag{2}$$

where $\underset{\sim}{r}$ is the normal vector of the hyperplane. In each region, say region m (denoted by R_m) the piecewise-linear function $\underset{\sim}{f}$ is represented by linear equations:

$$\underset{\sim}{A}^{(m)}\underset{\sim}{x} + \underset{\sim}{w}^{(m)} = \underset{\sim}{y}, \quad m = 1, 2, \cdots, \ell \tag{3}$$

where $\underset{\sim}{A}^{(m)}$ is a constant n×n matrix (called Jacobian matrix for convenience) and $\underset{\sim}{w}^{(m)}$ is a constant n-vector, both defined in region R_m. It is assumed that there are altogether ℓ regions in the $\underset{\sim}{x}$-space.

It is crucial to note that continuity of the function imposes a constraint on the Jacobian matrices of any two neighboring regions. Suppose that two neighboring regions, say R_1 and R_2, have a common boundary hyperplane H as shown in Fig. 1. For $\underset{\sim}{f}$ to be continuous on H it is necessary and sufficient that on the boundary where

$$\underset{\sim}{r}^T \Delta \underset{\sim}{x} = 0 \tag{4}$$

$$\underset{\sim}{A}^{(2)} \Delta \underset{\sim}{x} = \underset{\sim}{A}^{(1)} \Delta \underset{\sim}{x} \tag{5}$$

Furthermore, it is easy to see that the above constraint is satisfied if and only if

$$\underset{\sim}{A}^{(2)} - \underset{\sim}{A}^{(1)} = \underset{\sim}{c}\underset{\sim}{r}^T \tag{6}$$

where $\underset{\sim}{c}$ is an arbitrary constant n-vector. The above relation indicates that the difference of the Jacobian matrices of two neighboring regions is a dyad of a specific form. This turns out to be a key property of continuous, piecewise-linear functions, which plays a major role in the theory and computation of piecewise-linear networks.

The key property in Eq. (6) together with the well-known determinant formula

$$\det(\underset{\sim}{1}+\underset{\sim}{P}\underset{\sim}{Q}) = \det(\underset{\sim}{1}+\underset{\sim}{Q}\underset{\sim}{P}) \tag{7}$$

gives the following useful relation between the determinants of Jacobian matrices of two neighboring regions:

$$\begin{aligned} \det \underset{\sim}{A}^{(2)} &= \det(\underset{\sim}{1}+\underset{\sim}{c}\underset{\sim}{r}^T\underset{\sim}{A}^{(1)^{-1}})\det \underset{\sim}{A}^{(1)} \\ &= K \det \underset{\sim}{A}^{(1)} \end{aligned} \tag{8}$$

where

$$K = 1 + \underset{\sim}{r}^T\underset{\sim}{A}^{(1)^{-1}}\underset{\sim}{c} \tag{9}$$

is a scalar constant.

In the following we shall first review briefly some important concepts of global homeomorphism of continuous, piecewise-linear mappings as developed in Reference [10]. Special attention will be given to properties of the function at a boundary hyperplane. Two new theorems on existence of solutions will be presented.

A continuous function $\underset{\sim}{f}$ is said to be a homeomorphism of R^n onto itself if and only if Equation (1) has a unique solution for all $\underset{\sim}{y}$. The necessary and sufficient conditions for $\underset{\sim}{f}$ to be a homeomorphism as stated by Holzmann and Liu are [7]: (i) $\underset{\sim}{f}$ is a local homeomorphism for all $\underset{\sim}{y}$, and (ii) $\lim \|\underset{\sim}{f}(\underset{\sim}{x})\| = \infty$ as $\|\underset{\sim}{x}\| \to \infty$. The local homeomorphism at $\underset{\sim}{x}$ is defined as follows: There exists a neighborhood of $\underset{\sim}{x}$ which is mapped homeomorphically onto a neighborhood of $\underset{\sim}{f}(\underset{\sim}{x})$. For continuous, piecewise-linear functions it is not difficult to see that condition (ii) is automatically satisfied if, in all regions,

$$\det \underset{\sim}{A}^{(m)} \neq 0, \quad m = 1, 2, \cdots, \ell. \tag{10}$$

With this assumption, we only need to study condition (i). In this respect, Fujisawa and Kuh gave the following theorem [10]:

Theorem 1. Let $\underset{\sim}{f}$ be a continuous, piecewise-linear mapping of R^n into itself. A necessary and sufficient condition for $\underset{\sim}{f}$ to be a homeomorphism of R^n onto itself is that, for any unit vector $\underset{\sim}{\alpha}$ and for any $\underset{\sim}{x} \in R^n$, there exists one and only one nonzero vector $\underset{\sim}{\beta}(\underset{\sim}{\alpha},\underset{\sim}{x})$ such that

$$\underset{\sim}{f}(\underset{\sim}{x}+\nu\underset{\sim}{\beta}) = \underset{\sim}{f}(\underset{\sim}{x}) + \nu\underset{\sim}{\alpha} \tag{11}$$

for all sufficiently small positive ν.

Let us discuss the implication of this theorem. Obviously, if $\underset{\sim}{x}$ is an inner point of a region, say R_m, then $\det \underset{\sim}{A}^{(m)} \neq 0$ guarantees local homeomorphism; and the condition in Eq. (11) of the theorem is satisfied with $\underset{\sim}{\beta} = \underset{\sim}{A}^{(m)^{-1}} \underset{\sim}{\alpha}$.

Next, take the case in which $\underset{\sim}{x}$ lies on one and only one boundary hyperplane (called simple boundary.) With reference to Fig. 1, consider a point $\underset{\sim}{x}^*$ on the simple boundary hyperplane H of the two neighboring regions R_1 and R_2. For a given unit vector $\underset{\sim}{\alpha}$, according to Theorem 1, homeomorphism is equivalent to the existence of a unique nonzero vector $\underset{\sim}{\beta}$ which may lie in R_1, R_2 or on the boundary. As shown in Fig. 1, we define

$$\underset{\sim}{h}_1 \triangleq \underset{\sim}{A}^{(1)^{-1}} \underset{\sim}{\alpha} \tag{12a}$$

in R_1 and

$$\underset{\sim}{h}_2 \triangleq \underset{\sim}{A}^{(2)^{-1}} \underset{\sim}{\alpha} \tag{12b}$$

in R_2. By means of the key property in Eq. (6), we can derive a relation between the two vectors:

$$\begin{aligned} \underset{\sim}{h}_2 = \underset{\sim}{A}^{(2)^{-1}} \underset{\sim}{\alpha} &= (\underset{\sim}{A}^{(1)} + \underset{\sim}{c}\underset{\sim}{r}^T)^{-1} \underset{\sim}{\alpha} \\ &= [\underset{\sim}{A}^{(1)^{-1}} - \underset{\sim}{A}^{(1)^{-1}} \underset{\sim}{c}(\underset{\sim}{r}^T \underset{\sim}{A}^{(1)^{-1}} \underset{\sim}{c}+1)^{-1} \underset{\sim}{r}^T \underset{\sim}{A}^{(1)^{-1}}]\underset{\sim}{\alpha} \\ &= \underset{\sim}{h}_1 - \frac{1}{K} \underset{\sim}{A}^{(1)^{-1}} \underset{\sim}{c}\underset{\sim}{r}^T \underset{\sim}{h}_1 \end{aligned} \tag{13}$$

In the above, we have used the well-known Householder's formula

$$(\underset{\sim}{F}+\underset{\sim}{G}\underset{\sim}{L}\underset{\sim}{M})^{-1} = \underset{\sim}{F}^{-1} - \underset{\sim}{F}^{-1}\underset{\sim}{G}(\underset{\sim}{M}\underset{\sim}{F}^{-1}\underset{\sim}{G}+\underset{\sim}{L}^{-1})^{-1}\underset{\sim}{M}\underset{\sim}{F}^{-1} \tag{14}$$

and the identity in (9), From (13), we obtain

$$\underset{\sim}{r}^T\underset{\sim}{h}_2 = \frac{1}{K}\,\underset{\sim}{r}^T\underset{\sim}{h}_1 \tag{15}$$

where, from (8), we recall that K gives the ratio of the two determinants of the neighboring Jacobian matrices. In the following, we shall assume that $K > 0$, i.e. the determinants of the Jacobian matrices of two neighboring regions have the same sign. For a given $\underset{\sim}{\alpha}$, we compute $\underset{\sim}{h}_1$ according to (12a); clearly, there exist three possibilities, namely: (i) $\underset{\sim}{r}^T\underset{\sim}{h}_1 < 0$, (ii) $\underset{\sim}{r}^T\underset{\sim}{h}_1 > 0$, and (iii) $\underset{\sim}{r}^T\underset{\sim}{h}_1 = 0$.

Under case (i), $\underset{\sim}{r}^T\underset{\sim}{h}_1 < 0$; from Eq. (15) $\underset{\sim}{r}^T\underset{\sim}{h}_2 < 0$, which implies that there exists no such $\underset{\sim}{h}_2$ in R_2. The unique vector $\underset{\sim}{\beta}$ is simply $\underset{\sim}{h}_1$ in R_1. Under case (ii), $\underset{\sim}{r}^T\underset{\sim}{h}_1 > 0$; from Eq. (15) $\underset{\sim}{r}^T\underset{\sim}{h}_2 > 0$, thus the unique $\underset{\sim}{\beta}$ is simply $\underset{\sim}{h}_2$ in R_2. Under case (iii), both $\underset{\sim}{h}_1$ and $\underset{\sim}{h}_2$ lie on the boundary; Eq. (15) requires that $\underset{\sim}{h}_1$ and $\underset{\sim}{h}_2$ must have the same direction, and the continuity of the function requires that $\underset{\sim}{h}_1 = \underset{\sim}{h}_2$. Thus the unique $\underset{\sim}{\beta}$ is the vector $\underset{\sim}{h}_1 = \underset{\sim}{h}_2$ on the boundary.

It is easy to see that if $K < 0$, then Theorem 1 cannot be satisfied. Therefore, we have demonstrated that if a point $\underset{\sim}{x}$ lies on a simple boundary, the condition in Eq. (11) of Theorem 1 is satisfied if and only if $K > 0$, that is, the determinants of Jacobian matrices of the two neighboring regions have the same sign.

The discussion above might suggest that a necessary and sufficient

condition for $\underset{\sim}{f}$ to be a homeomorphism of R^n onto itself is the following: det $\underset{\sim}{A}^{(1)}$, det $\underset{\sim}{A}^{(2)}$, $\cdots$, det $\underset{\sim}{A}^{(\ell)}$ are all positive or all negative. However, a counter example given in [10] indicates that this is not the case. The crux lies in situations that $\underset{\sim}{x}$ is on more than one boundary. Under such a situation, $\underset{\sim}{x}$ may not satisy the condition of Theorem 1 even if all determinants of Jacobian matrices to which $\underset{\sim}{x}$ belongs have the same sign. If the function $\underset{\sim}{f}$ is continuously differentiable, let the Jacobian matrix be $\underset{\sim}{J}$, then det $\underset{\sim}{J}(\underset{\sim}{x}) \neq 0$ implies local homeomorphism at $\underset{\sim}{x}$. However, as far as we know there exists no corresponding statement in the continuous, piecewise-linear case. In Reference [10], Fujisawa and Kuh gave the following sufficient condition for homeomorphism, which, incidently, is of considerable interests from computation point of view as will be indicated in Section 4.

Theorem 2. Let $\underset{\sim}{f}$ be a continuous, piecewise-linear mapping of R^n into itself, and let $\underset{\sim k}{A}^{(m)}$ denote the leading minor of order k of $\underset{\sim}{A}^{(m)}$, that is, the matrix composed of the first k rows and k columns of $\underset{\sim}{A}^{(m)}$. The mapping $\underset{\sim}{f}$ is a homeomorphism of R^n onto itself, if, for each k = 1, 2, $\cdots$, n, the determinants of the k×k matrices

$$\underset{\sim k}{A}^{(1)}, \underset{\sim k}{A}^{(2)}, \cdots, \underset{\sim k}{A}^{(\ell)}$$

do not vanish and have the same sign.

Clearly, the above condition is satisfied by matrices which are positive or negative definite and matrices of Class P[12] (matrices with all positive principal minors). The application of interchange of rows

and columns, can be made to further extend the applicability of the above theorem. It should be emphasized nevertheless that the condition in the theorem is only sufficient but not necessary.

On the other hand, if we back track the determinant sign conditions discussed earlier, we can state the following two theroems concerning not of homeomorphism but of existence of solutions. The proof of the theorems are given in Appendix 1, and their application is given in Section 3.

Theorem 3. Let $\underset{\sim}{f}$ be a continuous, piecewise-linear mapping of R^n into itself. Then there exists at least one solution $\underset{\sim}{x}$ to an arbitary $\underset{\sim}{y}$ if $\det \underset{\sim}{A}^{(m)}$, $m = 1, 2, \cdots, \ell$ are all positive or all negative.

In Section 3, the above condition will be referred to frequently; therefore, for convenience, the condition is henceforth referred to as "condition ($\underset{\sim}{A}$)".

Theorem 4. Under the same assumptions as in Theorem 3, (i.e. condition ($\underset{\sim}{A}$)), for any unit vector $\underset{\sim}{\alpha}$ and any point $\underset{\sim}{x}$, there exists at least one nonzero vector $\underset{\sim}{\beta}$ such that $\underset{\sim}{f}(\underset{\sim}{x}+\nu\underset{\sim}{\beta}) = \underset{\sim}{f}(\underset{\sim}{x}) + \nu\underset{\sim}{\alpha}$ for all sufficiently small positive ν.

3. Solution Method.

In 1965, Katzenelson considered piecewise-linear resistive networks of a special class in which all resistors are uncoupled and are of strictly monotonically increasing type, and he developed a convergent computational algorithm[13]. The piecewise-linear approach was further extended to include more general cases by Chua[14] and Ohtsuki and Yoshida[15]. Recently, the Katzenelson algorithm has been used in computing the solu-

tion of resistive networks of much broader class[8,10]. In particular, Fujisawa and Kuh[10] have shown that the algorithm can be applied to Eq. (1) and it always converges to a solution as long as the equation has a unique solution for an arbitrary input $\underset{\sim}{y}$. In other words, homeomorphism implies the convergence of the algorithm. Then one might ask the following question: Does the algorithm always converge to a solution under the assumption that there exists at least one solution to any input? The answer is no as a later example will illustrate. Instead, we can state the following theorem.

Theorem 5. The Katzenelson algorithm converges within ℓ steps to a solution if Condition ($\underset{\sim}{A}$) is satisfied, i.e.,

$$\det \underset{\sim}{A}^{(m)}, \quad m = 1, \cdots, \ell$$

are all positive or all negative.

In the following paragraphs a proof of the above theorem is given together with a review of the essence of the Katzenelson algorithm.

The problem is to find a solution of Eq. (1) for a given input $\underset{\sim}{y}^*$. To begin, choose an arbitrary point $\underset{\sim}{x}^{(1)}$ which is an inner point of a region, say R_1. In R_1 the equation which characterizes the network is

$$\underset{\sim}{A}^{(1)}\underset{\sim}{x} + \underset{\sim}{w}^{(1)} = \underset{\sim}{y} \tag{16}$$

Substituting $\underset{\sim}{x}^{(1)}$ into the above, we obtain

$$\underset{\sim}{A}^{(1)}\underset{\sim}{x}^{(1)} + \underset{\sim}{w}^{(1)} = \underset{\sim}{y}^{(1)} \tag{17}$$

which usually differs from the given $\underset{\sim}{y}^*$. Denote the line segment joining

implies $K > 0$ because of Eq. (8). Therefore, if condition ($\underset{\sim}{A}$) is satisfied, then

$$\underset{\sim}{x}^{(2)}(\lambda) = \underset{\sim}{x}^{(2)} + \lambda \underset{\sim}{A}^{(2)^{-1}}(\underset{\sim}{y}^{*} - \underset{\sim}{y}^{(2)}), \quad \lambda \geq 0 \tag{19}$$

is the portion of the solution curve in R_2.

In case (ii) in which $\underset{\sim}{x}^{(2)}$ lies on more than one boundary hyperplanes as shown in Fig. 3, we have to make use of Theorem 4 to prove that the solution curve can be extended into a certain region beyond $\underset{\sim}{x}^{(2)}$. Now define $\underset{\sim}{\alpha} \triangleq \underset{\sim}{y}^{*} - \underset{\sim}{y}^{(2)}$, then it follows from Theorem 4 that there exists at least one nonzero vector $\underset{\sim}{\beta}$ such that

$$\underset{\sim}{f}(\underset{\sim}{x}^{(2)} + \nu\underset{\sim}{\beta}) = \underset{\sim}{y}^{(2)} + \nu\underset{\sim}{\alpha} \tag{20}$$

for all sufficiently small positive ν. The underlying assumption is of course that the condition ($\underset{\sim}{A}$) is satisfied. One can determine the region in which $\underset{\sim}{x}^{(2)} + \nu\underset{\sim}{\beta}$ is contained for all sufficiently small positive ν. The region is denoted by R_2. Then, $\underset{\sim}{\beta} = \underset{\sim}{A}^{(2)^{-1}}(\underset{\sim}{y}^{*} - \underset{\sim}{y}^{(2)})$ and, as before, Eq. (19) gives an extension of the solution curve.

As mentioned earlier, a point of the intersection of more than one boundary hyperplanes is called a corner. The observations made above then indicate that if the condition ($\underset{\sim}{A}$) is satisfied the solution curve is always extended into another region when it hits a boundary regardless of whether the curve hits a corner or not. Since there exists only a finite number of regions in the $\underset{\sim}{x}$-space, therefore, the proof of Theorem 5 can be completed if it is shown that the solution curve never comes back to a region which the curve once traversed. This is shown as fol-

$\underset{\sim}{y}^{(1)}$ and $\underset{\sim}{y}^{*}$ in the y-space by $L_{\underset{\sim}{y}}$. The problem is then reduced to one of determining a continuous curve, that is really a polygonal curve, starting with $\underset{\sim}{x}^{(1)}$ in the x-space such that $\underset{\sim}{f}$ gives a one-to-one correspondence between the set of points of the curve in the x-space and the set $L_{\underset{\sim}{y}}$ in the y-space. Then, the other end point of the curve is a solution.

The portion of the solution curve which lies in R_1 is determined by

$$\underset{\sim}{x}^{(1)}(\lambda) = \underset{\sim}{x}^{(1)} + \lambda \underset{\sim}{A}^{(1)^{-1}}(\underset{\sim}{y}^{*} - \underset{\sim}{y}^{(1)}) \qquad (18)$$

where $\lambda \geq 0$ is a parameter. If $\underset{\sim}{x}^{(1)}(1)$ happens to be in R_1, then $\underset{\sim}{x}^{(1)}(1)$ is the desired solution. The line segment joining $\underset{\sim}{x}^{(1)}$ and $\underset{\sim}{x}^{(1)}(1)$ is the solution curve and the algorithm terminates here. If, otherwise, the value of λ has to be determined so that $\underset{\sim}{x}^{(1)}(\lambda)$ lies on the boundary of R_1. Denote such value of λ by $\lambda^{(1)}$ and define $\underset{\sim}{x}^{(2)} = \underset{\sim}{x}^{(1)}(\lambda^{(1)})$ and $\underset{\sim}{y}^{(2)} = \underset{\sim}{f}(x^{(2)})$. The line segment joining $\underset{\sim}{x}^{(1)}$ and $\underset{\sim}{x}^{(2)}$ is thus the first portion of the desired solution curve. The next step is to extend the solution curve beyond $\underset{\sim}{x}^{(2)}$.

There are two cases to be considered separately: (i) $\underset{\sim}{x}^{(2)}$ lies on a simple boundary, and (ii) $\underset{\sim}{x}^{(2)}$ lies on more than one boundary hyperplanes, or stated in another way $\underset{\sim}{x}^{(2)}$ is at a corner.

In case (i), suppose that $\underset{\sim}{x}^{(2)}$ lies on the boundary hyperplane H_1 which is common to regions R_1 and R_2 as shown in Fig. 2. Define $\underset{\sim}{h}_1 \triangleq \underset{\sim}{A}^{(1)^{-1}}(\underset{\sim}{y}^{*}-\underset{\sim}{y}^{(1)})$ and $\underset{\sim}{h}_2 = \underset{\sim}{A}^{(2)^{-1}}(\underset{\sim}{y}^{*}-\underset{\sim}{y}^{(1)}) = c\underset{\sim}{A}^{(2)^{-1}}(\underset{\sim}{y}^{*}-\underset{\sim}{y}^{(2)})$ where c is a positive constant equal to $\|\underset{\sim}{y}^{*}-\underset{\sim}{y}^{(1)}\|/\|\underset{\sim}{y}^{*}-\underset{\sim}{y}^{(2)}\|$. Clearly $\underset{\sim}{r}^{(1)T}\underset{\sim}{h}_1 > 0$, and hence from Eq. (11) $\underset{\sim}{r}^{(1)T}\underset{\sim}{h}_2 > 0$ if $K > 0$. However, condition ($\underset{\sim}{A}$),

lows: Suppose that the solution curve enters region R_k at $\underset{\sim}{x}^{(k)}$ and leaves it at $\underset{\sim}{x}^{(k+1)}$ and that the curve later enters the same region at $\underset{\sim}{x}^{(j)}$ as is illustrated in Fig. 4. Clearly these three points $\underset{\sim}{x}^{(k)}$, $\underset{\sim}{x}^{(k+1)}$ and $\underset{\sim}{x}^{(j)}$ do not lie on a line. Therefore, the two vectors $\underset{\sim}{x}^{(k+1)} - \underset{\sim}{x}^{(k)}$ and $\underset{\sim}{x}^{(j)} - \underset{\sim}{x}^{(k)}$ are linearly independent whereas their images under linear mapping $\underset{\sim}{A}^{(k)}$ are both constant multipliers of vector $\underset{\sim}{y}^{*} - \underset{\sim}{y}^{(1)}$ and hence are linearly dependent. This contradicts the assumption that $\underset{\sim}{A}^{(k)}$ is non-singular. This completes the proof of Theorem 5.

We now present a simple example in which condition ($\underset{\sim}{A}$) is not satisfied, yet the equation has at least a solution for any input. It is seen that the Katzenelson algorithm does not work.

Example. Consider a scalar function f(x) of a single variable x as shown in Fig. 5. It is easily seen that there exists at least one solution of the equation f(x) = y for any given y. There are three regions, namely: $R_1 = (-\infty,0]$, $R_2 = [0,1]$, and $R_3 = [1,\infty)$. In these regions $A^{(1)} = 1$, $A^{(2)} = -1$, and $A^{(3)} = 1$ are the slopes of the respective segments. Thus the function does not satisfy the condition ($\underset{\sim}{A}$). Suppose that the input is $y^{*} = 2$ and that we start at $x^{(1)} = -1$. Then the solution curve starting with $x^{(1)}$ on the real line hits the boundary $x = 0$. The curve can not be extended into region R_2 because the value of f(x) increases for $-1 \leq x \leq 0$ and decreases for $x \geq 0$. This indicates that the existence of solution does not generally imply the convergence of the Katzenelson algorithm.

There remain two computational problems to be investigated. The first one is the corner problem. When a solution curve hits a corner,

there exists, theoretically, at least one vector $\underset{\sim}{\beta}$ as given in Eq. (20). The problem from computational point of view is how to find such a $\underset{\sim}{\beta}$. Fujisawa and Kuh gave a systematic perturbation method which forces the solution curve to cross a simple boundary hyperplane at a time[10]. This method always works but is rather tedious. Probably the best way in practice is to make a small move each time a corner is reached so that the modified solution curve does not hit any corner.

The second computational problem is how to best compute the direction vectors $\underset{\sim}{A}^{(1)^{-1}}(\underset{\sim}{y}^{*}-\underset{\sim}{y}^{(1)})$, $\underset{\sim}{A}^{(2)^{-1}}(\underset{\sim}{y}^{*}-\underset{\sim}{y}^{(2)})$, $\cdots$. The straightforward way of computing inverses $\underset{\sim}{A}^{(1)^{-1}}$, $\underset{\sim}{A}^{(2)^{-1}}$, $\cdots$ each time without using any of the previous information is obviously inefficient. Katzenelson[12] proposed to use the following relation based on the Householder's formula:

$$\underset{\sim}{A}^{(2)^{-1}} = \underset{\sim}{A}^{(1)^{-1}}\left[\underset{\sim}{1} - \frac{1}{K}\,\underset{\sim}{c}\underset{\sim}{r}^{T}\underset{\sim}{A}^{(1)^{-1}}\right] \tag{21}$$

where $\underset{\sim}{A}^{(2)}$ and $\underset{\sim}{A}^{(1)}$ are related to each other as in Eq. (6). Once $\underset{\sim}{A}^{(1)}$ is inverted, the inverses of the succeeding Jacobians $\underset{\sim}{A}^{(2)^{-1}}$, $\underset{\sim}{A}^{(3)^{-1}}$, $\cdots$ are obtained successively by the repeated use of the formula (21). We can continue the process until the accumulation of numerical errors becomes significant. This method takes advantage of the key property in Eq. (6) of continuous piecewise-linear functions and is definitely more efficient than finding the inverse matrices from scratch at each step. However, it should be pointed out that in large-scale networks, the Jacobian matrix is usually sparse. As is well-known, sparse matrix, if it is irreduceable, becomes full after inversion[16]. Since sparse

matrix technique is crucial in solving large systems of equations, any method which depends on finding the inverse of a large matrix is totally impractical. What is needed is the triangular factorization of a matrix based on Gaussian elimination. Thus a conceivable scheme is to use Gaussian elimination each time when we need to solve equations $\underset{\sim}{A}^{(m)}\underset{\sim}{\beta} = \underset{\sim}{\alpha}$. This scheme can be used to exploit the sparsity and thus cut down both the computing time and storage space drastically. However, the conventional Gaussian elimination scheme does not exploit the special relation of successive Jacobian matrices, a key property of continuous piecewise-linear functions.

According to the arguments made above we need to seek a method which not only takes advantage of the sparsity but also uses the key-property of continuous, piecewise-linear functions. In the following section such a method is presented. It is demonstrated that the method is more efficient than the one using Householder's formula in the case of full-matrix. It is also more efficient than Gaussian elimination in the case of sparce-matrix.

Finally, it should be remarked that under condition ($\underset{\sim}{A}$) multiple solutions may exist. By applying the Katzenelson algorithm once, we can only find one of the solutions. The process must be repeated with new initial guesses in order to find the other solutions.

4. Triangularization of matrices modified by dyads.

The problem in this section is to compute the triangular factorization of the matrix

$$\underset{\sim}{A}^{*} = \underset{\sim}{A} + \underset{\sim}{c}\underset{\sim}{r}^{T} \tag{22}$$

from that of $\underset{\sim}{A}$. In Eq. (22)

$$\underset{\sim}{A} = [a_{ij}] \tag{23}$$

is an n×n nonsingular matrix,

$$\underset{\sim}{c} = [c_1, c_2, \cdots, c_n]^T \tag{24}$$

and

$$\underset{\sim}{r} = [r_1, r_2, \cdots, r_n]^T \tag{25}$$

are n-dimensional column vectors. Thus $\underset{\sim}{A}^*$ can be expressed as

$$\underset{\sim}{A}^* = [a_{ij} + c_i r_j] \tag{26}$$

The underlying assumption is that $\underset{\sim}{A}$ has been factored by means of Gaussian elimination and the following form is obtained:

$$\underset{\sim}{U} = \underset{\sim}{T}_n \underset{\sim}{T}_{n-1} \cdots \underset{\sim}{T}_1 \underset{\sim}{A} \tag{27}$$

where $\underset{\sim}{U} = [u_{ij}]$ is an upper triangular matrix with unit elements on the diagonal and $\underset{\sim}{T}_k$ is equal to the identity matrix except for the k-th column which is of the form:

$$[0, \cdots, 0, t_{kk}, t_{k+1,k}, \cdots, t_{kn}]^T \tag{28}$$

The objective is to find the triangular factorization of $\underset{\sim}{A}^*$, if it exists, and to obtain

$$\underset{\sim}{U}^* = \underset{\sim}{T}_n^* \underset{\sim}{T}_{n-1}^* \cdots \underset{\sim}{T}_1^* \underset{\sim}{A}^* \tag{29}$$

based on the known factorization of Eq. (27). It should be mentioned that in 1965 Bennett considered a more general problem of obtaining the triangular factorizationof $\underset{\sim}{A}^* = \underset{\sim}{A} + \underset{\sim}{X}\ \underset{\sim}{G}\ \underset{\sim}{Y}^T$ from that of $\underset{\sim}{A}$, where $\underset{\sim}{X}$ and $\underset{\sim}{Y}$ are $n \times m$ matrices, and $m << n$[11]. In his approach, Bennett assumes that $\underset{\sim}{A}$ has been decomposed into the matrix triple product $\underset{\sim}{L}\ \underset{\sim}{D}\ \underset{\sim}{U}$ where $\underset{\sim}{L}$ is a lower triangular matrix with unit elements on the diagonal , $\underset{\sim}{U}$ an upper triangular matrix with unit elements on the diagonal, and $\underset{\sim}{D}$ a diagonal matrix. After $\underset{\sim}{A}$ is modified to become $\underset{\sim}{A}^*$, the corresponding factorization of $\underset{\sim}{A}^*$ into $\underset{\sim}{L}^*\ \underset{\sim}{D}^*\ \underset{\sim}{U}^*$ is derived. In our problem, we deal only with the case where $\underset{\sim}{A}$ is modified by a dyad $\underset{\sim}{c}\ \underset{\sim}{r}^T$ and we prefer to use the Gaussian elimination form rather than $\underset{\sim}{L}\ \underset{\sim}{D}\ \underset{\sim}{U}$ factorization. The method is presented in this section with its proof in Appendix II. In Section 5, special considerations are given for the sparse matrix case.

Before stating the main theorem which contains all the formulas, we need to introduce three sets of auziliary variables: two lower triangular matrices $\underset{\sim}{C}$ and $\underset{\sim}{R}$, and a diagonal matrix $\underset{\sim}{D}$. First

$$\underset{\sim}{C} = \begin{bmatrix} c_1^{(0)} & & & & & \\ c_2^{(0)} & c_2^{(1)} & & & & \\ c_3^{(0)} & c_3^{(1)} & c_3^{(2)} & & & \\ \vdots & \vdots & \vdots & \ddots & & \\ & & & & c_k^{(k-1)} & \\ & & & & & \ddots \\ c_n^{(0)} & c_n^{(1)} & c_n^{(2)} & \cdots & c_n^{(k-1)} & \cdots\ c_n^{(n-1)} \end{bmatrix} \tag{30}$$

where

$$c_i^{(0)} \triangleq c_i;\ i = 1, 2, \cdots, n \tag{31}$$

$$c_i^{(k)} \triangleq c_i^{(k-1)} + t_{ik} c_k^{(k-1)};\ \begin{cases} k = 1, \cdots, n-1 \\ i = k+1, \cdots, n \end{cases} \tag{32}$$

Note that the first column of $\underset{\sim}{C}$ is the given vector $\underset{\sim}{c}$. The k-th column is given by a vector of n-k+1 dimension, whose elements are specified by the elements of the k-1 th column and t_{ik} of $\underset{\sim}{T}_k$. Similarly, for the $\underset{\sim}{R}$ matrix, we have

$$\underset{\sim}{R} = \begin{bmatrix} r_1^{(0)} & & & & & \\ r_2^{(0)} & r_2^{(1)} & & & & \\ r_3^{(0)} & r_3^{(1)} & r_3^{(2)} & \cdots & r_k^{(k-1)} & \\ \vdots & \vdots & \vdots & & \vdots & \\ r_n^{(0)} & r_n^{(1)} & r_n^{(3)} & \cdots & r_n^{(k-1)} & \cdots\ r_n^{(n-1)} \end{bmatrix} \tag{33}$$

where

$$r_j^{(0)} \triangleq r_j;\ j = 1, 2, \cdots, n \tag{34}$$

$$r_j^{(k)} \triangleq r_j^{(k-1)} - r_k^{(k-1)} u_{kj};\ \begin{cases} k = 1, \cdots, n-1 \\ j = k+1, \cdots, n \end{cases} \tag{35}$$

The diagonal matrix $\underset{\sim}{D}$ is specified by its k-th element

$$d_k \triangleq d_{k-1} + c_k^{(k-1)} r_k^{(k-1)} t_{kk};\ k = 1, 2, \cdots, n \tag{36}$$

$$d_0 \triangleq 1$$

Theorem 6. The modified matrix $\underset{\sim}{A}^*$ can be triangularized if the diagonal

matrix $\underset{\sim}{D}$ is nonsingular, i.e.,

$$d_k \neq 0,\ k = 1, 2, \cdots n \tag{37}$$

and the elements of $\underset{\sim}{T}^*_n \cdots \underset{\sim}{T}^*_k \cdots \underset{\sim}{T}^*_1$ and $\underset{\sim}{U}^*$ are given by

$$t^*_{kk} = \frac{d_{k-1}}{d_k} t_{kk} = t_{kk} - \frac{c_k^{(k-1)} r_k^{(k-1)} t_{kk}^2}{d_k};\ k = 1, 2, \cdots, n \tag{38}$$

$$t^*_{ik} = t_{ik} - \frac{r_k^{(k-1)} t_{kk}}{d_k} c_i^{(k)};\ \begin{cases} i = k+1, \cdots, n \\ k = 1, \cdots, n-1 \end{cases} \tag{39}$$

$$u^*_{kj} = u_{kj} + \frac{c_k^{(k-1)} t_{kk}}{d_k} r_j^{(k)};\ \begin{cases} j = k+1, \cdots, u \\ k = 1, \cdots, n-1 \end{cases} \tag{40}$$

A proof is given in Appendix II.

In this section, sparsity of $\underset{\sim}{A}$ and $\underset{\sim}{A}^*$ is not considered. We treat instead the general full matrix case. We shall first give the complete details of the computation procedure. Next we compare our method with that which is based on the Householder's formula and which is based on the conventional Gaussian elimination.

For efficient use of computer storage, non-trivial entries of $\underset{\sim}{U}$ and $\underset{\sim}{T}_k$'s are packed in a two dimensional array as follows:

$$A(\ ,\ ) = [\underset{\sim}{T} \backslash \underset{\sim}{U}] = \begin{bmatrix} t_{11} & u_{12} & \cdots & u_{1n} \\ t_{21} & t_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ t_{n1} & t_{n2} & \cdots & t_{nn} \end{bmatrix} \tag{41}$$

where A(,) is the name of the array in programming. Two one-dimensional arrays named C( ) and R( ) are used to initially store $\underset{\sim}{c}$ and $\underset{\sim}{r}$, respectively. One more variable named D initially stores d_0. As the computational process proceeds, entries of A(,), C( ), R( ) and D are replaced by new ones successively and A(,) finally gives t^*_{ij} and u^*_{ij} for $\underset{\sim}{A}^*$.

Initially, A(,) contains (41), C( ) contains elements in the given vector $\underset{\sim}{c}$, i.e., $[c_1^{(0)}, \cdots, c_n^{(0)}]$, R( ) contains similarly $[r_1^{(0)}, \cdots, r_n^{(0)}]$ and D contains d_0. In the beginning of the k-th step A(,) contains the following

$$\begin{bmatrix} t^*_{11} & \cdots & u^*_{1,k-1} & u^*_{1,k} & \cdots & u^*_{1,n} \\ & \cdots & & \cdots & & \\ t^*_{k-1,1} & \cdots & t^*_{k-1,k-1} & u_{k-1,k} & \cdots & u^*_{k-1,n} \\ t^*_{k,1} & \cdots & t^*_{k,k-1} & t_{kk} & \cdots & u_{k,n} \\ \vdots & & \vdots & \vdots & \ddots & \vdots \\ t^*_{n,1} & \cdots & t^*_{n,n-1} & t_{n,k} & \cdots & t_{nn} \end{bmatrix} \tag{42}$$

and C( ) contains $[c_1^{(0)}, c_2^{(1)}, \ldots, c_k^{(k-1)}, \ldots, c_n^{(k-1)}]$, R( ) contains $[r_1^{(0)}, r_2^{(1)}, \cdots, r_k^{(k-1)}, \cdots, r_n^{(k-1)}]$ and D contains d_{k-1}. Note that C( ) and R( ) contains, respectively, the diagonal entries of matrices $\underset{\sim}{C}$ and $\underset{\sim}{R}$ for columns 1 $\cdots$ k and the non-zero entries of the k-th column.

At the k-th stage, the following computation is done successively

(i) For i = k+1, $\cdots$, n

$$c_i^{(k)} = c_i^{(k-1)} + t_{ik} c_k^{(k-1)} \tag{43}$$

$$r_i^{(k)} = r_i^{(k-1)} - r_k^{(k-1)} u_{ki} \tag{44}$$

These replace the last (n-k)-entries of C( ) and R( ), respectively.

(ii) $$f_k = t_{kk} c_k^{(k-1)} \tag{45}$$

$$e_k = d_{k-1} \tag{46}$$

$$d_k = d_{k-1} + f_k r_k^{(k-1)} \tag{47}$$

$$g_k = t_{kk}/d_k \tag{48}$$

$$p_k = r_k^{(k-1)} g_k \tag{49}$$

$$q_k = c_k^{(k-1)} g_k \tag{50}$$

The value of d_k replaces the content of D immediately after it has been computed. Of course, the condition $d_k \neq 0$ has to be checked. The variables f_k, d_k (or e_k), g_k, p_k and q_k are considered as temporary variables whose values are saved only in the k-th step and are used to cut down the arithmetic operation.

(iii) Finally t^*_{kk}, $t^*_{k+1,k}$, $\cdots$, $t^*_{n,k}$ and $u^*_{k,k+1}$, $\cdots$, $u^*_{k,n}$ are computed and used for updating the content of array A(,).

$$t^*_{kk} = t_{kk} - p_k f_k \tag{51a}$$

or

$$t^*_{kk} = g_k e_k \tag{51b}$$

For $i = k+1, \cdots, n$

$$t^*_{ik} = t_{ik} - p_k c_i^{(k)} \tag{52}$$

$$u^*_{ki} = u_{ki} + q_k r_i^{(k)} \tag{53}$$

This completes the computation of the k-th step. Finally, at the n-th step, it is seen that only t_{nn} needs to be calculated.

Now comparison is made for the use of Householder's formula and that of our method. Let the number of additions and/or subtractions be N_A, the number of multiplications be N_M and the number of divisions be N_D. For the new method, these numbers are estimated as follows:

$$N_A \sim 2n^2,\ N_M \sim 2n^2 \text{ and } N_D \sim n \tag{54}$$

Table I indicates how much we can gain from the use of the new method compared with that of the Householder's formula in full-matrix case.

If one uses the conventional Gaussian elimination each time in full-matrix case, then the necessary arithmetic operation count amounts to the order of $n^3/3$ whereas in our method it is of order n^2. Table II gives exact operation counts for $n = 6, 7, 8$ and indicates that if $n \geq 7$ the new method is superior. The difference becomes very significant as n is large. However, it should be pointed out that this gain in computing time partly is offset by sacrificing numerical accuracy. If the conventional Gaussian elimination is applied to Jacobian matrices each time, a complete control of numerical error can be achieved, for instance, by using partial or complete pivoting[17]. On the other hand,

Table I. Comparison of Householder's Formula and the New Method.

	Initial inversion or triangularization			Each Modification		
	$\pm$	x	÷	$\pm$	x	÷
Householder's formula	n^3	n^3	n	$3n^2$	$3n^2$	1
New method	$\frac{1}{3}n^3$	$\frac{1}{3}n^3$	n	$2n^2$	$2n^2$	n

in the use of the new method, we have to keep the same pivoting order as the initial Jacobian matrix used at the first step, which may not be optimal for the succeeding matrices. Furthermore, the round-off error tends to accumulate as the process goes on. In this respect the observation in the following paragraphs is important.

In appendix II it is shown that $1/t^*_{11} = (d_1/d_0)(1/t_{11})$, $1/t^*_{22} = (d_2/d_1)(1/t_{22})$, $\cdots$, $1/t^*_{nn} = (d_n/d_{n-1})(1/t_{nn})$ are the pivoting elements for the original $\underset{\sim}{A}$. It is well known[17] that if these pivot elements are small compared with other elements, then the numerical accuracy of triangularization can not be good. Therefore, a plausible policy is to set a lower bound for these ratios d_1/d_0, d_2/d_1, $\cdots$, d_n/d_{n-1}. Once the bound is violated, we scratch the process and restart the process by applying Gaussian elimination to the Jacobian matrix at this stage where a new solution is obtained by using a new pivoting order.

In Appendix II, we point out that the application of the Gaussian elimination is possible if the leading principal minors do not vanish[17]. If otherwise, one has to interchange rows and columns to obtain a triangularization. Under the assumption that the condition ($\underset{\sim}{A}$) is satisfied, the above interchange, i.e., pivoting, is usually necessary. Furthermore, the change in pivoting orders may be needed from time to time as the process progresses. In this respect Theorem 2 is very interesting for it guarantees the applicability of Gaussian elimination without using interchange of rows and columns. As a matter of fact the leading minor of order k of $\underset{\sim}{A}^*$ is the product of d_k and the corresponding minor of $\underset{\sim}{A}$. Hence the condition of Theorem 2 in terms of leading principal minors gurantees the condition in Eq. (37) of Theorem 6.

Table II. Comparison of Gaussian Elimination and the New Method.

	Gaussian		New Method	
n	±	x	±	x
6	55	85	72(66)*	88
7	91	133	98(91)*	117
8	140	196	128(120)*	150

* Using Eq. (51b).

5. Sparse matrix method.

The method described in the preceeding section provides a very efficient means of exploiting sparseness if the sparseness structure of the Jacobian matrices is fixed in the entire course of computation. This is the case in the analysis of piecewise-linear resistive networks. The fixed sparseness implies that $\underset{\sim}{A}$ and $\underset{\sim}{A}^*$ have the same zero locations, or

$$a_{ij} = 0 \Rightarrow c_i r_j = 0 \tag{55}$$

With the same sparsity structure for $\underset{\sim}{A}$ and $\underset{\sim}{A}^*$, assuming no numerical cancellation, we may state that $[T\backslash U]$ and $[\underset{\sim}{T}^*\backslash \underset{\sim}{U}^*]$ have the same sparsity structure, i.e.,

$$t_{ij} \text{ (or } u_{ij}) = 0 \Leftrightarrow t^*_{ij} \text{ (or } u^*_{ij}) = 0 \tag{56}$$

Therefore when the first Jacobian matrix is triangularized, the sparseness structure of the triangular form, that is, zero-nonzero pattern of t_{ij} and u_{ij} is determined. Thereafter we only have to store and operate on nonzero elements.

Before going into the discussion of data structure and computer program, we give a comparison of the arithmetic operation counts of the sparse Gaussian elimination and of the new method. The sparse Gaussian elimination is no different from the usual Gaussian elimination method except that it skips all trivial calculations such as $0 + k = k$ and $0 \cdot k = 0$[18]. Let γ_k denote the number of nonzero entries of the k-th column of $\underset{\sim}{T}_k$, and let ρ_k denote the number of nonzero entries of the k-th row of $\underset{\sim}{U}$. Then, the number of multiplications for sparse Gaussian

elimination is essentially

$$N_1 \sim \sum_{k=1}^{n-1} (\gamma_k \rho_k + \gamma_k + \rho_k), \tag{57}$$

whereas that of the new method, by the most conservative estimate, is

$$N_2 \sim 2 \sum_{k=1}^{n-1} (\gamma_k + \rho_k), \tag{58}$$

From Eqs. (57) and (58) we conclude that with rare exceptions the new method is much more efficient than sparse Gaussian scheme. One of the exceptions is a full tridiagonal matrix case, where both algorithms provide a process of order n. For the purpose of testing the sparse matrix version of the new method, a sparse matrix of order 57, which was posed in Fig. 1 of [18] was processed as an example. The exact number of arithmetic operations involved was counted by computer simulation for ten different pairs of $\underset{\sim}{c}$ and $\underset{\sim}{r}$, which were chosen so that they were consistent with the sparseness structure of $\underset{\sim}{A}$. In Table III the results are summarized and compared with the operation count for sparse Gaussian elimination. Thus a considerable amount of saving on machine time may be expected by the use of our method.

In the computer program developed by the present authors, $c_i^{(k)}$ and $r_j^{(k)}$ are stored in two one-dimensional arrays C( ) and R( ) as in the full-matrix case: However, the following three one-dimensional arrays are used to compactly pack nontrivial entries of t_{ij}, u_{ij}, t^*_{ij} and u^*_{ij}:

DT( ): Initially t_{11}, ..., t_{nn} are stored, and finally replaced by t^*_{11}, ..., t^*_{nn}.

T( ): Initially off-diagonal nonzero entries t_{ij} are stored column-wise, and finally replaced by t^*_{ij}.

U( ): Initially off-diagonal nonzero entries u_{ij} are stored row-wise, and finally replaced by u^*_{ij}.

The arrays T( ) and U( ) pack all nonzero off-diagonal entries t_{ij}, u_{ij}, and therefore we need to devise a method to locate the address where t_{ij} or u_{ij} is stored. To be more specific, we have to locate the address ℓ, i.e. the ℓ-th position in the array T( ), where t_{ik} is stored. In other words, $t_{ik} = T(\ell)$. The following four one-dimensional arrays provide a means necessary for address computation. In order to understand our scheme, the example in Table IV is given. Note that the nonzero elements of the matrix $[\underset{\sim}{T} \backslash \underset{\sim}{U}]$ are marked by cross in (a). The four arrays are defined and illustrated as follows:

ξ( ): $\xi(k)$ gives the position in the array T( ) where the first nonzero entry of $t_{k+1,k}$, ..., t_{nk} is found. In the example, for $k = 1$, $\xi(1) = 1$. For $k = 2$ to 6, there is only one such nonzero entry for each column, thus T( ) stores the nonzero entry successively in order. For $k = 7$, there are two nonzero entries, we have $\xi(7) = 7$ and the next two entries in T( ) contain t_{87} and t_{97}. Therefore, for $k = 8$, $\xi(8) = 9$, and T(9) contains t_{98}. If there are no nonzero entry in the k-th column, $\xi(k)$ gives a special symbol to indicate the fact.*

---

* It should be pointed out that if the original matrix is irreduciable, this never happens.

$\alpha(\ )$: $\alpha(\ell) = i$, if t_{ik} is stored in $T(\ell)$, that is $T(\ell) = t_{ik}$. In the figure (c), for example, $\alpha(1) = 4$, which indicates that $T(1) = t_{41}$. Thus $\alpha(\ )$ gives the designation of the row of the nonzero entry of the lower triangular half of $[\underset{\sim}{T} \backslash \underset{\sim}{U}]$.

The remaining two arrays $\eta(\ )$ and $\beta(\ )$ are similarly defined, which give the address location of u_{ki}.

$\eta(\ )$: $\eta(k)$ gives the position in the array $U(\ )$ where the first nonzero entry of $u_{k,k+1} \cdots, u_{kn}$ is found.

$\beta(\ )$: $\beta(\ell) = i$ if $U(\ell) = u_{ki}$

The order in which nonzero entries of t_{ij}, u_{ij} are stored is consistent with the computation procedure (43)-(53). Therefore, the numerical computation can be performed in the same way as that in the full-matrix case. However, trivial operations due to zero t_{ij}, u_{ij}, are automatically skipped in tracing down $T(\ )$ and $U(\ )$. Other trivial computations caused by zero entries of $c_i^{(k)}$, $r_j^{(k)}$ are also skipped simply by testing whether or not $c_i^{(k)} = 0$ and $r_j^{(k)} = 0$. Unlike sparse Gaussian elimination, address computation in the method developed here is very simple.

The storage requirement for the computer program under consideration can be estimated by the lengths of various one-dimensional arrays. Let m be the number of nonzero elements of t_{ij} and u_{ij}, and n be the order of $\underset{\sim}{A}$. Then the storage requirement is estimated to be

$$(m+2n) \text{ floating-point numbers} + (m+2n) \text{ integers} \tag{58}$$

One of the main advantages of the new method is the surprisingly small storage required to represent the sparseness structure. Note that, if

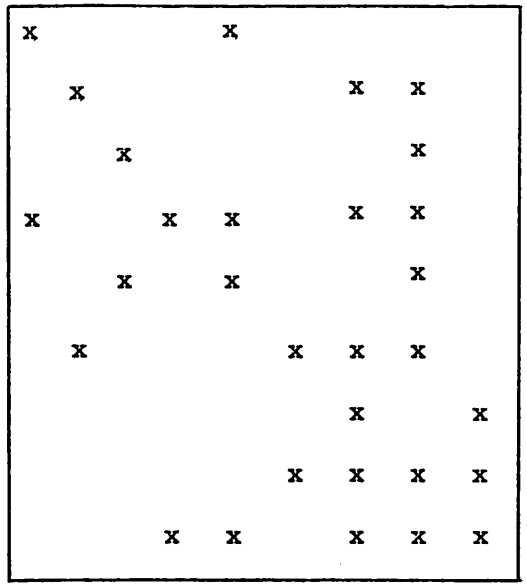
we are to use Gaussian elimination for $\underset{\sim}{A}^*$, it is necessary to have non-trivial searching procedures for computing addresses in order to write data in or to read data out. If one wishes to avoid the search, some other redundant processing is needed. For instance, a program called GNSO[18] generates a long, linear sequence of FORTRAN statements called SOLVE[18]. Once SOLVE is obtained, it can eliminate all searching procedures and thus speed up the process provided that the core memory is not overloaded by its length. As is mentioned in the foregoing section, the pivoting order may have to be changed from time to time. In that case, each time GNSO has to be called for into operation; thus overall economy is questionable.

Another important aspect of our method lies in its easy decomposability of data structure, which is needed if data overloads the core memory. The data are stored linearly from the top to the bottom of T( ) and U( ) in accordance with the sequence of computation in Eqs. (43)-(53). Thus, the interchange of data between core and back up memory is very simple. The ordering of t_{ij} and u_{ij} is also consistent with the forward and backward substitutions. It should be noted that the fundamental operation in Gaussian elimination

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} + t_{ik}a_{kj}^{(k-1)}; \; i,j > k \tag{60}$$

is inconsistent with this decomposability requirement. Generally t_{ik} and $a_{kj}^{(k-1)}$ are stored near the top, while $a_{ij}^{(k)}$'s are located near the bottom of the array.

$[\underset{\sim}{T} \backslash \underset{\sim}{U}] =$



(a)

k	DT(k)
1	t_{11}
2	t_{22}
3	t_{33}
4	t_{44}
5	t_{55}
6	t_{66}
7	t_{77}
8	t_{88}
9	t_{99}

(b)

Table IV

An example to illustrate the address computation

k	ξ(k)
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	9

ℓ	T(ℓ)	α(ℓ)
1	t_{41}	4
2	t_{62}	6
3	t_{53}	5
4	t_{94}	9
5	t_{95}	9
6	t_{86}	8
7	t_{87}	8
8	t_{97}	9
9	t_{98}	9

(c)

k	η(k)
1	1
2	2
3	4
4	5
5	8
6	9
7	11
8	12

ℓ	U(ℓ)	β(ℓ)
1	u_{15}	5
2	u_{27}	7
3	u_{28}	8
4	u_{38}	8
5	u_{45}	5
6	u_{47}	7
7	u_{48}	8
8	u_{58}	8
9	u_{67}	7
10	u_{68}	8
11	u_{79}	9
12	u_{89}	9

(d)

Table III

Cases	nonzero c_i's	nonzero r_j's	±	x
Initial Sparse Gaussian Elimination			665	1237
1	c_1	r_1	396	468
2	c_6	r_{46}	163	208
3	c_2	r_1	376	446
4	c_{43}	r_{18}	169	217
5	c_{57}	r_{57}	2	5
6	c_1, c_2	$r_1, r_2, r_8, r_9, r_{11}, r_{12}$ $r_{14}, r_{43}, r_{44}, r_{45}$	396	468
7	$c_{32}, c_{33}, c_{34}, c_{35}, c_{36}, c_{37}$ $c_{38}, c_{39}, c_{40}, c_{41}, c_{42}$	r_{42}	219	277
8	$c_{19}, c_{20}, c_{21}, c_{22}, c_{29}$	$r_{19}, r_{20}, r_{21}, r_{22}, r_{29}$	379	463
9	$c_{32}, c_{33}, c_{34}, c_{35}, c_{42}$	$r_{32}, r_{33}, r_{34}, r_{35}, r_{42}$	356	434
10	$c_{47}, c_{48}, c_{49}, c_{50}, c_{57}$	$r_{47}, r_{48}, r_{49}, r_{50}, r_{57}$	112	145

Appendix I

Proof of Theorems 3 and 4

Proof of Theorem 3. First some general observations need to be made to clarify the nature of continuous piecewise-linear mapping $\underset{\sim}{f}$. Each region is a convex polyhedron, either bounded or unbounded. It is to be noted that $\underset{\sim}{f}$ is an affine transformation $\underset{\sim}{A}^{(m)}\underset{\sim}{x} + \underset{\sim}{w}^{(m)} = \underset{\sim}{y}$ in each region R_m. It follows that the image of each region under mapping $\underset{\sim}{f}$ is also either a bounded closed polyhedron or an unbounded closed polyhedron. Therefore the range space $\underset{\sim}{f}(R^n)$ is the finite union of convex closed polyhedra.

The proof is by contradiction. Assume that $\underset{\sim}{f}(R^n)$ does not fill up the whole space R^n. Then there exists an (n-1)-dimensional face F which is a boundary of $\underset{\sim}{f}(R^n)$ as is illustrated in Fig. 6. The face F is a finite union of the images of some (n-1)-dimensional boundary faces of regions in the $\underset{\sim}{x}$-space. Hence there exists a point $\underset{\sim}{y}^* \in F$ which lies on F but not on any other faces and a point $\underset{\sim}{x}^*$ which lies on one and only one boundary hyperplane in the $\underset{\sim}{x}$-space such that $\underset{\sim}{y}^* = \underset{\sim}{f}(\underset{\sim}{x}^*)$. Let $\underset{\sim}{\alpha}$ be the outward normal vector of face F. By assumption $\underset{\sim}{y}^*$ is on F but not on any other faces, and hence there exists a positive constant $\varepsilon > 0$ such that

$$\underset{\sim}{y}^* + \nu\underset{\sim}{\alpha} \notin \underset{\sim}{f}(R^n) \text{ for } 0 < \nu < \varepsilon \qquad (I\text{-}1)^*$$

---

* Though the image of each region is convex, their sum $\underset{\sim}{f}(R^n)$ may not be convex. Hence one can not say $\underset{\sim}{x}^* + \nu\underset{\sim}{\alpha} \notin \underset{\sim}{f}(R^n)$ for all $\nu > 0$.

6. Conclusion.

In this paper we considered two problems on piecewise-linear resistive networks. First, we developed two new theorems concerning the existence of solutions and presented a convergent method for finding at least one of the solutions. The result can be concisely stated in terms of the determinants of the Jacobian matrices. Essentially, if, in all regions, the determinants are nonzero and have the same sign, there exists at least one solution for an arbitrary input; and the Katzenelson algorithm can be used to find a solution. This extends the early work of Fujisawa and Kuh on networks with a unique solution. The question to be asked next is of course for the case in which all Jacobian matrices are nonsingular but have either positive or negative determinants.

The second problem which we presented deals with the development of an efficient computational algorithm for solving large, piecewise-linear, resistive networks. The problem is essentially to solve successively linear equations $\underset{\sim}{A}_i \underset{\sim}{\beta} = \underset{\sim}{\alpha}$, for $i = 1, 2, \cdots$, in which two successive $\underset{\sim}{A}_i$'s differ by a matrix of rank one. The method is based on the triangular factorization of modified matrices developed by Bennett, which make use of Gaussian elimination and takes advantage of the key property that $\underset{\sim}{A}_i$'s possess. When, in addition, the matrices are sparse and have the same sparsity structure, the method is extremely efficient in terms of the required number of arithmatic operations as well as data handling and storage. The method can be used for problems other than networks, for example, linear programming.

The computation for our example was done on IBM 1800. We acknowledge the programming assistance of Mr. James Sporal.

Appendix II

Gaussian elimination and proof of Theorem 6.[†]

The nature of Gaussian elimination can be best seen by examining its major first step. Let

$$\underset{\sim}{A} \triangleq \underset{\sim}{A}^{(0)} = [a_{ij}^{(0)}]$$

$$= \left[\begin{array}{c|c} a_{11}^{(0)} & \underset{\sim}{b}^{(0)T} \\ \hline \underset{\sim}{a}^{(0)} & \underset{\sim}{G}^{(0)} \end{array}\right] \qquad \text{(II-1)}$$

where $\underset{\sim}{a}^{(0)}$ and $\underset{\sim}{b}^{(0)}$ are n-1 dimension column vectors and $\underset{\sim}{G}^{(0)}$ is an (n-1) × (n-1) matrix. Assuming that $a_{11}^{(0)}$ is nonzero an elementary row operation results in the following:

$$\underset{\sim}{A}_1 \triangleq \underset{\sim}{T}_1 \underset{\sim}{A} = \left[\begin{array}{c|c} \dfrac{1}{a_{11}^{(0)}} & \underset{\sim}{0} \\ \hline -\dfrac{\underset{\sim}{a}^{(0)}}{a_{11}^{(0)}} & \underset{\sim}{1} \end{array}\right] \underset{\sim}{A}$$

$$= \left[\begin{array}{c|c} 1 & \dfrac{\underset{\sim}{b}^{(0)T}}{a_{11}^{(0)}} \\ \hline \underset{\sim}{0} & \underset{\sim}{A}^{(1)} = [a_{ij}^{(1)}] \end{array}\right] \qquad \text{(II-2)}$$

[†] It should be pointed out that the notation used here is independent of that of Sec. 2 when we considered matrices at different regions. Here superscripts are used to denote iteration steps.

As $\underset{\sim}{x}^*$ lies on one and only one boundary hyperplane, there exists a neighborhood ∇ of $\underset{\sim}{x}^*$ which is mapped homeomorphically onto a neighborhood of $\underset{\sim}{y}^*$. This contradicts Eq. (I-1). Thus the proof is complete.

Proof of Theorem 4. From the arguments in Section 2 it is clear that the statement of the theorem is true if either $\underset{\sim}{x}$ is an interior point or it lies on a simple boundary.

Let us assume that point $\underset{\sim}{x}^*$ lies on mutually distinct boundary hyperplanes $H_1, \cdots, H_k$. The property to prove is a local property at $\underset{\sim}{x}^*$ and hence it can be assumed that hyperplanes other than these above do not exist. This is equivalent to the consideration of a modified continuous piecewise-linear function $\underset{\sim}{f}^*$ which is obtained by extending the local property of $\underset{\sim}{f}$ at $\underset{\sim}{x}^*$ to the whole space R^n. It is clear that if $\underset{\sim}{f}$ satisfies the condition of Theorem 3, then $\underset{\sim}{f}^*$ does too. Therefore $\underset{\sim}{f}^*$ is a homeomorphism of R^n onto itself due to Theorem 3. Therefore, for any unit vector $\underset{\sim}{\alpha}$, there exists at least one non-zero vector $\underset{\sim}{\beta}$ such that

$$\underset{\sim}{f}^*(\underset{\sim}{x}^*+\nu\underset{\sim}{\beta}) = \underset{\sim}{f}^*(\underset{\sim}{x}^*) + \nu\underset{\sim}{\alpha} \qquad \text{(I-2)}$$

for all $\nu > 0$. Since $\underset{\sim}{f}^*$ and $\underset{\sim}{f}$ coincide with each other in a neighborhood of $\underset{\sim}{x}^*$, (I-2) is valid for $\underset{\sim}{f}$ for sufficiently small positive ν. This completes the proof.

$$= \begin{bmatrix} 1 & u_{12} & & & & u_{1n} \\ 0 & 1 & u_{23} & & & u_{2n} \\ & & \cdots & & & \\ & & \cdots & & & \\ 0 & \cdots & 0 & 1 & u_{k,k+1} \cdots & u_{kn} \\ 0 & \cdots & 0 & & & \\ & \cdots & & & \tilde{A}^{(k)} = [a_{ij}^{(k)}] & \\ & \cdots & & & & \\ 0 & \cdots & 0 & & & \end{bmatrix}$$

where

$$t_{kk} = \frac{1}{a_{kk}^{(k-1)}} \tag{II-9}$$

$$t_{ik} = - a_{ik}^{(k-1)} \, t_{kk}; \; i = k+1, \cdots, n \tag{II-10}$$

$$u_{kj} = t_{kk} \, a_{kj}^{(k-1)}; \; j = k+1, \cdots, n \tag{II-11}$$

and

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} + t_{ik} \, a_{kj}^{(k-1)}; \; i, j = k+1, \cdots, n \tag{II-12}$$

In the above, it is seen that the term $a_{kk}^{(k-1)} = \frac{1}{t_{kk}}$ is used as the pivoting element at each step, and, by assumption, it is nonzero. Thus t_{kk} is well defined. It is also of interest to point out the following property of the matrix $\tilde{A}$ in terms of $a_{kk}^{(k-1)} = \frac{1}{t_{kk}}$: The leading principal

where

$$\underset{\sim}{A}^{(1)} = \underset{\sim}{G}^{(0)} - \frac{1}{a_{11}^{(0)}} \underset{\sim}{a}^{(0)} \underset{\sim}{b}^{(0)^T} \qquad \text{(II-3)}$$

According to the terminology used in Section 4 for $\underset{\sim}{T}_k$'s and $\underset{\sim}{U}$, the above relations can be written as

$$t_{11} = \frac{1}{a_{11}^{(0)}} \qquad \text{(II-4)}$$

$$t_{i1} = - a_{i1}^{(0)} \; t_{11}; \; i = 2, \cdots, n \qquad \text{(II-5)}$$

$$u_{ij} = t_{11} a_{ij}^{(0)}; \; j = 2, \cdots, n \qquad \text{(II-6)}$$

and

$$a_{ij}^{(1)} = a_{ij}^{(0)} + t_{i1} \, a_{1j}^{(0)}; \; i, j = 2, \cdots, n \qquad \text{(II-7)}$$

For the k-th step, the general formulas are:

$$\underset{\sim}{A}_k = \underset{\sim}{T}_k \underset{\sim}{A}_{k-1} \qquad \text{(II-8)}$$

$$= \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ & \cdots & & \\ 0 & \cdots & t_{kk} & \cdots \\ & \cdots & t_{k+1,k} & \cdots \\ 0 & \cdots & t_{nk} & \cdots \end{bmatrix} \begin{bmatrix} 1 & u_{12} & \cdots & u_{1n} \\ 0 & 1 & u_{23} & \cdots \\ & & \cdots & \\ 0 & \cdots & 0 & 1 \\ 0 & \cdots & & 0 \\ & \cdots & & \\ & \cdots & & \\ & \cdots & & \\ 0 & \cdots & & 0 \end{bmatrix} \begin{bmatrix} u_{k-1,k} & \cdots & u_{k-1,n} \\ \multicolumn{3}{c}{\underset{\sim}{A}^{(k-1)} = [a_{ij}^{(k-1)}]} \end{bmatrix}$$

$$= \left[\begin{array}{c|c} \dfrac{1}{a_{11}^{(0)} + \frac{1}{d_0} c_1^{(0)} r_1^{(0)}} & \underset{\sim}{0} \\ \hline - \dfrac{\underset{\sim}{a}^{(0)} + \frac{1}{d_0} \underset{\sim}{c}_1^{(0)} r_1^{(0)}}{a_{11}^{(0)} + \frac{1}{d_0} c_1^{(0)} r_1^{(0)}} & \underset{\sim}{1} \end{array}\right] \underset{\sim}{A}^{*}$$

$$= \left[\begin{array}{c|c} 1 & \dfrac{\left(\underset{\sim}{b}^{(0)} + \frac{1}{d_0} c_1^{(0)} \underset{\sim}{r}_1^{(0)}\right)^T}{a_{11}^{(0)} + \frac{1}{d_0} c_1^{(0)} r_1^{(0)}} \\ \hline \underset{\sim}{0} & \underset{\sim}{A}^{*(1)} = [a_{ij}^{*(1)}] \end{array}\right] \qquad \text{(II-16)}$$

where

$$\underset{\sim}{A}^{*(1)} = \underset{\sim}{G}^{(0)} + \frac{1}{d_0} \underset{\sim}{c}_1^{(0)} \underset{\sim}{r}_1^{(0)T} - \frac{\left(\underset{\sim}{a}^{(0)} + \frac{1}{d_0} \underset{\sim}{c}_1^{(0)} r_1^{(0)}\right)\left(\underset{\sim}{b}^{(0)} + \frac{1}{d_0} c_1^{(0)} \underset{\sim}{r}_1^{(0)}\right)^T}{a_{11}^{(0)} + \frac{1}{d_0} c_1^{(0)} r_1^{(0)}}$$

$$= \underset{\sim}{A}^{(1)} + \frac{\left(\underset{\sim}{c}_1^{(0)} - \frac{c_1^{(0)}}{a_{11}^{(0)}} \underset{\sim}{a}^{(0)}\right)\left(\underset{\sim}{r}_1^{(0)} - \frac{r_1^{(0)}}{a_{11}^{(0)}} \underset{\sim}{b}^{(0)}\right)^T}{a_{11}^{(0)} + \frac{1}{d_0} c_1^{(0)} r_1^{(0)}}$$

$$= \underset{\sim}{A}^{(1)} + \frac{1}{d_1} \underset{\sim}{c}^{(1)} \underset{\sim}{r}^{(1)T} \qquad \text{(II-17)}$$

In (II-17)

$$\underset{\sim}{c}^{(1)} \triangleq \underset{\sim}{c}_1^{(0)} - \frac{c_1^{(0)}}{a_{11}^{(0)}} \underset{\sim}{a}^{(0)} \qquad \text{(II-18)}$$

minor of order k, i.e. the determinant which is formed by the first k rows and first k columns of $\underset{\sim}{A}$, is equal to

$$a_{11}^{(0)} a_{22}^{(1)} \cdots a_{kk}^{(k-1)} = \frac{1}{t_{11} t_{22} \cdots t_{kk}} . \qquad \text{(II-13)}$$

Thus another way of stating that t_{kk} is well defined for k = 1, 2, $\cdots$, n is the nonvanishing of all the leading principal minors of $\underset{\sim}{A}$.

The problem at hand is to obtain the Guassian triangularization of $\underset{\sim}{A}^*$ where

$$\underset{\sim}{A}^* \triangleq \underset{\sim}{A}^{*(0)} = \underset{\sim}{A}^{(0)} + \underset{\sim}{c}\underset{\sim}{r}^T \triangleq \underset{\sim}{A}^{(0)} + \frac{1}{d_0} \underset{\sim}{c}^{(0)} \underset{\sim}{r}^{(0)T} \qquad \text{(II-14)}$$

As in Eq. (II-1), we may write $\underset{\sim}{A}^{*(0)}$ as follows:

$$\underset{\sim}{A}^{*(0)} = \left[\begin{array}{c|c} a_{11}^{(0)} + \frac{1}{d_0} c_1^{(0)} r_1^{(0)} & \underset{\sim}{b}^{(0)T} + \frac{1}{d_0} c_1^{(0)} \underset{\sim}{r}_1^{(0)T} \\ \hline \underset{\sim}{a}^{(0)} + \frac{1}{d_0} \underset{\sim}{c}_1^{(0)} r_1^{(0)} & \underset{\sim}{G}^{(0)} + \frac{1}{d_0} \underset{\sim}{c}_1^{(0)} \underset{\sim}{r}_1^{(0)T} \end{array}\right] \qquad \text{(II-15)}$$

where $\underset{\sim}{c}_1^{(0)} \triangleq [c_2^{(0)}, \cdots, c_n^{(0)}]^T$ is the (n-1)-vector containing the last n-1 elements of $\underset{\sim}{c}^{(0)}$ and $\underset{\sim}{r}_1^{(0)} \triangleq [r_2^{(0)}, \cdots, r_n^{(0)}]^T$ is the (n-1)-vector containing the last n-1 elements of $\underset{\sim}{r}^{(0)}$. Then the major first step for $\underset{\sim}{A}^{*(0)}$ is

$$\underset{\sim}{A}_1^* = \underset{\sim}{T}_1^* \underset{\sim}{A}^*$$

and

$$\underset{\sim}{r}^{(1)} \triangleq \underset{\sim}{r}_1^{(0)} - \frac{r_1^{(0)}}{a_{11}^{(0)}} \underset{\sim}{b}^{(0)} \tag{II-19}$$

Comparing these with Eqs. (30)-(35), we recognize that $\underset{\sim}{c}^{(1)}$ is the second column vector of the triangular matrix $\underset{\sim}{C}$ and $\underset{\sim}{r}^{(1)}$ is the second column vector of $\underset{\sim}{R}$ in Section 4. Furthermore using these formulas in Section 4 and Eqs. (II-4)-(II-7), we can easily prove the following:

$$t_{11}^{*} = \frac{d_0}{d_1} t_{11} = t_{11} - \frac{c_1^{(0)} r_1^{(0)} t_{11}^2}{d_1} \tag{II-20}$$

$$- \frac{\underset{\sim}{a}^{(0)} + \frac{1}{d_0} c_1^{(0)} r_1^{(0)}}{a_{11}^{(0)} + \frac{1}{d_0} c_1^{(0)} r_1^{(0)}} = - \frac{\underset{\sim}{a}^{(0)}}{a_{11}^{(0)}} - \frac{r_1^{(0)} \underset{\sim}{c}^{(1)}}{d_1 a_{11}^{(0)}} \tag{II-21}$$

$$\frac{\underset{\sim}{b}^{(0)} + \frac{1}{d_0} c_1^{(0)} \underset{\sim}{r}_1^{(0)}}{a_{11}^{(0)} + \frac{1}{d_0} c_1^{(0)} r_1^{(0)}} = \frac{\underset{\sim}{b}^{(0)}}{a_{11}^{(0)}} + \frac{c_1^{(0)} \underset{\sim}{r}^{(1)}}{d_1 a_{11}^{(0)}} \tag{II-22}$$

From (II-21) and (II-22), if we take the i-th term and the j-th term, respectively, we obtain

$$t_{i1}^{*} = t_{i1} - \frac{r_1^{(0)} t_{11}}{d_1} c^{(1)}; \; i = 2, \cdots, n \tag{II-23}$$

and

$$u_{ij}^{*} = u_{ij} + \frac{c_1^{(0)} t_{11}}{d_1} r_j^{(1)}; \; j = 2, \cdots, n \tag{II-24}$$

Eqs. (II-20), (II-23) and (II-24) correspond to Eqs. (38)-(40) in Theorem 6 for k = 1.

The next step is to start with Eq. (II-17) and perform the second Gaussian elimination step. The proof for k = 2 is identical with the above. Extending this step we can state that Eqs. (38)-(40) hold for any k = 1, 2, $\cdots$, n.

As we mentioned before, the underlying assumption is that the given matrix $\underset{\sim}{A}$ can be triangularized, i.e., $a_{kk}^{(k-1)} = \frac{1}{t_{kk}} \neq 0$ for k = 1, 2, $\cdots$, n. Thus t_{kk}, for k = 1, 2, $\cdots$, n must be well defined in order to be able to perform the Gaussian elimination without reordering the rows and columns. In terms of the modified matrix $\underset{\sim}{A}^*$, the corresponding condition is that t_{kk}^*, for k = 1, 2, $\cdots$, n are well defined. Since

$$t_{kk}^* = \frac{d_{k-1}}{d_k} t_{kk}$$

the condition is satisfied if and only if $d_k \neq 0$, for k = 1, 2, $\cdots$, n. This completes the proof of Theorem 6.

References

1. D. W. Martin and G. Peters, "The Application of Newton's Method to Network Analysis by Digital Computer", J. Inst. Water Engineers 17, (1963) pp. 115-129.

2. R. K. Livesley, "The Analysis of Large Structural Systems", The Computer Jl. 3, (1960), pp. 34-39.

3. T. A. Porsching, "Jacobi and Gauss-Seidel Method for Nonlinear Network Problems", SIAM J. Numer. Anal. 6, (1969), pp. 437-449.

4. I. W. Sandberg, "Theorems on the Analysis of Nonlinear Transistor Networks", Bell Syst. Tech. J., vol. 49, Jan. 1970, pp. 95-114.

5. H. Nikaido, Convex Structures and Economic Theory, New York: Academic Press, 1968.

6. R. S. Palais, "Natural Operations on Differential Forms", Trans. Amer. Math. Soc., vol. 92, July 1959, pp. 125-141.

7. C.A. Holzmann and R. W. Liu, "On the Dynamic Equations of Nonlinear Networks with n-Coupled Elements", Proc. 3rd Ann. Allerton Conf. Circuit and System Theory, 1965, pp. 536-545.

8. E. S. Kuh and I. Hajj, "Nonlinear Circuit Theory: Resistive Networks", Proc. IEEE, vol. 59, Mar. 1971, pp. 340-355.

9. T. Fujisawa and E. S. Kuh, "Some Results on Existence and Uniqueness of Solutions of Nonlinear Networks", IEEE Trans., vol. CT-18, No. 5, Sept. 1971, pp. 501-506.

10. T. Fujisawa and E. S. Kuh, "Piecewise Linear Theory of Nonlinear Networks", SIAM J. App. Math., vol. 20, No. 1, Jan. 1972.

11. J. M. Bennett, "Triangular Factors of Modified Matrices", Numer. Math., vol. 7, 1965, pp. 217-221.

12. M. Fielder and V. Pták, "Some Generalizations of Positive Definitions and Monotonicity", Numer. Math., vol. 9, 1966, pp. 163-172.

13. J. Katzenelson, "An Algorithm for Solving Nonlinear Resistive Networks", Bell Syst. Tech. J., vol. 44, Oct. 1965, pp. 1605-1620.

14. L. Chua, "Efficient Computer Algorithm for Piecewise-linear Analysis of Resistive Nonlinear Networks", IEEE Trans., vol. CT-18, No. 1, Jan. 1971, pp. 73-85.

15. T. Ohtsuki and N. Yoshida, "DC Analysis of Nonlinear Networks Based on Generalized Piecewise-linear Characterization", IEEE Trans., vol. CT-18, No. 1, Jan. 1971, pp. 146-152.

16. R. A. Willoughby, "Some Pivot Considerations in Direct Methods for Sparse Matrices", Proc. Mexico International Conf. on Systems, Networks and Computers, Oaxtepex, Mexico, Jan. 1971, pp. 387-390.

17. J. H. Wilkinson, The Algebraic Eigenvalue Problem, Claredon Press, Oxfo-d, England, 1965.

18. F. G. Gustavson, W. Liniger and R. Willoughby, "Symbolic Generation of an Optimal Count Algorithm for Sparse Systems of Linear Equations", J. ACM, vol. 17, No. 1, Jan. 1970, pp. 78-109.

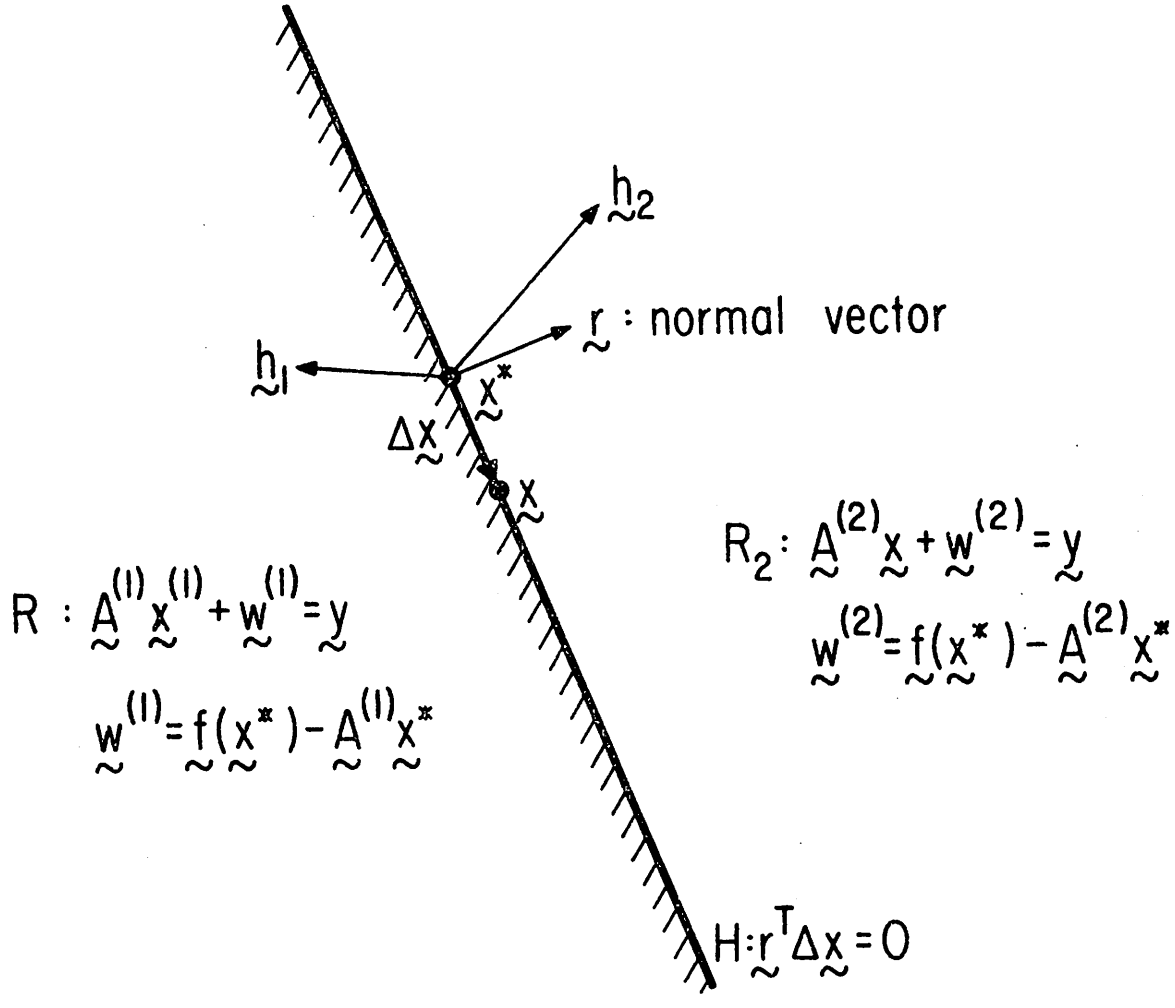


Fig. 1. Boundary hyperplanes between two neighboring regions.

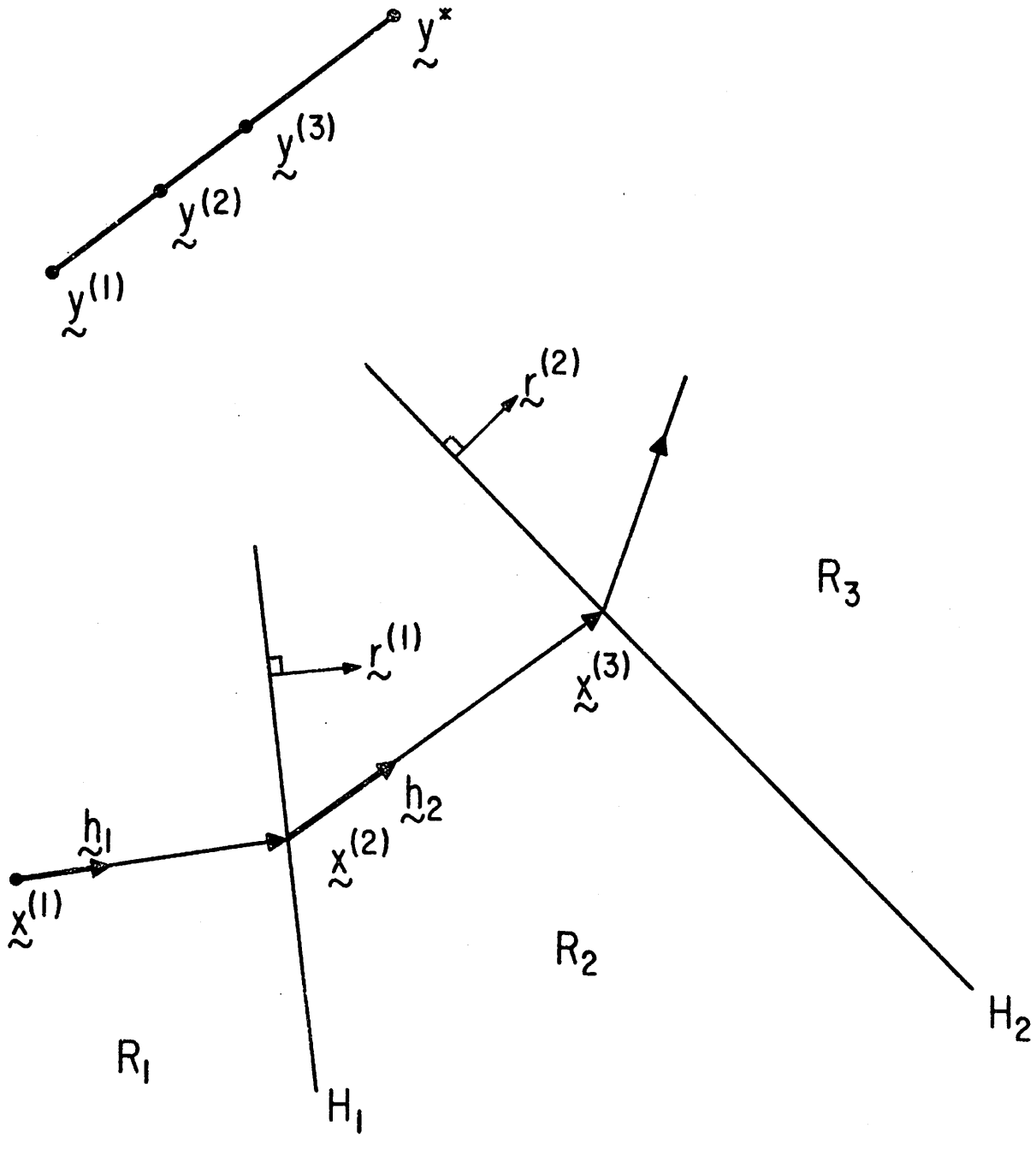


Fig. 2. Construction of the solution curve.

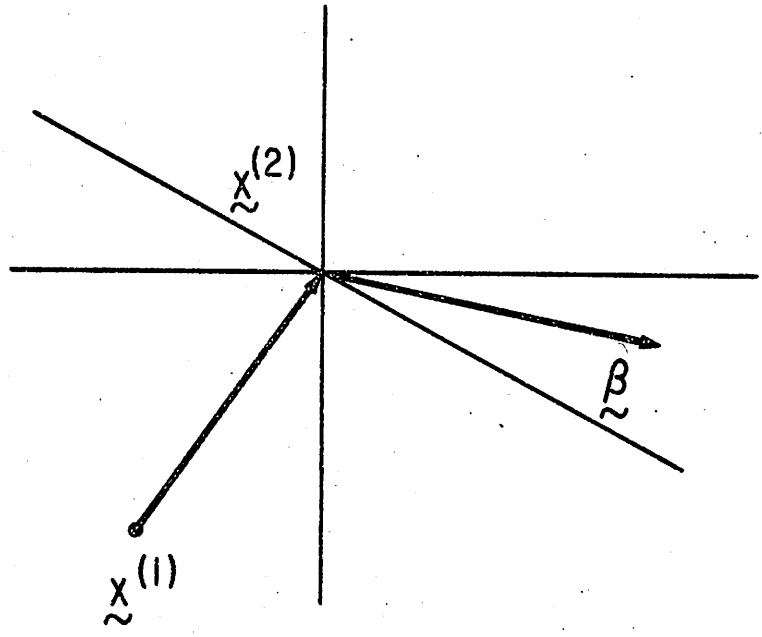


Fig. 3. Solution curve hitting a corner.

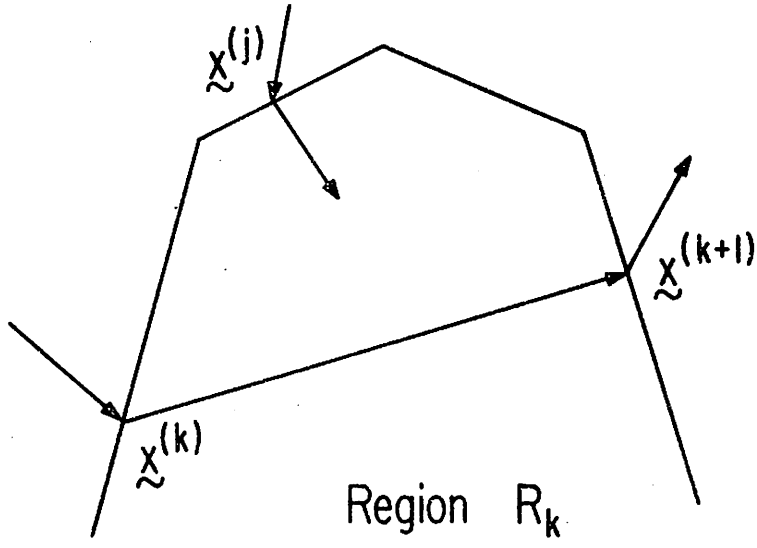


Fig. 4. A solution curve reenters region R_k.

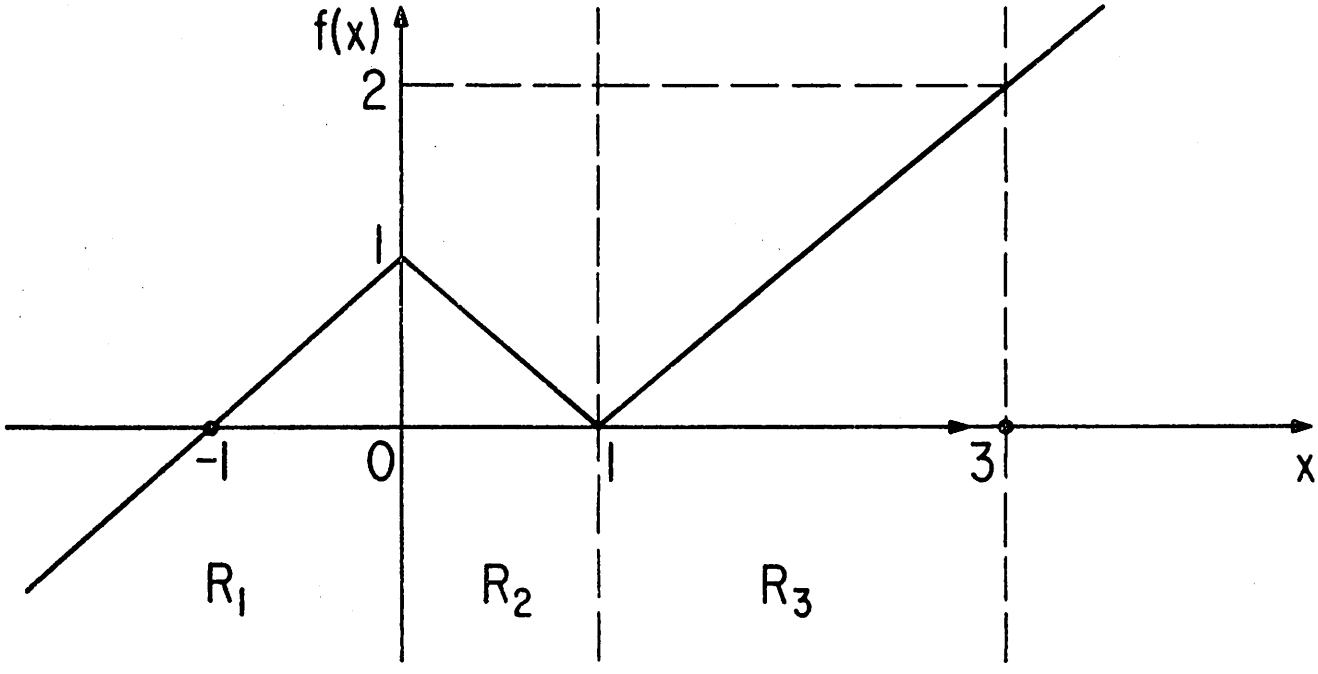


Fig. 5. A continuous piecewise-linear function which has at least one solution for all inputs.

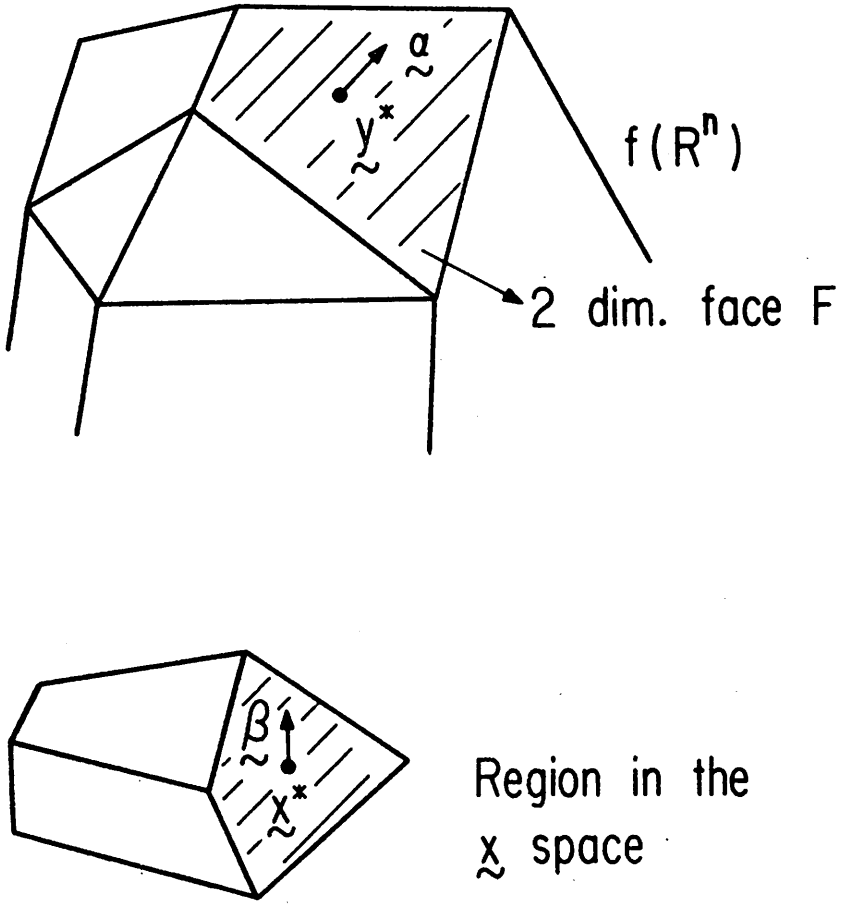


Fig. 6. Multi-dimensional illustration of the range space $\underset{\sim}{f}(R^n)$.