RICE UNIVERSITY

A Linear Graph and State Variable Formulation of The Electrical Network Analysis Problem

by

Gerald L. Barksdale, Jr.

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

ELECTRICAL ENGINEERING

Thesis Director's signature:

Houston, Texas

May 1966
ABSTRACT

A LINEAR GRAPH AND STATE VARIABLE FORMULATION OF
THE ELECTRICAL NETWORK ANALYSIS PROBLEM

Gerald L. Barksdale, Jr.

This paper reports on the development of a network analysis technique employing both a linear graph theory formulation of the topological relations and a state variable analysis of these relations. As a consequence of these investigations, a network analysis program, GENA, was developed. Special points of interest include the development of an algorithm for efficient determination of the fundamental cut-set equations of the network and investigation of restrictions on the state variable formulation for electrical networks.
A LINEAR GRAPH AND STATE VARIABLE FORMULATION OF
THE ELECTRICAL NETWORK ANALYSIS PROBLEM

1. Introduction
   .1 Analysis of Generalized Networks
   .2 An Outline of Program Operation

2. The State Variable Approach to Network Analysis
   .1 Advantages and Limitations of the Formulation
   .2 A State Variable Formulation of the General Electrical
   Network Analysis Problem

3. Special Techniques
   .1 Topological Techniques: An Efficient Algorithm for the
     Determination of the Cut-Set Equations of a Network
   .2 Solution Methods for First-Order Linear Differential
     Equations

4. Open Areas for Research

Appendices

A1. A Notational Glossary
A2. Solution of Systems of Complex Algebraic Equations

Acknowledgments

Bibliography
1.1 INTRODUCTION

The analysis of networks, both electrical and mechanical, has been the object of intensive research during the past half century. At first, analysis was carried out for specific networks with procedures developed to expedite the solution of each specific network. Later, when computer analysis became possible, the rationale still required the analysis of specific classes of networks with specific programs created for the analysis of each specific class of network. The configuration of the networks was fixed by the nature of the programs which were written to analyze them.

It is the developments of the late 1950s and the early part of this decade which laid the foundation for the analysis and program development [2] which are reported in this paper. These developments were twofold: first, the electronic computer became of age as a practical and readily available tool of science and technology; and second, the well developed mathematical field [4] of linear graph theory began to be exploited by engineers [22] to provide a basis for a general approach to the analysis of networks.

The basic concept behind generalized network analysis is as follows: A Network is a structure which may be represented by a system graph which catalogues the interconnections of the elements in the system (and assigns a reference direction for flows, in the case of directed graphs). This system graph completely defines the topology of the problem. There remains only the specification of the relation
of the "through and across variables", as Koenig [11] so quaintly states it. In the electrical engineering context, we will recognize these relations to be those for voltage and current, i.e., \( V = RI \), \( I = CV \), and \( V = LI \).

Given the element and the topological relations, it is possible to mathematically describe the network under consideration, and thus to perform an analysis of the network. The technique is without question impractical without the use of a digital computer, for the manipulation involved in analyzing the simplest of networks is staggering. However, using a medium sized computer, it is possible to analyze a network of considerable size in a matter of seconds. And the effort required of the user is no more than that required to give the computer a list of the components, their values, and their interconnections in terms of the nodes of a graph.

In this paper we shall discuss the salient points of the analysis necessary to develop a network analysis program based on the linear graph formulation of the problem. We shall discuss some of the features of the program, as well as interpret some of the limitations of the linear graph and state variable formulation of the network analysis program.
1.2 AN OUTLINE OF PROGRAM OPERATION -- GENA

The operation of the general electrical network analysis program which has been developed can be broken down into four basic operational divisions: the GENA language compiler; the topological and equation-forming subroutines; the analysis (equation solving) subroutines; and, the output subroutines. In the following paragraphs, we shall discuss the interrelations of each of these areas with the overall analysis problem.

The program, because of its great length, requires formulation as a multiple core load (CHAIN) job for processing. Each of the above divisions represents a separate link in the chain. Intermediate results are stored on the IBM 1301 disk file. The user of the program will need to be concerned only with the first and last links of the chain, since the equation formation and solution will produce no output, except in the case where program errors occur.

The first link, GENIN, is a compiler for the GENA network analysis language. This link processes all input data, analyzes it for errors in grammar and logic, and prints diagnostic information for the user. Any error in the input language causes the deletion of execution. The input format is designed with user convenience in mind. In addition to assimilation of the input data, this link reduces the input connection data to the fundamental cut-set matrix which will be needed in subsequent links.

Link two, ACTR, manipulates the topological and element information derived from GENIN to form a system of first order linear

---

1 GENA is an acronym for General Electrical Network Analyzer.
differential equations and the systems of algebraic equations necessary for the complete solution of the network problem. In the event that a DC analysis is requested, the output from this link consists of the algebraic equations only.

Link three, ACS, is the frequency domain solution program. This link uses the Laplace transforms of the differential equations which were formed in ACTR and solves the complex equations which result when we let \( s = j\omega \). The solution of this set of equations for the state variables gives us all the information which is necessary\(^2\) for the calculation of any voltage or current in the network. After calculation, the desired voltages and currents are stored for later processing by the output link. If the frequency is being varied, the forward and back solutions are recursively executed until the variation list has been satisfied, or a maximum of fifty variations has been made, whichever occurs first. Note that there has been no return to the equation forming link, as would have been required if the impedance or admittance formulation had been used.

The fourth link, TRS, is similar to the preceding link, except that the solution for the state variables is effected by numerical integration rather than through transform techniques. The integration subroutine uses a Runge-Kutta starter and an Adams-Moulton integration scheme thereafter. Up to fifty output points may be stored for each of the desired output quantities.

\(^2\) This concept is discussed in greater detail in Section 2.1.
The last link, GENOUT, controls the output for GENA. The program is designed to give the user maximum flexibility in data output. He has a choice of output devices and of formats which will be used. Through the use of the FUNCTION option, he may obtain output of binary combinations of the primary network variables (voltage ratios, impedances, etc.).

GENA was developed in order to have a flexible network analysis program available for use in modeling studies, as well as a means of analyzing electrical networks in a more efficient manner than that provided by ECAP [3,10]. The program developed can well be considered a success, since there has been a substantial decrease in the running time for AC analyses (in comparison with ECAP) and program flexibility has been greatly increased.
2.1 THE STATE SPACE FORMULATION

The analysis of RLC networks may be considered from a number of viewpoints. First, the classical methods, nodal and loop analysis, find wide application; this is perhaps due to the physical meaning which can readily be attached to the represented quantities, as well as the ease with which the equations can be formed. However, the resultant integro-differential or second order differential equations are at times objectionable. This is especially true if the solution must be achieved by numerical integration.

To circumvent this problem, we shall consider a second formulation, one which is a mixture of the two previous methods: this analysis technique makes use of only first order differential equations; it is known as the "state variable" formulation of the system. In brief, the state variables comprise the minimum amount of information which we must know to predict the behavior of the system at a later time. For a dynamic system, this minimum information corresponds to a knowledge of the energy which is stored in the independent energy storage elements of the system [20].

For the classical dynamic system, we may describe the energy balances by means of a system of first order differential equations. We refer to these equations as the state variables for the system. In matrix notation, we have the relation

\[ \dot{x} = [A] \times x] + f \]

where \( x \) represents the state variables, \( A \) is the system matrix, and \( f \) is the forcing function.
For electrical networks, the independent energy storage elements will be the tree capacitors (voltage independence) and the cotree inductors (current independence). Thus, the tree capacitor voltages and the cotree inductor currents constitute a set of state variables for an electrical network. Once the solution of the state equations is obtained, other variables in the network may be obtained with the aid of the topological relations which we have at hand. This topological solution process is, however, much less of a computational problem than solving a larger system of simultaneous equations.

A second advantage of the state space formulation of the network problem is that the same formulation technique is valid for both time and frequency domain analyses since we have included no assumptions regarding the nature of the solution technique in the formation of the differential equations of the system, except that the coefficients are constant. This independence of the formulation and solution techniques gives additional flexibility which is not available in other analysis programs. Further, there is no need to reform the system of equations each time it is desired to examine the system at a new frequency, as is necessary in either the impedance or admittance formulations.

Computation with the linear graph -- state space formulation of the problem is indeed cumbersome if attempted by hand; the procedure is only practical as a computational method when used in conjunction with a digital computer. In many ways, the linear graph and state variable method of analysis is like a high level computer language:
with ease in use and increased generality, the absolute efficiency of execution of any given class of problems (on a machine level) is reduced, often by as much as a factor of two or three. However, when viewed as a man-machine time measurement, the results are a startling improvement in the total time spent in analysis.

For a single frequency point, the state variable formulation is favorable if the number of reactive elements represents a small percentage of the network elements. In the example shown below we see that the number of simultaneous equations which must be solved under the classical solution methods is greater than that for the state variable approach. If a frequency response is desired, there will be additional savings in computational time through the use of the state variable formulation.

![Diagram](image-url)  

5 loop equations  
4 node equations  
2 state variables  

Figure 1.

1 For further elaboration on this concept, see the DYANA manual introduction [12].
The second example shows an advantage in using the classical analysis methods if no frequency response is desired. Note that for more than one frequency point it is necessary to reform either the loop or node equations at each frequency. The state variable analysis requires only a single formation of the system equations, with the frequency dependence being inserted in the solution loop only. For details of this procedure, see Appendix II.

Figure 2.

4 loop equations
3 node equations
5 state variables
2.2 A STATE VARIABLE FORMULATION OF THE GENERAL ELECTRICAL NETWORK ANALYSIS PROBLEM

In order to standardize the ordering of the elements, we shall establish the following priorities for the placement of components in the tree which will be chosen for the network:

1. All voltage sources must be in the tree.
2. Choose as many capacitors as possible for inclusion in the tree.
3. Continue to fill out the tree with resistors.
4. If the tree is not yet complete, then choose such inductors as are necessary to complete it.
5. Under no circumstance may a current source be placed in the tree.

By following the above procedure, we will always have maximum freedom of specification of initial values for the circuit. It is assumed that we limit ourselves to the specification of capacitor voltages and inductor currents, since tree voltages and cotree currents may be arbitrarily specified. This ordering procedure also guarantees that certain entries in the circuit and cut-set matrices will always be zero. This results from the fact that no inductor or resistor may be included in the tree to the exclusion of a capacitor. Similarly, no resistor would be placed in the cotree to the exclusion of an inductor.

The four basic matrix equations which are used in the analysis are:

\[ \begin{align*}
\mathbf{V}_c + [B] \mathbf{V}_t &= 0 \\
\mathbf{I}_t + [S] \mathbf{I}_c &= 0
\end{align*} \]

Kirchhoff's Voltage Law 1)
Kirchhoff's Current Law 2)
In expanded form, our four basic equations are:

\[
\begin{align*}
V_{CC} &= \begin{bmatrix} B_{11} & B_{12} & B_{13} & 0 & 0 \end{bmatrix} V_{SP} \quad 0 \\
V_{GC} &= \begin{bmatrix} B_{21} & B_{22} & B_{23} & B_{24} & 0 \end{bmatrix} V_{SG} \quad 0 \\
V_{LC} &= \begin{bmatrix} B_{31} & B_{32} & B_{33} & B_{34} & B_{35} \end{bmatrix} V_{CT} \quad = 0 \\
V_{ISG} &= \begin{bmatrix} B_{41} & B_{42} & B_{43} & B_{44} & B_{45} \end{bmatrix} V_{RT} \quad 0 \\
V_{ISP} &= \begin{bmatrix} B_{51} & B_{52} & B_{53} & B_{54} & B_{55} \end{bmatrix} V_{LT} \quad 0
\end{align*}
\]

In equations 1) and 2), the subscripts C and T refer to the cotree and tree, respectively. The matrices B and S are the fundamental circuit and cut-set matrices. The manner in which these equations relate to the incidence matrix is discussed in Section 3.1.

See Appendix I for a discussion of notational conventions.
It is desired to reduce the above equations to a set of first order differential equations in terms of the state variables of the system. This set of equations describes the voltages in the tree capacitors and currents in the cotree inductors, thus allowing the determination of all other voltages and currents in the network.
procedure can always be carried out. However, if the network includes proportional sources, it is possible to obtain a network which cannot be completely described in terms of the classic state variables.

Since most network configurations can be analyzed in terms of the state variables, the network analysis program solves only state variable problems. However, it is necessary to examine each network in order to determine the completeness of the set of differential equations which were formed. This check is easily made by examining each term in particular matrices which are formed during the equation formation procedure. If any of these terms is nonzero, then the set of differential equations includes independent relations for the voltage across tree inductors and/or the current through cotree capacitors.

To understand the reason for this, we note that the absorption of the controlling branch into the \( S \) or \( B \) matrices, or the associated unity matrix, will generate a pair of matrices which we shall designate \( N \) and \( M \) respectively. These new matrices form a set of "pseudo-topological" matrices which describe the interaction of the tree and cotree elements.

When we allow dependent sources, we are faced with the possibility of nonzero terms in those locations of the \( S \) and \( B \) matrices which were previously guaranteed to be zero by our tree picking criterion. Further, certain choices of the driving branch will require an inverse to be taken in the evaluation of \( N \) or \( M \). Should the inverse not exist,
then this is an indication that improper (impossible) source conditions have been declared for the network.

To consider this result in more detail, we shall examine the matrices which are used to generate the \( N \) matrix:

\[
N = -\begin{bmatrix}
1 + S_{35}J_{11} & S_{35}J_{12} & S_{35}J_{13} \\
S_{45}J_{11} & 1 + S_{45}J_{12} & S_{45}J_{13} \\
S_{55}J_{11} & S_{55}J_{12} & 1 + S_{55}J_{13}
\end{bmatrix}^{-1}
\]

\[
\times\begin{bmatrix}
S_{31} + S_{35}J_{21} & S_{32} + S_{35}J_{22} & S_{33} + S_{35}J_{23} & S_{34} \\
S_{45}J_{21} & S_{42} + S_{45}J_{22} & S_{43} + S_{45}J_{23} & S_{44} \\
S_{55}J_{21} & S_{55}J_{22} & S_{53} + S_{55}J_{23} & S_{54}
\end{bmatrix}
\]

We are able to catalog the following facts concerning each entry in the above matrices which includes a \( J_{ik} \) term.

1. If a particular \( J_{ik} \) is nonzero, then all other \( J \)'s which multiply the same \( S \) term will be zero valued.

2. As a consequence of (1), each dependent current source will produce one and only one nonzero \( SJ \) product.

3. The inverse will exist except when the \( SJ \) term on the diagonal is equal to \(-1.0\). If this is the case, we have specified an inconsistent set of conditions on the network.

These properties help us in the cataloging of the interdependences which preclude a state variable formulation. Note that there is an
analogous set of relations for the $M$ matrix; these indicate the
dependent voltage source contributions to that matrix.

As is noted by Kuh and Rohrer [16], the state variables are
sufficient for the solution of the network if we restrict the con¬
trolling branches for current-controlled current sources to the
cotree and the output branches to the tree; for a voltage-controlled
voltage source we similarly restrict the input to the tree and the
output to the cotree of the network. Although these conditions are
sufficient for the writing of the state equations for a network, they
are by no means necessary.

To be able to write the state equations for the network, we must
be able to express all voltages and currents in the network as functions
of the tree capacitor voltages and the cotree inductor currents. To
do this, we must not have an explicit appearance of the tree inductor
voltages or the cotree capacitor currents in the system of differential
equations which we generate. This condition may be achieved if the
matrix which premultiplies the $\dot{V}_{LT}$ and $\dot{I}_{CC}$ vector is null. The
occurrence of a nonzero entry in this matrix may be caused if the
following network dependencies exist:

1. tree inductors and cotree capacitors which are dependent
   upon each other;
2. tree inductor currents which are dependent upon cotree
   conductance currents;
3. cotree capacitor voltages which are dependent on tree
   resistor voltages;
(4) tree resistor currents which are dependent upon currents through cotree inductances or conductances;

(5) cotree conductance voltages which are dependent upon resistor or inductor voltages in the tree.

It should be stressed that these conditions may cause nonzero entries in our final matrix. If the particular combination of operations performed to form the final matrix is such that each of the offending conditions (2) through (5) is fortuitously multiplied by a zero valued term, we are still able to form the normal state equations for the system. In general, any dependence which introduces nonzero terms into the N and M matrices in those positions which formerly corresponded to invalid topological dependencies (guaranteed zero entries) will require the use of additional differential equations.

Networks which cannot be solved by the normal state formulation are generally unstable in nature, or contain articulation points. The role of articulation points is to make possible element assignments which would not occur if the graph were nonseparable, thus allowing conditions (1) through (5) to be achieved.

If we have been able to form the state equations for the network we will have the following system of equations to solve:

\[
\begin{align*}
\dot{V}_{\text{CT}} & = \begin{bmatrix} A \end{bmatrix} V_{\text{CT}} + \begin{bmatrix} B \end{bmatrix} V_{SG} + \begin{bmatrix} C \end{bmatrix} I_{SG} \\
\dot{I}_{\text{LC}} & = \begin{bmatrix} A \end{bmatrix} I_{\text{LC}} + \begin{bmatrix} B \end{bmatrix} I_{SG} + \begin{bmatrix} C \end{bmatrix} I_{SG}
\end{align*}
\]

This system of equations may then be solved by any of the techniques discussed in Section 3.2

By "normal" we mean those differential equations which involve only $V_{CT}$ and $I_{LC}$. 
3.1 AN EFFICIENT COMPUTATIONAL TECHNIQUE FOR DETERMINATION OF THE
CUT-SET EQUATIONS OF A NETWORK

Introduction

This section presents an efficient algorithm for the simultaneous determination of the fundamental cut-set equations and the inverse of the tree portion of the incidence matrix of a connected graph. Both matrices are determined directly from the incidence matrix of the network through the use of modulo-3 algebra. A discussion of the computational techniques and pertinent background material is included.

If we discuss non-oriented graphs, the relation of the cut-set matrix to the incidence matrix can be seen clearly through the use of modulo-2 algebra [26]. We can obtain the fundamental cut-set equations from the incidence matrix by performing row and column operations (modulo-2 addition and multiplication only) on the \( N \) independent rows of the incidence matrix, where \( N \) is the rank of the matrix.

In this paper we will make use of the properties of the residue class modulo-3 which allow similar relation of the incidence and fundamental cut-set matrices for directed flow graphs.

Computational Procedure

In order to determine the fundamental cut-set equations from the incidence matrix for a directed graph, the user must prepare the data in such a way that the first \( N \) columns of the incidence matrix corres-

---

1 This section, in slightly modified form, has been accepted for publication as a Correspondence item in the IEEE Transactions on Circuit Theory, Vol. CT-13-3, September, 1966, Paper #GCT 2097(2).
pond to a tree in the network to be analyzed. This is a relatively easy task if a tree-sorting algorithm such as that proposed by Hale [14] is used. The present algorithm will allow the user to find the fundamental cut-set equations for any tree of the connected graph.

Note that the restriction on the ordering of the first \( N \) columns is not a mathematical restriction on the network, since reordering of the columns is allowed before the reduction procedure is begun. The physical reason for not allowing column operations other than exchange is simply that any addition of columns would destroy branch identities within the incidence matrix. In the present application, the ordering criterion used for the preliminary list requires all voltage sources to be in the tree and all current sources to be in the cotree. All other elements are placed in the tree with the following priorities: capacitors first, resistors second, and inductors last. The network elements, arranged in the desired order are then checked to determine the most desirable tree configuration. If any rearrangement is necessary, those elements which cannot be tree branches are moved to leading positions in the cotree. The initial ordering of the network guarantees that the least desirable elements will always be those which are shifted from the tree to the cotree. It is of interest to note that the reduction procedure fails if one of the first \( N \) columns of the incidence matrix is generated by a chord rather than a tree branch.

In computing the fundamental cut-set matrix, we first delete one row of the incidence matrix (the datum node row). We then convert this reduced matrix to our new number base, with the following correspondences:
The reduction procedure for computing the one-tree-branch cut-sets from the incidence matrix with modulo-3 algebra is as follows:

i) Examine the first diagonal element of the square matrix in the leading N positions of the reduced incidence matrix, A.

ii) If the diagonal element is non-zero, proceed with the reduction; otherwise, search the column below the diagonal until the first non-zero entry is found. Add this row to the diagonal row.²

iii) Reduce all off-diagonal terms to zero by successive addition of the diagonal row. Note that the maximum number of additions of the diagonal row required to reduce any term is two.

iv) Examine the next diagonal term and repeat the above steps.

The resultant diagonal will consist of both 1's and 2's. The final reduction to a unity matrix is simple, since any "2" term of the diagonal can be changed to a "1" by adding the row to itself. As a result of the reduction procedure, we have replaced the reduced incidence matrix A by the fundamental cut-set matrix of the tree consisting of the first N branches of the incidence matrix.

In a practical implementation of this procedure on a digital computer, where modulo-3 operations can be obtained only as the result of a remaindering process, the resultant sums are carried as decimal sums until

² If the diagonal element and all terms below it are zero, then the column is not a tree branch, and the reduction fails. This property may be used in place of Hale's method to determine the tree of the network. However, the computation required is approximately the same.
the modulo-3 properties are needed in the reduction procedure. We are able to do this because of the congruence relation between the two number bases.

In order to determine the inverse of the tree portion of the incidence matrix, it is necessary to append a unity matrix to the reduced incidence matrix. By extending the row operations to this matrix, the diagonalization of the leading N rows of the incidence matrix results in the formation of their inverse in the appended matrix. Thus, we have an efficient means of calculating the node-to-datum-path matrix described by Branin [8]. The method has an advantage over conventional methods in that the only operation necessary for reduction is addition; further, there is no round-off error. The reader will note that the reduction procedures described in this paper are readily (and efficiently)\(^3\) coded in FORTRAN.

Before reverting to a packed machine word reduction procedure [27], the computational cost of packing and unpacking the data in order to obtain a form which is suitable for carrying out further arithmetic operations should be carefully weighed. It should be noted that the procedure reported by So does not provide the user with a direct method for the calculation of the circuit or cut-set matrices, whereas the present procedure directly generates the fundamental cut-set matrix from the incidence matrix. The reader should note the logical connection

\(^3\) For a network of 20 nodes and 100 branches, the incidence matrix was formed, the proper tree was found, and the fundamental cut-set matrix was obtained in less than 2 seconds on an IBM 7040.
which the use of modulo arithmetic provides between the incidence matrix and its derived cut-set matrix. The beauty of the present direct method for calculation of the fundamental cut-set matrix is that the calculation of the tree inverse, per se, is unnecessary. Should extreme network size require storage conservation, a logical reduction method similar to that employed by So [27] would prove valuable as a means of implementing the proposed direct reduction procedure.
3.2 METHODS OF SOLUTION OF FIRST-ORDER LINEAR DIFFERENTIAL EQUATIONS: A DISCUSSION OF TECHNIQUES.

The method of solution of the differential equations which have been formed has been the object of considerable discussion. Three basic methods were considered, of which two were chosen as being practical. We shall briefly discuss each of these methods in the following section.

The first method which we considered for the solution of the AC problem was a scheme which generated an inverse of the system of differential equations in terms of the transform variables. At first this method seemed quite powerful, since there would be no need for reinversion of the system equations for different frequencies, as the solution at any frequency could be readily evaluated by replacing the transform variable by the appropriate value of \( j\omega \). This procedure was developed [1] and then implemented on the IBM 7040. Several disadvantages of the method were immediately evident: i). the solution times were extremely long; ii). numerical accuracy, even when double precision operations were used, was inadequate for practical use as an analysis tool; and iii). storage requirements were excessive. Thus, polynomial inversion was discarded as a reasonable means of finding solutions to AC problems.

The second method which was developed involved the solution of a set of complex simultaneous equations. A special algorithm was developed to effect this efficiently. (See Appendix II.) The reduction of the problem from one of inversion to that of solution of a set of...
simultaneous equations reduced the number of calculations by a factor of \( N \), the dimension of the problem. Further, an extremely efficient simultaneous equations solution routine, LED50 [5] was available.

This method of solution has proved to be quite accurate, extremely rapid, and of reasonable size in terms of storage requirements. It is the method which is presently used to implement the solution of the AC problem.

Transient analysis has at present received little attention, since the major programming effort has been directed toward the development of the frequency domain solution algorithm. The transient solution, except for the actual solution of the state variables, will be essentially the same as the AC solution, i.e., the back substitution and variable extraction will be done in the same manner. The solution of the state variable equations will be initially implemented with NIODES, a subroutine for the solution of differential equations which uses a Runge-Kutta starter and Adams-Moulton techniques for solution.
4.1 OPEN AREAS FOR FURTHER RESEARCH

First, the various numerical solution techniques which are used should be evaluated for accuracy and speed. Such things as error check columns in the inversion and simultaneous equation subroutines should be investigated. Further, there is room for improvement in the numerical integration routine.

The analysis program should be generalized in order to be able to solve nonlinear problems. Special techniques may have to be developed in order to do this efficiently. In addition, the inclusion of "black boxes" in the analysis capability of the program will allow the analysis of extensive networks.

The equation forming techniques which have been developed may be readily applied to problems in other disciplines. The requirements for the analysis technique are that a simple triad relation (such as Ohm's law) exist between the through and across variables. If the proper relations can be found, it is possible to use the equation formulation and the analysis for studies in such areas as economic modeling, production planning, biological modeling and traffic planning.
Al. A NOTATIONAL GLOSSARY

Extensive systems of equations require a notational convention which, though perhaps cumbersome at first, will allow concise expression of the functional relations of the analysis to be presented. The following table gives the basic notational scheme which is used in this thesis.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B_{ij}$</td>
<td>The element in the $i$th row and $j$th column of the fundamental circuit matrix.</td>
</tr>
<tr>
<td>$S_{ij}$</td>
<td>The $i,j$ element of the fundamental cut-set matrix.</td>
</tr>
<tr>
<td>$D_{CT}$</td>
<td>An element matrix (diagonal) which consists of the values of the capacitors (C) which are in the tree (T) of the network.</td>
</tr>
<tr>
<td>$V_{CC}$</td>
<td>The voltage across a capacitor (C) in the cotree (C).</td>
</tr>
<tr>
<td>$I_{VSG}$</td>
<td>The current through a given (independent voltage source ($S_{V}$)) in the tree. Note that the $T$ is omitted since a voltage source must, by definition, be in the tree.</td>
</tr>
<tr>
<td>$V_{SP}$</td>
<td>The value of a proportional (P) voltage source ($S_{P}$).</td>
</tr>
<tr>
<td>$I_{SG}$</td>
<td>The current through a given (G) [specified independent] current source ($S_{G}$).</td>
</tr>
</tbody>
</table>

Symbols beginning with $I$ denote current and symbols starting with $V$ denote voltage. The first subscript denotes the element type (R,L,G,C) and the second denotes the position of the element in the network (tree or cotree). Note that the second subscript has not been specified in the case of voltage and current sources since

---

1 The standard notation for the fundamental cut-set or circuit matrix includes the unity matrices as a part of $S$ and $B$. For the sake of brevity and clarity we have departed from this convention, assuming $S$ and $B$ to exclude the unity matrix.
their requisite independence requires their placement in the tree and cotree, respectively.

Components which may be used in the network include resistors, capacitors, and inductors (no mutual inductance, however), voltage controlled voltage sources, current controlled current sources, and ideal voltage and current sources.
Let us consider the system of transformed differential equations


(1)

where \( A, B, C \) are matrices with real coefficients and \( X \) and \( F \) are vectors of Laplace transforms of the response function and the forcing function respectively. Substituting \( j\omega \) for \( s \) and separating our system of equations into a real and an imaginary system, we have

\[
\begin{align*}
\text{(Real)} & \quad -[A]X_1 - \omega[I]X_2 = [C]F_1 - \omega[B]F_2 \\
\text{(Imag)} & \quad \omega[I]X_1 - [A]X_2 = \omega[B]F_1 + [C]F_2
\end{align*}
\]

(2)

where the subscripts 1 and 2 refer to the real and imaginary parts of the original function, respectively.

From (2b) we see that

\[ X_1 = \frac{1}{\omega} [A] X_2 + [B]F_1 + \frac{1}{\omega} [C]F_2 \]  

(3)

Substituting (3) into (2a) and multiplying through by \( \omega \), we have,

\[
\begin{align*}
\left( [A]^2 + \omega^2 [I] \right) X_2 &= -\omega \left( [A][B] + [C] \right) F_1 \\
&\quad - \left( [A][C]F_2 + \omega^2 [B]F_2 \right)
\end{align*}
\]

(4)

After carrying out the indicated multiplications on the right hand side of equation (4), we may solve for \( X_2 \). Note that we may perform all matrix multiplications before including the \( \omega \) factor, thus requiring updating of vectors rather than systems of matrices.
The solution of the system of equations is carried out by LED50 [5], a general purpose subroutine for solution of systems of equations.

We may now solve for $X_1$ by substitution of $X_2$ into equation (2a). We then have

$$X_1 = \frac{1}{\omega} [A]X_2 + [B]F_1 + \frac{1}{\omega} [C] F_2.$$  \hspace{1cm} (5)

This completes the solution of our set of simultaneous equations.

Note that we have found the solution to $2N$ simultaneous equations by solving only $N$ equations simultaneously.
ACKNOWLEDGEMENTS

Many have helped: parents, teachers, friends -- all deserve my most sincere thanks for their respective roles in the development of the one who now writes. But it is to my co-laborer and fellow student, Gábor I. Ugron, that I must express my deepest thanks; without his help, this thesis might still be unwritten.

Special thanks must also be given Dr. J. V. Leeds, my adviser in this project, for his active interest throughout its development. Thanks are also due Drs. C. S. Burrus and L. E. Davis for their helpful criticism of this thesis and R. S. Eanes for his programming of the GENA language compiler.

This research was supported in part by National Science Foundation grants GK-262 and GU-1153.
BIBLIOGRAPHY


5. P. M. Blair, "Solution of Linear Simultaneous Equations," Program Description for PR LED50; October 2, 1964.


18. J. V. Leeds, "Linear Graph Formulation - Controlled Sources", Unpublished Notes, Rice University, Houston, Texas, 1965.


