S. R. Sedore

SCEPTRE: A Program for Automatic Network Analysis*

Abstract: This paper describes the mathematical formulation of a computer program for automatic transient analysis of electronic networks. The formulation is based on the "state-variable" approach to network analysis and differs from other such programs primarily in the way that the network equations are manipulated to produce a solution. SCEPTRE includes a number of features aimed at providing greater flexibility and convenience for users of the program. Important among these features is that no prescribed equivalent circuit for active elements is required for program operation. Also, linearly dependent voltage and current sources in a network can be handled by the program, and provision has been made to allow a free-form format for input data. The paper includes a discussion of the program's ability to solve networks containing time-varying passive elements, and considers the factors that influence program running time.

Introduction

The digital computer has been used for some time to aid in the solution of electronic circuit problems. A procedure required of analysts who use some of these programs is that they write loop or node equations to describe the network and then translate them into a form that can be accommodated by the computer. In practice this procedure often is carried out by two people: the engineer who writes the equations and the programmer who puts them into computer useable form. Even in the cases where one individual performs both tasks, the process is error prone and time consuming. The desirability of eliminating these disadvantages has led to the development of automatic circuit analysis programs in which the user does not write circuit equations or do any programming. In such programs the network is described topologically and quantitatively according to an easily learned format. Network solution is carried out by the program, which effectively "writes" the appropriate differential and algebraic equations. This equation writing ability is inherent in the program and is based on a mathematical formulation that provides a solution to the general transient analysis program.

Kuo[1] has recently published a survey that includes an extensive bibliography of computer programs currently available for such problems. The most well known of these programs are ECAP and NET-1. ECAP offers dc, ac, and transient solutions and provides the option of worst case and sensitivity analysis with all dc solutions. Its most serious drawback is that the particular transistor equivalent circuit required for use in the program limits the processing of nonlinear networks.[2,3] NET-1, on the other hand, offers dc and transient analysis capability and is not hampered by transistor equivalent circuit nonlinearities.[4] Unfortunately, this program requires the use of a specific topology that is provided in the stored diode and transistor model. No provision is made for the entry of nonlinear elements that are not part of these two stored models. Another program, AEDNET,[19] has been designed to handle a greater range of nonlinearities than do the other programs but has not been widely disseminated at this time.

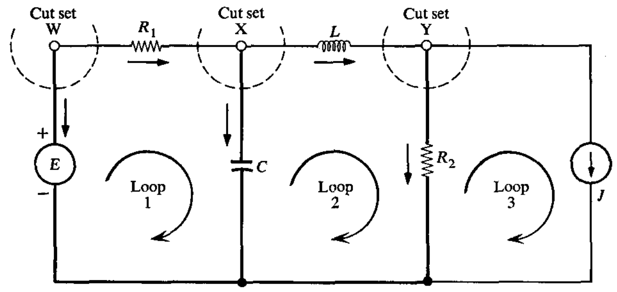
PREDICT,[5] the precursor to the program discussed in this paper, was intended primarily for use in transient analysis problems. Its mathematical formulation, i.e., the state-variable approach,[6–12] was essentially the same as that used in ECAP and NET-1. But it had the important advantage that the analyst was not restricted to using a particular transistor model. Its successor, SCEPTRE,[13,14] also uses the state-variable approach and has considerably less restriction on acceptable network topologies than do the previous programs. While the network equations generated in the SCEPTRE program are the same as in other programs of this type, the manner in which the equations are processed is different. The new procedure leads to a significant reduction in computer solution time for many problems. The improvement is brought about in part by incorporating voltage sources directly into the network "tree" (to be defined later) and forming and processing a larger topological matrix than is usually done.

* Work supported by the Air Force Weapons Laboratory under contract AF 29(601)-6852. Dissemination of the SCEPTRE program is controlled by the Air Force Weapons Laboratory, Attn: WLRET (Capt. Gary Pritchard), Kirtland Air Force Base, New Mexico 87117.

The important new features of SCEPTRE that resulted from experience with PREDICT are summarized as follows:

1. Any active element or interconnected group of elements that can be described (subject to some restrictions) as a combination of voltage or current sources, passive elements, and mutual inductance may be stored on tape by the user and called into use at any point in a network.
2. The user has the option of using a special portion of a program to determine the dc steady-state response of a network. These results may then be automatically inserted as initial conditions of the transient section of the program, or they may be accepted as output of the program. The program can operate in the dc mode, the transient mode, or in a combination of the two.
3. Variations of a basic problem can be studied if the user supplies the changes that will apply for each repeated run.
4. A special section of the program enables the user to define output parameters other than the usual source voltages and currents or passive-element voltages and currents. Systems of linear or nonlinear first-order differential equations that may or may not describe electrical networks may be entered into this section of the program.
5. Voltage and current sources that are linearly dependent on resistor voltages and currents, respectively, can be accommodated without computational delay. This feature makes feasible the extensive use of a family of small signal transistor equivalent circuits.
6. FORTRAN subroutines may be inserted into otherwise conventional SCEPTRE runs. This option provides extension of the scope of the program.
7. A new approach to solution (mentioned earlier) has been employed to assure more rapid solution and, therefore, a smaller amount of computer solution time for most practical problems.
8. Runs may be automatically terminated contingent upon the behavior of specified network quantities.
9. Provision has been made for a free-form format for input data.

This paper has been written with the aim of providing circuit designers an insight into the mathematical processes by which SCEPTRE reaches a problem solution. Hence, mathematicians will find much that is familiar in the following sections. The paper reviews the state-variable approach to analysis, shows the particular way this approach is used in SCEPTRE, and describes the information that must be supplied by the designer. It also analyzes the program's ability to solve networks containing time-varying passive elements and discusses the factors that influence program running time.



Loop 1: E, R_1, C	Cut set W: E, R_1
Loop 2: C, L, R_2	Cut set X: R_1, C, L
Loop 3: R_2, J	Cut set Y: L, R_2, J

Figure 1 Sample network showing elements contained in loops and cut sets and identifying network tree.

The state-variable approach

Consider the network shown in Fig. 1. Assume that the values of source voltage, E, and source current, J, are known functions of time. Also assume that the capacitor voltage V_C and inductor current I_L are specified initial conditions; i.e., their values are known at time t_0. It is easy to see that all other voltages and currents for the network elements can be determined at t_0 by solving, in the order given, the following equations:

$$I_{R1} = (E - V_C)/R_1 \quad \text{(a)}$$

$$V_{R1} = R_1 I_{R1} \quad \text{(b)}$$

$$I_{R2} = I_L - J \quad \text{(c)}$$

$$V_{R2} = R_2 I_{R2} \quad \text{(d)}$$

$$I_C = I_{R1} - I_L \quad \text{(e)}$$

$$V_L = V_C - V_{R2}\,. \quad \text{(f)}$$

To compute the transient values of the network voltages and currents, the next step is to evaluate the equations (a)–(f) at some later instant of time, t_1. It is possible to do this if new "initial conditions" for the interval starting at t_1 can be obtained. Hence, updated values for V_C and I_L are reached by computing their time derivatives at t_0 and then integrating them over the period from t_0 to t_1. After Equations (a)–(f) have been solved at t_0, the derivatives can readily be calculated from

$$\dot{V}_C = I_C/C \quad \text{(g)}$$

$$\dot{I}_L = V_L/L\,. \quad \text{(h)}$$

The process of solving Equations (a)–(h) and integrating (g) and (h) to get updated initial conditions can be repeated as many times as is necessary to achieve a complete transient solution of the network.

Digital computer analysis of a network composed of many voltage and current sources, resistors, capacitors, inductors, and mutual inductances proceeds essentially in the way just described. The generalized version of this procedure is called the state-variable approach to network analysis. The state variables of any system may be defined as the minimum set of variables that, together with all source inputs (and under certain conditions, the time derivatives of inputs), is sufficient to determine all other system variables at a given time. In general, the state variables are the set of *independent* initial conditions of the network.[12] In particular, they are the set of all capacitor voltages diminished by a subset equal in number to the number of voltage-source-capacitor loops in the network, and the set of all inductance currents diminished by a subset equal in number to the number of current-source-inductance cut-sets in the network. (All capacitors in such a loop cannot have independent voltages and all inductors in such a cut-set cannot have independent currents.)

Computer programs that use the state-variable approach precede the solution of network equations by automatically forming the vectors and matrices involved in the equations. In SCEPTRE the voltage and current vectors are formed by construction of a specific network "tree." A tree is defined as a connected subgraph of the network which includes every node of that network, but contains no closed loops. All elements of the network that are included in the tree are called *tree branches* and all elements excluded from the tree are called *links.* The specific tree formed by SCEPTRE contains all voltage sources, a maximum number of capacitors, a minimum number of inductors, and no current sources. The program is capable of handling disconnected networks since it can form multiple trees.

The procedure for identifying the specific tree of a given network is as follows: Using the coded description of the network supplied by the user, SCEPTRE compiles the names of all network elements into a list such that all voltage sources come first, then capacitors, resistors, inductors, and current sources, in that order. The numbers assigned by the user to the nodes of the first voltage source in the list are compared with the numbers assigned to the terminal nodes of all other elements. Any element which has the same node numbers as the voltage source must be a parallel element; it is immediately removed from the list and categorized as a link. Any element which has a node coincident with the TO node of the first voltage source will have that node number changed to the number of the FROM node of the source. The first voltage source is then established as a tree branch.

At this point the process is repeated with the nodes of the second voltage source in the list serving as the reference pair. Eventually, every element that serves as a reference is classified as a tree branch, and every element that is to become a link is removed from the list. If the network topology has forced the program to classify any voltage source as a link or any current source as a tree branch, the run will be aborted and an appropriate diagnostic message will be supplied to the user. It is important to note that this procedure does not require that networks submitted for analysis be free of capacitor loops or inductor cut-sets. Further, any source dependencies that may exist seldom dictate the position that the dependent source may occupy in the network. For example, a voltage-dependent current source may be inserted at almost any position that can accommodate an independent current source. (The restrictions are discussed in a later section.)

Once an element has been classified as a tree branch or link by the procedure above, it is labeled according to the following scheme:

Class 1—Capacitor Links
Class 2—Resistor Links
Class 3—Inductor Links
Class 4—Capacitor Tree Branches
Class 5—Resistor Tree Branches
Class 6—Inductor Tree Branches
Class 7—Voltage Source Tree Branches
Class 8—Current Source Links.

These class numbers will appear as subscripts on the voltage and current vectors that are used in the detailed description of the mathematical formulation. The state variables defined previously are now identified as the vector of capacitor tree branch voltages, V_4, and the vector of inductor link currents, I_3.

Referring back to the simple example, Equations (a) and (b) correspond in the general case to the solution for resistor link currents and voltages, I_2 and V_2, respectively. Equations (c) and (d) correspond to resistor tree branch currents and voltages, I_5 and V_5; Equation (e), to capacitor tree branch currents, I_4; and Equation (f), to inductor link voltages, V_3. Since the circuit contained no capacitor loops or inductor cut sets, it was not necessary to solve any equations corresponding to I_1, V_1, or I_6, V_6.

Matrices of network element values are also formed by the program for use in solving the network equations. The diagonal matrices R_{22} and R_{55} contain the values of link and tree branch resistors, respectively; the diagonal matrices S_{11} and S_{44} are the inverses of the matrices C_{11} and C_{44} which contain the values of link and tree branch capacitors, respectively. Symmetric matrix L_{33} contains values of inductor links and mutual inductance between inductor links, and L_{66} contains values of inductor tree branches and inductances between tree branch inductors. The nonsymmetric matrix $L_{36} = L_{63}{}^T$ contains values of mutual inductance between inductor links and inductor tree branches.

Figure 2 summarizes the transient solution process in terms of the variables used in SCEPTRE. The process begins with the insertion of the independent initial conditions $V_4(t_0)$ and $I_3(t_0)$. All other current and voltage vectors, and the derivatives $\dot{V}_4$ and $\dot{I}_3$ are calculated at t_0. Then $\dot{V}_4$ and $\dot{I}_3$ are integrated to provide $V_4(t_1)$ and $I_3(t_1)$ and the process

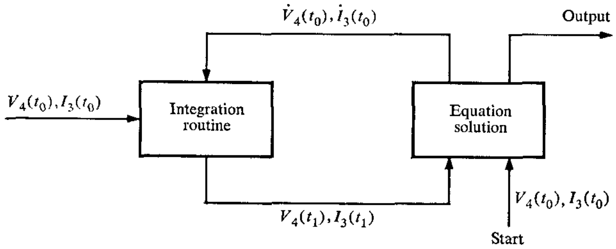


Figure 2 Block diagram of computer solution procedure.

is repeated for as many time steps as are needed to reach the desired termination time. An awareness of these major operations in the solution process should help the program user anticipate the approximate amount of computer time that will be required for his particular problem.

The numerical integration routine controls the size of the solution increments that can be taken, and therefore, the number of solution increments that are required to reach the specified termination. SCEPTRE uses a variable-step integration method for this purpose.[13,15] The efficiency with which the equation solution operations are carried out controls the amount of computer time required per solution increment.

Topological description of network[9–12]

This section shows how the F-matrix, a topological description of the interconnections between tree branch and link elements, is used in the computation of network currents and voltages. The F-matrix and its transpose are submatrices of the "fundamental circuit matrix," T, and the "fundamental cut-set matrix," Q. For a general connected network that includes b elements and n nodes, the matrix T contains b columns and $b - n + 1$ rows. It is formed by comparing the assumed direction of each fundamental circuit loop with the reference direction assigned to each element in the loop. The matrix element values are determined by the rule:

$$t_{ij} = \begin{cases} +1 \text{ if the direction of loop } i \text{ coincides with the reference direction of element } j \text{ in loop } i, \\ -1 \text{ if the direction of loop } i \text{ is opposite to the reference direction of element } j \text{ in loop } i, \\ 0 \text{ if element } j \text{ is not in loop } i. \end{cases}$$

By arranging appropriately the columns that correspond to the link elements, T can be partitioned as

$$T = [UF],$$

where the columns of the $(b - n + 1) \times (b - n + 1)$ unit matrix, U, correspond to the link elements, and the columns of the $(b - n + 1) \times (n - 1)$ matrix F correspond to tree branch elements.

If V represents the vector of all element voltages, the matrix equation for the Kirchhoff voltage law is $TV = 0$. Partitioning V leads to an expression for link voltages, V_L, as a function of the F matrix and tree branch voltages, V_T. That is,

$$TV = [UF]\begin{bmatrix} V_L \\ V_T \end{bmatrix} = 0\,,$$

$$V_L = -FV_T\,. \tag{1}$$

The fundamental cut-set matrix, Q, contains b columns and $n - 1$ rows. It is formed by observing whether the reference direction of each network element in a fundamental cut-set is toward or away from the node. Matrix element values are given by:

$$q_{ij} = \begin{cases} +1 \text{ if the direction of cut-set } i \text{ coincides with the reference direction of element } j \text{ in cut-set } i, \\ -1 \text{ if the direction of cut-set } i \text{ is opposite to the reference direction of element } j \text{ in cut-set } i, \\ 0 \text{ if element } j \text{ is not in cut-set } i. \end{cases}$$

Arranging the columns of Q in the same order as was done for T permits the partition

$$Q = -F^T U\,,$$

where F^T is the transpose of the previously defined matrix, F, and U is an $(n - 1) \times (n - 1)$ unit matrix. Here, the columns of F^T correspond to network link elements and the columns of U correspond to tree branch elements. The matrix equation for the Kirchhoff current law is $QI = 0$. Partitioning the current vector I leads to an expression for tree branch currents I_T in terms of F^T and the link currents I_L. That is,

$$QI = [-F^T U]\begin{bmatrix} I_L \\ I_T \end{bmatrix} = 0\,,$$

$$I_T = F^T I_L\,. \tag{2}$$

Figure 3 shows the partitioned T and Q matrices for the network of Figure 1. Equations (1) and (2) have eliminated the T and Q matrices, and all subsequent derivations are framed in terms of the F matrix.

Figure 3 (a) Incidence matrix Q for network of Figure 1; (b) Incidence matrix T for same network.

$$\begin{array}{ccc} & & \text{Elements} \\ & & \begin{array}{cccccc} R_1 & L & J & C & R_2 & E \end{array} \\ \text{Cut sets} & \begin{array}{c} X \\ Y \\ W \end{array} & \left[\begin{array}{ccc|ccc} -1 & 1 & 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \\ & & \left[\begin{array}{c|c} -F^T & U \end{array}\right] \\ & & (a) \end{array}$$

$$\begin{array}{ccc} \text{Elements} & & \\ \begin{array}{cccccc} R_1 & L & J & C & R_2 & E \end{array} & & \\ \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 0 & -1 \\ 0 & 1 & 0 & -1 & 1 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 \end{array}\right] & \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \text{Loops} \\ \left[\begin{array}{c|c} U & F \end{array}\right] & & \\ (b) & & \end{array}$$

Further partitioning of the F matrix gives the submatrices that are actually formed by SCEPTRE for use in obtaining network solutions:

$$F = \begin{bmatrix} F_{14} & 0 & 0 & F_{17} \\ F_{24} & F_{25} & 0 & F_{27} \\ F_{34} & F_{35} & F_{36} & F_{37} \\ F_{84} & F_{85} & F_{86} & F_{87} \end{bmatrix},$$

where the subscripts indicate that submatrix F_{rs} describes the interconnections between the network elements in Classes r and s. The zero submatrices can be explained by observing that the submatrix $F_{rs} \neq 0$ only if the network contains Class r and Class s elements, and at least one Class r element closes a loop containing a Class s element. For example, the condition $F_{15} \neq 0$ would require at least one capacitor link (Class 1) that closes a loop containing a resistor tree branch (Class 5). The method used in SCEPTRE to form the specific network tree prohibits the occurrence of this condition, as well as the conditions F_{16} and $F_{26} \neq 0$.

Partitioning of V_L, V_T, I_L and I_T according to the classes defined earlier, leads to the following expansions of the Kirchhoff equations (1) and (2):

$$V_1 = -F_{14}V_4 \qquad\qquad\qquad - F_{17}E_7 \tag{3a}$$

$$V_2 = -F_{24}V_4 - F_{25}V_5 \qquad\qquad - F_{27}E_7 \tag{3b}$$

$$V_3 = -F_{34}V_4 - F_{35}V_5 - F_{36}V_6 - F_{37}E_7 \tag{3c}$$

$$V_8 = -F_{84}V_4 - F_{85}V_5 - F_{86}V_6 - F_{87}E_7\,, \tag{3d}$$

and

$$I_4 = F_{14}{}^T I_1 + F_{24}{}^T I_2 + F_{34}{}^T I_3 + F_{84}{}^T J_8 \tag{4a}$$

$$I_5 = F_{25}{}^T I_2 + F_{35}{}^T I_3 + F_{85}{}^T J_8 \tag{4b}$$

$$I_6 = F_{36}{}^T I_3 + F_{86}{}^T J_8 \tag{4c}$$

$$I_7 = F_{17}{}^T I_1 + F_{27}{}^T I_2 + F_{37}{}^T I_3 + F_{87}{}^T J_8\,. \tag{4d}$$

Derivation of network quantities

The purpose of this section is to present the matrix equations that SCEPTRE uses to compute all unknown network quantities in terms of known quantities (state variables, V_4 and I_3; sources, E_7 and J_8; and the submatrices of the F matrix). Appropriate adjustments for networks containing elements whose values are functions of time will be discussed in a later section.

- *Resistive quantities*

All resistor currents and voltages in any network are included in the vectors I_2, I_5, V_2 and V_5. The solution for I_2 makes use of the fact that

$$V_2 = R_{22}I_2 \tag{5}$$

$$V_5 = R_{55}I_5\,. \tag{6}$$

Substituting Eqs. (4b), (5) and (6) into Eq. (3b) yields

$$R_{22}I_2 = F_{24}V_4 - F_{25}R_{55}(F_{25}{}^T I_2 + F_{35}^T I_3 + F_{85}^T J_8) - F_{27}E_7\,,$$

or

$$I_2 = M_R^{-1}[-F_{24}V_4 - F_{25}R_{55}(F_{35}^T I_3 + F_{85}^T J_8) - F_{27}E_7]\,, \tag{7}$$

where

$$M_R = R_{22} + F_{25}R_{55}F_{25}^T\,. \tag{8}$$

The utility of Eq. (7) depends on the existence of the inverse of M_R. It is a straightforward matter to show that M_R is a positive definite matrix and, hence, its inverse exists.

Once I_2 has been computed from Eq. (7), the remaining resistive quantities I_5, V_2 and V_5 are computed by solving Eqs. (4b), (5) and (6), in that order.

It is possible to attain the same end using an alternate approach. Instead of using Eqs. (5) and (6), one would use the expressions

$$I_2 = G_{22}V_2 \tag{9}$$

$$I_5 = G_{55}V_5\,, \tag{10}$$

where $G_{22} = R_{22}^{-1}$ and $G_{55} = R_{55}^{-1}$

Then, substituting (3b), (9) and (10) into Eq. (4b) gives

$$G_{55}V_5 = F_{25}{}^T G_{22}[-F_{24}V_4 - F_{25}V_5 - F_{27}E_7] + F_{35}{}^T I_3 + F_{85}{}^T J_8$$

or

$$V_5 = M_G^{-1}[F_{25}{}^T G_{22}(-F_{24}V_4 - F_{27}E_7) + F_{35}{}^T I_3 + F_{85}{}^T J_8]\,, \tag{11}$$

where M_G is a positive definite matrix given by

$$M_G = G_{55} + F_{25}{}^T G_{22}F_{25}\,. \tag{12}$$

Once V_5 is known, the remaining resistive quantities V_2, I_2 and I_5 can be determined from Eqs. (3b), (9) and (10), respectively.

The practical difference between the two approaches lies in the size of the matrix that must be inverted. The M_R matrix of Eq. (8) is of dimension $n \times n$ where n equals the number of resistor links in any network. The M_G matrix of Eq. (12) is of dimension $m \times m$ where m equals the number of resistor tree branches in any network. In practice, it happens more often than not that $m \leq n$. Consequently, the second method of obtaining resistive quantities often requires the inversion of a smaller matrix than does the first method. SCEPTRE compares the number of tree branch resistors with the number of link resistors and evaluates M_R or M_G, depending on which is smaller.

- *Capacitative quantities*

All capacitor currents and voltages are included in the vectors I_4, I_1, V_4 and V_1. The vector of state variables, V_4, is a known quantity at the beginning of each time step, but the vector of the derivatives of the state variables, $\dot{V}_4$, must be determined. The unknown capacitive quantities then are I_1, I_4, V_1 and $\dot{V}_4$ and they are computed in that sequence.

The solution for I_1 is derived from the relations

$$\dot{V}_4 = S_{44}I_4 \tag{13}$$

$$\dot{V}_1 = S_{11}I_1\,, \tag{14}$$

and from differentiation of Eq. (3a)

$$\dot{V}_1 = -F_{14}\dot{V}_4 - F_{17}\dot{E}_7\,. \tag{15}$$

Note that Eq. (15) introduces the need for the derivative of the source variable, $\dot{E}_7$. Substituting Eqs. (4a), (13), and (14) into Eq. (15) gives

$$S_{11}I_1 = -F_{14}S_{44}[F_{14}^T I_1 + F_{24}^T I_2 + F_{34}^T I_3 + F_{84}^T J_8] - F_{17}\dot{E}_7$$

or

$$I_1 = M_C^{-1}[-F_{14}S_{44}(F_{24}^T I_2 + F_{34}^T I_3 + F_{84}^T J_8) - F_{17}\dot{E}_7]\,, \tag{16}$$

where M_C is a positive definite matrix given by

$$M_C = S_{11} + F_{14}S_{44}F_{14}^T\,. \tag{17}$$

The rest of the capacitive quantities, I_4, V_1 and $\dot{V}_4$ can now be determined from Eqs. (4a), (3a) and (13), respectively.

- *Inductive quantities*

All inductive currents and voltages are included in the vectors I_3, I_6, V_3 and V_6. The vector of state variables, I_3, is a known quantity at the beginning of each time step, but the vector of the derivatives of the state variable, $\dot{I}_3$ must be determined. The unknown inductive quantities, then, are $\dot{I}_3$, V_3, V_6 and I_6 and they are computed in that sequence.

The derivation of $\dot{I}_3$ makes use of the relations

$$V_3 = L_{33}\dot{I}_3 + L_{36}\dot{I}_6 \tag{18}$$

$$V_6 = L_{63}\dot{I}_3 + L_{66}\dot{I}_6 \tag{19}$$

and the derivative of Eq. (4c)

$$\dot{I}_6 = F_{36}^T\dot{I}_3 + F_{86}^T\dot{J}_8\,. \tag{20}$$

Note that Eq. (20) introduces the need for the derivative of the source variable $\dot{J}_8$. Manipulation of Eqs. (3c), (18), (19) and (20) leads to

$$\begin{aligned} L_{33}\dot{I}_3 + L_{36}(F_{36}^T\dot{I}_3 + F_{86}^T\dot{J}_8) &= -F_{34}V_4 - F_{35}V_5 - F_{36}L_{63}\dot{I}_3 \\ &\quad - F_{36}L_{66}F_{36}^T\dot{I}_3 - F_{36}L_{66}F_{86}^T\dot{J}_8 - F_{37}E_7 \end{aligned}$$

or

$$\dot{I}_3 = M_L^{-1}[-F_{34}V_4 - F_{35}V_5 - (F_{36}L_{66}F_{86}^T + L_{36}F_{86}^T)\dot{J}_8 - F_{37}E_7] \tag{21}$$

where

$$M_L = L_{33} + L_{36}F_{36}^T + F_{36}L_{63} + F_{36}L_{66}F_{36}^T\,. \tag{22}$$

Since L_{33} and L_{66} are not necessarily diagonal, M_L is not necessarily nonsingular. In fact, it can be shown that if unity coupling between inductors exists, M_L is singular. Hence, users of SCEPTRE are cautioned against permitting unity coupling in networks submitted for analysis.

Equation (21) gives the inductor-link current derivative vector in terms of known quantities. The remaining inductive quantities, V_3, V_6, and I_6, can now be obtained from Eqs. (18) and (20), (19) and (20), and (4c), respectively.

- *Current-source voltages and voltage-source currents*

Now that all passive-element currents and voltages have been determined, Eq. (3d) yields the vector of current-source voltages, and Eq. (4d) yields the vector of voltage source currents.

- *Special classes of dependent sources*

The formulation of the equations used to compute network quantities can be modified slightly to allow analysis of networks containing certain voltage and current sources that are dependent on resistor voltages and currents, respectively. The class of allowable dependent voltage sources, called Class Y elements, is defined by the matrix equation

$$E_Y = k_1V_2 + k_2V_5\,, \tag{23}$$

where k_1 is a matrix of constants, having a number of rows equal to the number of Class Y elements and a number of columns equal to the number of resistor links; k_2 has a number of rows equal to the number of Class Y elements and a number of columns equal to the number of tree branch resistors.

The class of allowable dependent current sources, called Class X elements, is defined by

$$J_X = k_3I_2 + k_4I_5\,, \tag{24}$$

where the matrices k_3 and k_4, are described analogously to k_1 and k_2.

The F matrix for networks containing such sources is enlarged to include the submatrices describing the interconnections between these sources and the other elements. That is,

$$F = \begin{bmatrix} F_{14} & 0 & 0 & F_{17} & F_{1Y} \\ F_{24} & F_{25} & 0 & F_{27} & F_{2Y} \\ F_{34} & F_{35} & F_{36} & F_{37} & F_{3Y} \\ F_{84} & F_{85} & F_{86} & F_{87} & F_{8Y} \\ F_{X4} & F_{X5} & F_{X6} & F_{X7} & F_{XY} \end{bmatrix}.$$

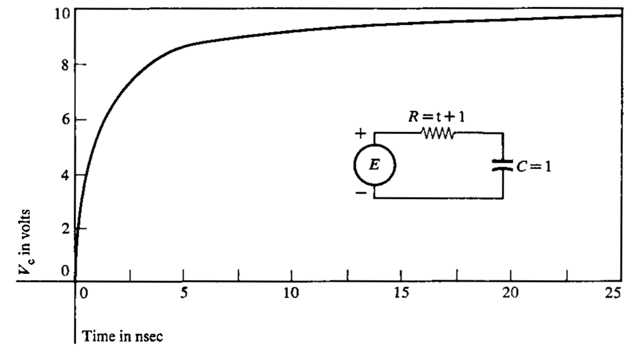


Figure 4 Computed solution for capacitor voltage in network containing time-varying resistor.

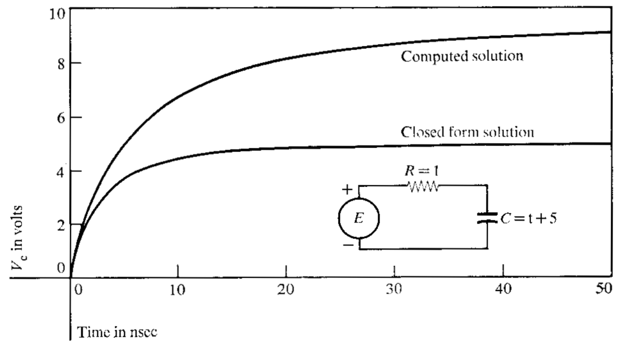


Figure 5 Comparison of computed solution and closed form solution for capacitor voltage in network containing time-varying capacitor.

Modification of the equations for the computation of voltage and current vectors is straightforward. Notice, however, that the modified equations for $\dot{V}_1$ and $\dot{I}_6$ (see Eqs. (15) and (20)) require the derivatives $\dot{E}_Y$ and $\dot{J}_X$ whenever F_{1Y} and F_{X6} are nonzero:

$$\dot{V}_1 = -F_{14}\dot{V}_4 - F_{17}\dot{E}_7 - F_{1Y}\dot{E}_Y$$

$$\dot{I}_6 = F_{36}^T\dot{I}_3 + F_{86}^T\dot{J}_8 + F_{X6}^T\dot{J}_X \,.$$

Differentiation of Eqs. (23) and (24) shows that $\dot{E}_Y$ depends on $\dot{V}_2$ and $\dot{V}_5$, and $\dot{J}_X$ depends on $\dot{I}_2$ and $\dot{I}_5$. Since SCEPTRE contains no provision for computing these derivatives, the program automatically checks for the existence of non-zero F_{1Y} or F_{X6} and terminates the run if one of them occurs. Hence, there is some restriction on the position that dependent sources may occupy in a network to be analyzed by SCEPTRE. The prohibited circuits are capacitor loops containing a dependent voltage source, and inductor cut-sets containing a dependent current source.

Extension to time-varying networks

The purpose of this section is to discuss the application of the preceding formulation to the class of networks in which one or more of the passive elements are variable functions of time. First, consider a series RC network in which the capacitance is constant, $C = 1$, and the resistor is a variable function of time, $R = t + 1$. Assume there is no initial charge on the capacitor and that a 10-V step function is applied to the circuit. The network is described by the equation

$$\frac{dq}{dt} + \frac{q}{(t+1)C} = \frac{10}{t+1}\,,$$

and the capacitor voltage is given by the particular solution

$$V_c(t) = -\frac{10}{t+1} + 10\,.$$

A plot of the computed solution based on the formulation of this paper is given in Fig. 4. This result is virtually indistinguishable from the closed form solution since the maximum error was about 0.02 V. This error can be attributed to roundoff in the computer and truncation in the integration routine.

Now consider the same network, but let the resistance be constant, $R = 1$, and the capacitance a variable function of time, $C = t + 5$. Again, assume there is no initial charge and that a 10-V step function is applied. The network equation is

$$\frac{dq}{dt} + \frac{q}{R(t+5)} = \frac{10}{R}\,,$$

and the closed form solution for the capacitor voltage is

$$V_c(t) = -\frac{125}{(t+5)^2} + 5\,.$$

Fig. 5 presents plots of both the closed form solution and the computed solution. A significant error is evident. The source of this error can be determined by an examination of the manner in which state variable derivatives are computed. The relation $I = C(dV/dt)$ is implicit in Eq. (13): $\dot{V}_4 = S_{44}I_4$. However, it is always true that if $C = Q/V$,

$$I = \frac{dQ}{dt} = \frac{d}{dt}(CV) = C\frac{dV}{dt} + \frac{dC}{dt}V\,. \qquad (25)$$

The formulation of SCEPTRE has not included the $(dC/dt)V$ term. Hence, it is clear that this error will be incurred for networks containing time varying capacitors unless the analyst applies a compensating procedure.

One such procedure is to shunt the capacitor by a current generator J_c where $J_c = V_c(dC/dt)$. The total current into the node becomes $(CdV/dt) + (VdC/dt)$, which effects the necessary correction. A computer run with the additional current generator was made for purposes of corroboration. The error was again about 0.02 V.

Exactly the dual situation exists for time-varying inductors. Equation (18) in the program formulation ($V_3 = L_{33}\dot{I}_3 + L_{36}\dot{I}_6$) implies that $V = L(dI/dt)$. It is always true that $V = d\phi/dt$ and if $L = \phi/I$,

$$V = \frac{d\phi}{dt} = \frac{d}{dt}(LI) = L\frac{dI}{dt} + \frac{dL}{dt}I. \tag{26}$$

Hence the formulation will produce an error in the solution of networks with time-varying inductors. Correction may be applied in the form of a series voltage source E_L, where $E_L = (dL/dt)I$.

Another small but very practical class of networks that cannot be classed as linear may be accurately accommodated by this formulation. Consider the situation in which one or more variable capacitors exist in a network such that $C = C(V)$. If capacitance is defined as $C = Q/V$, then (25) may be written as

$$I = \frac{dQ}{dt} = \frac{d}{dt}C(V)V = C(V)\frac{dV}{dt} + V\frac{d}{dt}C(V)$$

$$= \left[C(V) + V\frac{dC(V)}{dV}\right]\frac{dV}{dt}. \tag{27}$$

It is clear that (27) contains a term that is not provided by the formulation Eq. (13) and an error must result. If, however, the capacitances of interest are instead defined as $C = dQ/dV$, then substitution into (13) yields simply $I = dQ/dt$ which is certainly correct as it stands. The class of voltage dependent capacitors that are defined this way are the transition and diffusion capacitances associated with semiconductor junctions.[16,18] This definition allows the formulation in this paper to accurately determine the transient response of transistor and diode networks without change.

Application and utilization

The special subroutines and auxiliary functions that make up the SCEPTRE program system require over 15,000 instructions, which in turn require over 5000 punched cards. As a matter of convenience, most of the instructions are stored on a system tape and the user need supply only a comparatively small number of cards. The cards that are supplied by the user may be divided into two groups: a starter deck and a problem deck. The information contained in the starter deck together with the IBSYS Monitor System controls the insertion of the basic program into core storage and onto an overlay tape. The latter tape is necessary because the entire program cannot be accommodated by the 32,000-word capacity that is available with conventional IBM 7094 computing systems. It should be emphasized that the starter deck is entirely independent of the specific network that is to be solved.

Once the information contained in the starter deck has been processed, the specific network information contained in the problem deck is utilized to carry out the desired transient solution. When the amount of output information exceeds an allotted amount, it is buffered onto the intermediate output tape as many times as necessary. At the conclusion of the run, the output data is processed into the proper format and stored on the output tape. This information is then converted into conventional printed and machine plotted form at any subsequent time. The starter deck consists of about 200 cards and the problem deck usually contains from 20 to 100. The problem deck must contain all the topological and quantitative information that describes the particular network under analysis. The following prerequisites must be met before the problem deck for any network can be prepared:

1. The network must be composed of resistances, capacitances, inductances (including mutual inductances), and voltage and current sources. All active elements must, therefore, be represented by equivalent circuits that are made up of these basic components.
2. A consistent set of units must be used to describe all quantitative information.
3. All nodes in the network must be identified.
4. Assumed current directions must be chosen for all passive elements in the network.

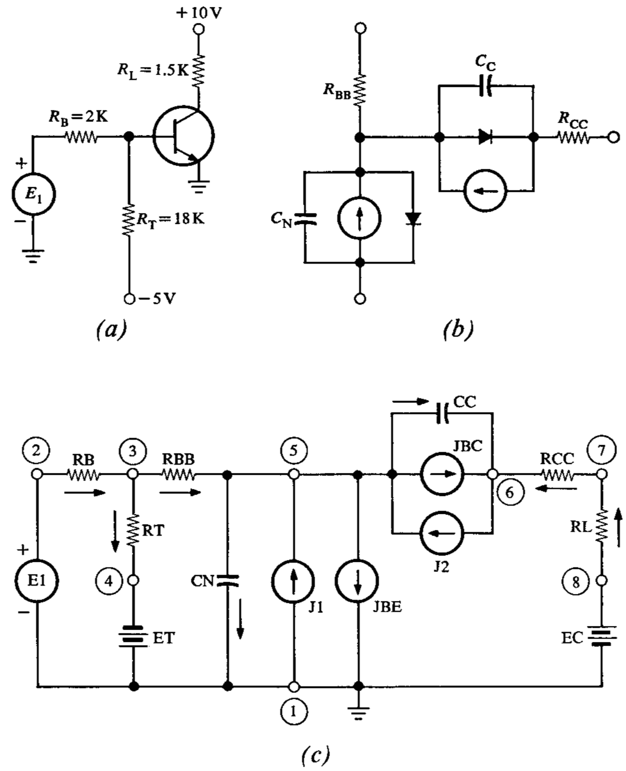


Figure 6 Preparation of circuit diagram for solution by computer program. (a) Inverter circuit (b) Ebers-Moll equivalent transistor circuit (c) Inverter circuit diagram in format required by program.

The circuit shown in Fig. 6a will be used to illustrate the procedure required of the designer who wishes to make use of SCEPTRE. The schematic shows an inverter circuit which is biased in the off condition by the negative dc voltage that is applied to the transistor base. Let it be desired to determine the transient response of this circuit to an input voltage pulse supplied by the voltage generator E1. The first requirement is met by replacing the transistor by the conventional Ebers-Moll large-signal equivalent circuit shown in Fig. 6b. (It is not required, however, that this particular equivalent circuit be used.) The diodes of the equivalent circuit are in turn represented by voltage dependent current sources that reflect the current-voltage relationships that exist in these devices.

The second prerequisite requires that all quantitative information be entered in a consistent set of units. One set of units that works well with modern transistorized networks is

Time in nanoseconds
Voltage in volts
Current in milliamperes
Resistance in kilohms
Capacitance in picofarads
Inductance in microhenries.

To satisfy the other two preliminary requirements, each of the nodes in the network is identified by a number (or letter) and assumed current directions through each of the passive elements in the network are indicated. The nodes may be numbered in any arbitrary fashion; numerical sequence need not be observed. The assumed current direction through the passive elements is also arbitrary. The effect of an "incorrect" choice will show up only in the sign of the computed current or voltage. If the equivalent circuit of Fig. 6b replaces the transistor in Fig. 6a, the resulting network of Fig. 6c is in a form suitable for the preparation of a problem deck. A problem deck listing for this circuit is given in Table 1. There are five header cards in this listing. The information contained under each heading is coded in the following fashion:

1. ELEMENTS—The names of all circuit elements (resistors, capacitors, inductors, mutual inductances, voltage and current sources), their values, their terminal nodes, and the assumed directions of current through them are coded in the format: (ELEMENT), (FROM NODE) — (TO NODE) = (ELEMENT VALUE). For example, RT, 3–4 = 18 indicates that resistor *RT* is connected between nodes 3 and 4 with the assumed positive sense of current flow from 3 toward 4, and that this resistor contains 18 units of resistance. If an element has a variable value specified by some function in either tabular or equation form, the coded entry for ELEMENT VALUE will be, for example, TABLE 1 or EQUATION 3. The table or equation is then given in detail under the "Functions" header.

Table 1 Problem deck

```
ELEMENTS
E1, 1 – 2 = TABLE 1
ET, 4 – 1 = 5
EC, 1 – 8 = 10
CN, 5 – 1 = EQUATION 3 (10., 80., JBE)
CC, 5 – 6 = EQUATION 3 (10., 400., JBC)
RB, 2 – 3 = 2
RT, 3 – 4 = 18
RBB, 3 – 5 = .2
RCC, 7 – 6 = .015
RL, 8 – 7 = 1.5
*JBE, 5 – 1 = DIODE EQUATION (1.E – 7,35.)
JBC, 5 – 6 = DIODE EQUATION (5.E – 7,37.)
J1, 1 – 5 = .1 * JBC
J2, 6 – 5 = .98 * JBE
OUTPUTS
VCN, IRL
INITIAL CONDITIONS
VCN = –.5
VCC = –10.5
FUNCTIONS
TABLE 1
0, 0
10, 5
25, 2.5
40, 1.5
60, .7
90, .1
100, 0
120, 0
EQUATION 3 (A, B, C) = (A + B * C)
RUN CONTROLS
STOP TIME = 500
END
```

* The diode equation is prestored in the program and need not be supplied in detail.

2. OUTPUTS—All voltages and currents that the program user wishes to see as output are listed under this heading. The notation VCN calls for the voltage across capacitor CN. The output values are always produced in printed form, but a machine plotted version is optionally available.
3. INITIAL CONDITIONS—The initial values of all capacitor voltages and inductor currents are specified under this heading. This section of the list may be omitted if all initial values are zero. If the initial conditions are unknown, they can be automatically computed by the program upon request.
4. FUNCTIONS—This section is used to define explicitly all the tables and equations, if any, that were referenced but not described under the ELEMENTS subheading. The general format for table descriptions includes a card giving the name of the table followed by cards which contain pairs of numbers describing the numerical values of the independent and dependent variables in that order. All intermediate points of the independent variable are used in a linear interpolation routine to determine the corresponding value of the dependent variable. It is clear

that a more accurate representation of a nonlinear function may be obtained by the use of more entries in the table.

5. RUN CONTROLS—This section contains all the auxiliary information needed to control the run. Most of these quantities (for example, maximum, minimum, and starting step sizes for the integration routine) have automatically preset entries that hold unless the user chooses to replace them with other values. The one quantity that must always be supplied is the real-time problem duration in the time units that are used.

A complete discussion of the coding procedure is given in the SCEPTRE user's manual.[13]

Factors that influence running time

Users of automatic transient circuit analysis programs are understandably concerned with the amount of computer time required for the solution of problems. A knowledge of the factors that lead to excessive solution times can often permit the user to modify if not completely avoid them. The most important factors are:

1. Number of differential equations required. This number is exactly equal to the number of independent capacitors and inductors in the network and is not necessarily proportional to the size of the network.
2. Variable resistors. If all network resistors are constant in value, only one inversion of the matrix M_R or M_G is necessary. If one or more resistors are variable, however, the appropriate matrix must be inverted at each time step. Since most practical networks require hundreds or thousands of time steps for solution, it is clear that many more matrix inversions, and therefore, more computer time is needed.
3. Large forcing functions or dependent sources. These quantities lead to large values for the state variable derivatives and force the integration routine to proceed at smaller time steps to maintain solution accuracy.
4. Network time constants. Small time constants in the network will cause the integration routine to operate at small step sizes and will, therefore, require that many steps be taken.

The influence of network time constants on computer solution time can be illustrated by example. Consider the network of Fig. 7 which contains poles in the s-plane at $\lambda_1 = 1$ and $\lambda_2 = 10{,}001$.

The largest step size that may be taken during the transient solution is limited by the stability radius of the particular integration routine that is used. It can be shown that $h_{\max}$ (i.e., maximum time increment in an integration routine) of a modified trapezoidal integration routine[5] is approximately equal to six times the reciprocal of the absolute value of the largest pole in the network.

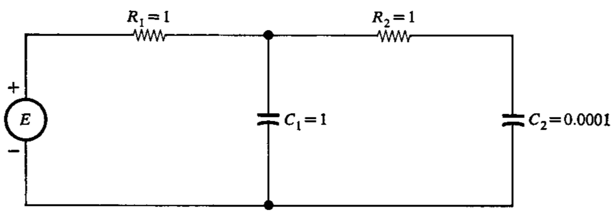


Figure 7 Two-pole network.

More information is available when it is realized that the closed form solution for the state variables of this network must be:

$$V_{C1} = K_1 e_1^{\lambda_1 t} + K_2 e^{\lambda_2 t} + K_3$$
$$V_{C2} = K_4 e^{\lambda_1 t} + K_5 e^{\lambda_2 t} + K_6 .$$

Since $|\lambda_2| > |\lambda_1|$, it is clear that the problem duration must be at least $4/|\lambda_1|$ to assure that the network has practically settled to its final value. The minimum number of steps required for a complete transient solution is simply the ratio of the problem duration to the maximum step size. For the example of Fig. 7 the minimum number of steps is $(4/\lambda_1)/(6/\lambda_2) = 6667$. This may be contrasted to the situation that arises if C_2 in Fig. 7 is increased to 0.01 units of capacitance. Then, poles exist at $\lambda_1 = -1$ and $\lambda_2 = -101$ and only 68 steps are required for a complete transient solution. Since the amount of computer solution time per step is unaffected by the size of any element, it is clear that the larger C_2 reduces computer time by a factor of about 100. It is therefore quite clear that the computer user must be very critical of the insertion of unnecessarily small time constants in practical networks since these can lead to widely separated poles (or eigenvalues). The analysis given in the preceding paragraph obviously depends upon the characteristics of the numerical integration method that is used. SCEPTRE users have the choice of three different integration methods, the best of which, from the standpoint of the accuracy-speed trade-off, is usually the Fowler-Warten method.[20]

Acknowledgments

The author acknowledges the indispensible contributions to the development of the program by Dr. R. Warten and J. Sents. In addition, he thanks Dr. T. Bickart of Syracuse University for his advice on numerous technical points.

References

1. F. F. Kuo, "Network Analysis by Digital Computer," *IEEE Proceedings* **54,** No. 6, 820–829 (June, 1966).
2. *1620 Electronic Circuit Analysis Program,* Users' Manual H20-0170-1, International Business Machines Corporation, White Plains, New York, 1965.
3. F. H. Branin, "DC and Transient Analysis of Networks Using a Digital Computer," *IEEE International Convention Record, Part* **2,** pp. 236–256 (1962).

4. A. Malmberg and F. Cornwell, *NET-1 Network Analysis Program*, Report LA2853, Los Alamos Scientific Laboratory, Los Alamos, New Mexico, 1963.
5. *Automated Digital Computer Program for Determining Responses of Electronic Systems to Transient Nuclear Radiation*, Vol. II, IBM File No. 64-521-5, Owego, New York, July, 1964.
6. S. Seshu and N. Balbanian, *Linear Network Analysis*, Wiley & Sons, Inc., New York, 1959
7. L. Zadeh and C. Desoer, *Linear System Theory*, McGraw-Hill Book Company, New York, 1963
8. S. Seshu and M. B. Reed, *Linear Graphs and Electrical Networks*, Addison-Wesley, Reading, Mass., 1961.
9. T. R. Bashkow, "The A Matrix, New Network Description," *IRE Trans. Circuit Theory* **CT-4,** 117–120 (Sept., 1957).
10. P. Bryant, "The Explicit Form of Bashkow's A Matrix," *IRE Trans. Circuit Theory* **CT-9** (Sept., 1962).
11. A. Dervisoglu, "Bashkow's A Matrix for Active RLC Networks," *IEEE Trans. Circuit Theory*, **CT-11** (Sept., 1964).
12. E. Kuh and R. Rohrer, "The State-Variable Approach to Network Analysis," *IEEE Proceedings* **53,** 672–686 (July, 1965).
13. *Automated Digital Computer Program for Determining Responses of Electronic Systems to Transient Nuclear Radiation* (*SCEPTRE*), Vol. 1, IBM No. 66-928-611, Owego, New York, September, 1966.
14. S. Sedore and J. Sents, "SCEPTRE Gives Designers New Power," *Electronic Design* **15** (March 15, 1967).
15. R. M. Warten, "Automatic Step Size Control for Runge-Kutta Integration," *IBM Journal* **7,** 340–341 (Oct., 1963).
16. A. B. Phillips, *Transistor Engineering*, McGraw-Hill Book Company, New York (1962).
17. E. Kreyszig, *Advanced Engineering Mathematics*, Wiley & Sons, Inc., New York (1962).
18. R. P. Nanavati *An Introduction to Semiconductor Electronics*, McGraw-Hill Book Co., New York (1963).
19. J. Katznelson, "AEDNET: A Simulator for Nonlinear Networks," *Proc. IEEE* **54,** No. 11, 1536 (1966).
20. M. E. Fowler and R. M. Warten, "A Numerical Integration Technique for Ordinary Differential Equations with Widely Separated Eigenvalues," *IBM Journal* **11,** No. 5, 537 (1967).

Received January 24, 1967