

AD-751 519

MATHEMATICAL FORMULATION OF SCEPTRE

Stephen R. Sedore, et al

IBM Federal Systems Division

Prepared for:

Air Force Weapons Laboratory

September 1972

DISTRIBUTED BY:

NTIS

National Technical Information Service
U. S. DEPARTMENT OF COMMERCE
5285 Port Royal Road, Springfield Va. 22151

AFWL-TR-72-77

AFWL-TR-
72-77

AD751519



MATHEMATICAL FORMULATION OF SCEPTRE

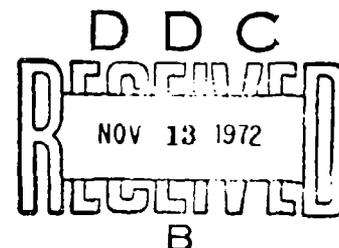
Stephen R. Sedore

John R. Sen's

IBM Corporation

TECHNICAL REPORT NO. AFWL-TR-72-77

September 1972



AIR FORCE WEAPONS LABORATORY

Air Force Systems Command

Kirtland Air Force Base

New Mexico

**NATIONAL TECHNICAL
INFORMATION SERVICE**

Approved for public release; distribution unlimited.

70

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified.)

1. ORIGINATING ACTIVITY (Corporate author) IBM Corporation Electronics Systems Center Owego, New York 13827		2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	
		2b. GROUP	
3. REPORT TITLE MATHEMATICAL FORMULATION OF SCEPTRE			
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) 15 April 1970 through 1 July 1972			
5. AUTHOR(S) (First name, middle initial, last name) Stephen R. Sedore and John R. Sents			
6. REPORT DATE September 1972		7a. TOTAL NO. OF PAGES 68	7b. NO. OF REFS 9
8a. CONTRACT OR GRANT NO. F29601-70-C-0038		9a. ORIGINATOR'S REPORT NUMBER(S) AFWL-TR-72-77	
b. PROJECT NO. 5710			
c. Subtask TC015		9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	
d.			
10. DISTRIBUTION STATEMENT Approved for public release; distribution unlimited.			
11. SUPPLEMENTARY NOTES		12. SPONSORING MILITARY ACTIVITY AFWL (ELT) Kirtland AFB, NM 87117	
13. ABSTRACT (Distribution Limitation Statement A) SCEPTRE is a general DC and transient analysis program with a capability for handling radiation effects circuit analysis problems. This report contains the theory and formulation that provides a basis for SCEPTRE and a description including flow diagrams of the program operation.			

I

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Automated circuit analysis Response, electronic circuits Radiation nuclear, transient Digital computer program SCEPTRE						
II						

AFWL-TR-72-77

MATHEMATICAL FORMULATION OF SCEPTRE

Stephen R. Sedore

John R. Sents

IBM Corporation

TECHNICAL REPORT NO. AFWL-TR-72-77

III

Approved for public release; distribution unlimited.

AIR FORCE WEAPONS LABORATORY
Air Force Systems Command
Kirtland Air Force Base
New Mexico 87117

When US Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

DO NOT RETURN THIS COPY. RETAIN OR DESTROY.

IV

AFWL-TR-72-77

MATHEMATICAL FORMULATION OF SCEPTRE

Stephen R. Sedore

John R. Sents

IBM Corporation

TECHNICAL REPORT NO. AFWL-TR-72-77

Approved for public release; distribution unlimited.

V

FOREWORD

This report was prepared by the IBM Corporation, Owego, New York, under Contract F29601-70-C-0038. The research was performed under Program Element 61102H, Project 5710, Subtask TC015.

Inclusive dates of research were 15 April 1970 through 1 July 1972. The report was submitted 2 August 1972 by the Air Force Weapons Laboratory Project Officer, Captain Raymond B. McBride (ELT).

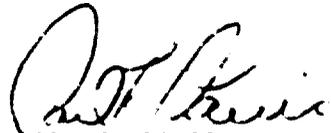
This technical report has been reviewed and is approved.



RAYMOND B. McBRIDE
Captain, USAF
Project Officer



CARL F. PORTER
Lt Colonel, USAF
Chief, Transient Radiation
Effects Branch



CARL F. DAVIS
Colonel, USAF
Chief, Electronics Division

VI

ILLUSTRATIONS

Figure	Title	Page
1	Composite B Matrix in SCEPTR	3
2	Transient-Solution B Matrix in SCEPTR	7
3	B Matrix from Hypothetical Network	13
4	Low-Frequency h-Parameter Equivalent Circuit	14
5	B Matrix with Linearly Dependent Sources	15
6	Capacitor Cut Set Circuit	24
7	Inductor Loop Circuit	25
8	A Current-Source Capacitor Cut Set	26
9	Root Locus of TRAP Characteristic Equation	32
10	System Flow Diagram	44
11	Program Generator (EXEC1) Flow Diagram	45
12	Circuit Description Processor (CDPRS) Flow Diagram	46
13	Model Editor (MEDIT) Flow Diagram	47
14	Circuit Equation Generator (GENER 8) Flow Diagram	48
15	Rerun Processor and Data Generator (RRPRO) Flow Diagram	49
16	Circuit Solution Executive (EXEC 2) Flow Diagram	50

TABLES

Table	Title	Page
I	Element Classifications in SCEPTR	4
II	General B Matrix from the L Tree	18

Section 1

INTRODUCTION

Under Air Force Weapons Laboratory (AFWL) contract AF29 (601)-5399 IBM developed an automated transient circuit analysis digital computer program, PREDICT, for analyzing the responses of electronic circuits in transient radiation environments, and a system of digital computer programs, TREAT, for processing component radiation effects data. These programs were subsequently maintained, modified, and distributed to various aerospace companies throughout the United States. After extensive application of PREDICT, a number of modifications and improvements were suggested.

Under AFWL contract AF29 (601)-6489 IBM developed and evaluated the suggested improvements, and recommended that a new circuit analysis program incorporating these features be developed. The effort under AFWL contract AF29 (601)-6852 consisted primarily of formulating and coding this new circuit analysis program, which was called SCEPTRE. The most recent contract, F29601-70-C-0038, was awarded for the purpose of implementation of a series of desirable improvements.

Section II

FORMULATION AND THEORY

2.1 INTRODUCTION

Any automatic transient analysis program is designed to relieve the user of the necessity of writing and programming the differential and algebraic equations that describe networks. PREDICT and other programs already perform this basic task, but the degree of flexibility permitted the user varies widely among programs. SCEPTRE, written as a successor to PREDICT, incorporates many improvements.

This section presents the formulation and theory that serve as the basis for SCEPTRE. Therefore, the discussion is mathematically oriented. Those features of SCEPTRE that are not mathematical are not included here. For example, the extremely useful features of Rerun and Model Storage are not described.

2.2 GENERAL SOLUTION

SCEPTRE consists of two separate formulations that combine to produce the general transient solutions of a given network. One is referred to as the initial-conditions program. This will determine the network voltages that prevail before any time-varying forcing function is applied. This program does not treat time as an independent variable; instead it holds time constant, and iterates on selected voltages. The output of the initial-conditions program may be obtained independently or it may be automatically used as the starting point for the transient program. Thus, the output of the initial-conditions program effectively supplies the initial conditions for the system of differential equations that are solved by the transient program.

The second program is called the transient program. This program uses time as an independent variable and solves systems of differential equations as functions of time. The output of this program represents the transient response of a given network. As implied above, the transient program may be used in conjunction with the initial-conditions program, or it may be used by itself if the initial conditions of the network are known.

The general solution procedure described here concerns the definition of the terms, matrices and procedures that are common to both programs. Other parts of this volume will provide the detailed explanations and derivations. The first step in either program is the construction of a tree* (Ref. 1) according to prescribed rules, which differ for the two programs. This permits formation of a B matrix that effectively expresses link voltages in terms of tree branch voltages and tree branch currents in terms of link currents. Figure 1 shows a composite B matrix that contains all possible element classifications and submatrices. This matrix is derived in appendix I. The element classifications are given in table I.

		Tree Branches				
		Class 4	Class 5	Class 6	Class 7	Class Y
Links	Class 1	B_{14}	B_{15}	B_{16}	B_{17}	B_{1Y}
	Class 2	B_{24}	B_{25}	B_{26}	B_{27}	B_{2Y}
	Class 3	B_{34}	B_{35}	B_{36}	B_{37}	B_{3Y}
	Class 8	B_{84}	B_{85}	B_{86}	B_{87}	B_{8Y}
	Class 9	B_{94}	B_{95}	B_{96}	B_{97}	B_{9Y}
	Class 0	B_{04}	B_{05}	B_{06}	B_{07}	B_{0Y}
	Class X	B_{X4}	B_{X5}	B_{X6}	B_{X7}	B_{XY}

Figure 1. Composite B Matrix in SCEPTR

Reproduced from
best available copy.

*A tree is defined as any connected network subgraph that contains all nodes of the network but no complete loops. All circuit elements that are members of the tree are termed tree branches. All circuit elements excluded from the tree are termed links. A "C" tree is defined in this report as one in which tree members are chosen in the preference order E, C, R and L. All current sources (J) must be excluded from the "C" tree. Therefore, these sources are links. A cut set is defined as that group of elements that would isolate two groups of nodes when removed from a network.

Table I

ELEMENT CLASSIFICATIONS IN SCEPTR

Class	Element
1	Capacitor links
2	Resistor links
3	Inductor links
4	Capacitor tree branches
5	Resistor tree branches
6	Inductor tree branches
7	Voltage sources
8	Independent current sources
9	Primary current sources (dependent on voltage across terminals). This class will appear only in the derivation of the initial-conditions program.
0	Secondary current sources (dependent on other current sources). This class will appear only in the derivation of the initial-conditions program.
Y	Voltage sources that are dependent on resistor voltages. This class will appear only in the derivation of the transient program.
X	Current sources that are dependent on resistor currents. This class will appear only in the derivation of the transient program.

The following matrices and vectors may also be defined as a result of the element classification in table I:

- R_{22} - a diagonal matrix composed only of resistor links
- $G_{22} = R_{22}^{-1}$
- R_{55} - a diagonal matrix composed only of resistor tree branches
- $G_{55} = R_{55}^{-1}$
- C_{11} - a diagonal matrix composed only of capacitor links
- $S_{11} = C_{11}^{-1}$
- C_{44} - a diagonal matrix composed only of capacitor tree branches
- $S_{44} = C_{44}^{-1}$
- L_{33} - a matrix composed only of inductor links and the mutual inductance between inductor links
- L_{36} - a matrix composed only of the mutual inductance between inductor links and inductor tree branches
- L_{66} - a matrix composed only of inductor tree branches and the mutual inductance between inductor tree branches
- G_{99} - a diagonal matrix composed only of the voltage derivatives of primary current sources
- I_1, V_1 - vectors composed only of the currents or voltages associated with capacitor links

- I_2, V_2 - vectors composed only of the currents or voltages associated with resistor links
- I_3, V_3 - vectors composed only of the currents or voltages associated with inductor links
- I_4, V_4 - vectors composed only of the currents or voltages associated with capacitor tree branches
- I_5, V_5 - vectors composed only of the currents or voltages associated with resistor tree branches
- I_6, V_6 - vectors composed only of the currents or voltages associated with inductor tree branches
- J_8, V_8 - vectors composed only of the currents or voltages associated with independent current sources
- J_9, V_9 - vectors composed only of the currents or voltages associated with primary current sources
- J_0, V_0 - vectors composed only of the currents or voltages associated with secondary current sources
- E_7 - a vector composed only of voltage sources.

2.3 TRANSIENT SOLUTION

The state variables of any system can be defined as the minimum set of quantities that will suffice to determine all other quantities in the system at any instant. It can be shown that the knowledge of the set of all capacitor tree branch voltages (V_4) and inductor tree link currents (I_3) is sufficient to determine all other element currents and voltages, and furthermore, that this selection of state variables will allow network formulation in terms of first-order differential equations. The starting point of the derivation then is that quantities V_4 and I_3 are known and the list of unknowns is made up of $V_1, V_2, V_3, V_5, V_6, I_1, I_2, I_4, I_5, I_6, I_7$ and V_8 . In addition, the derivatives of the state variables, V_4 and I_3 , must be obtained in preparation for the numerical integration routine which produces the updated state variables that are valid at the next time increment.

Some of the equations needed to solve the unknown quantities may be obtained from the transient-solution B matrix (figure 2). The B matrix itself arises from a "C" tree, which is formed by an E, C, R, L preference order. Note that the B matrix differs from the composite matrix of

		Tree Branches			
		Class 4	Class 5	Class 6	Class 7
Links	Class 1	B_{14}	0	0	B_{17}
	Class 2	B_{24}	B_{25}	0	B_{27}
	Class 3	B_{34}	B_{35}	B_{36}	B_{37}
	Class 8	B_{84}	B_{85}	B_{86}	B_{87}

Figure 2. Transient-Solution B Matrix in SCEPTRE

Figure 1 in that some submatrices are zero valued* and that no distinction is made between types of sources. Since link voltages can be written in terms of tree branch voltages directly from the B matrix, the following equations may be written in matrix form:

$$V_1 = -B_{14} V_4 - B_{17} E_7 \quad (1)$$

$$V_2 = -B_{24} V_4 - B_{25} V_5 - B_{27} E_7 \quad (2)$$

$$V_3 = -B_{34} V_4 - B_{35} V_5 - B_{36} V_6 - B_{37} E_7 \quad (3)$$

$$V_8 = -B_{84} V_4 - B_{85} V_5 - B_{86} V_6 - B_{87} E_7 \quad (4)$$

Since tree branch currents can be written in terms of link currents, there arises:

$$I_4 = B_{14}^T I_1 + B_{24}^T I_2 + B_{34}^T I_3 + B_{84}^T I_8 \quad (5)$$

*For example, B_{15} must be zero since the "C" tree preference prohibits the possibility of a capacitor link closing a loop that contains a resistor tree branch.

$$I_5 = B_{25}^T I_2 + B_{35}^T I_3 + B_{85}^T J_8 \quad (6)$$

$$I_6 = B_{36}^T I_3 + B_{86}^T J_8 \quad (7)$$

$$I_7 = B_{17}^T I_1 + B_{27}^T I_2 + B_{37}^T I_3 + B_{87}^T J_8 \quad (8)$$

where the superscript T is used to indicate the transpose of a matrix. Two additional equations may be obtained from differentiating Equations (1) and (7) yielding:

$$\dot{V}_1 = -B_{14} \dot{V}_4 - B_{17} \dot{E}_7 \quad (9)$$

$$\dot{i}_6 = B_{36}^T \dot{i}_3 + B_{86}^T \dot{j}_8 \quad (10)$$

Note that source derivatives \dot{E}_7 and \dot{j}_8 have been introduced in the last two equations. * These equations together with a few fundamental relations will be used to derive all of the network currents and voltages in terms of known quantities.

2.3.1 SOLUTION OF RESISTIVE QUANTITIES

The resistive quantities of interest are I_2 , I_5 , V_2 , and V_5 . The fundamental voltage current relations for resistors permit:

$$V_2 = R_{22} I_2 \quad (11)$$

$$V_5 = R_{55} I_5 \quad (12)$$

I_2 may be explicitly solved for by manipulation of equations (2), (6), (11) and (12) to get

$$I_2 = M_R^{-1} \left\{ -B_{24} V_4 - B_{27} E_7 - B_{25} R_{55} [B_{35}^T I_3 + B_{85}^T J_8] \right\} \quad (13)$$

where

$$M_R = R_{22} + B_{25} R_{55} B_{25}^T \quad (14)$$

*See section 2.5 for a discussion on source derivatives.

The significance of equation (13) is that the vector of resistor link currents has been solved in terms of all known quantities since the right side of the equation is composed entirely of state variables V_4 and I_3 , known sources E_7 and J_8 , and known incidence submatrices. Once I_2 is known, the vectors I_5 , V_2 , and V_5 can be determined from Equations (6), (11), and (12), respectively.

An alternate approach may sometimes be preferable. Equations (2), (6), (11), and (12) may also be manipulated to obtain

$$V_5 = M_G^{-1} \left\{ -B_{25}^T G_{22} [B_{24} V_4 + B_{27} E_7] + B_{35}^T I_3 + B_{85}^T J_8 \right\} \quad (15)$$

where

$$M_G = G_{55} + B_{25}^T G_{22} B_{25} \quad (16)$$

Equation (15) gives the vector of resistor tree branch voltages in terms of quantities that are all known. Then, V_2 , I_2 , and I_5 can be solved by Equation (2), yielding

$$I_2 = G_{22} V_2 \quad (17)$$

$$\text{and } I_5 = G_{55} V_5 \quad (18)$$

respectively. The two approaches differ in the size of the matrix to be inverted. The first approach requires the inversion of a matrix (M_R) containing the number of rows and columns equal to the number of class-2 elements in the network. The second approach requires the inversion of a matrix (M_G) containing the number of rows and columns equal to the number of class-5 elements in the network. Networks containing resistors that are all constant require only one matrix inversion. There is no practical difference between the two approaches. If, however, the network contains at least one variable resistor, a matrix must be inverted at each solution time increment. Hundreds or even thousands of matrix inversions are necessary, and the size of the matrix becomes of significant importance. SCEPTRE will automatically determine which of the two approaches should be taken for each individual network.

2.3.2 SOLUTION OF CAPACITOR QUANTITIES

The capacitor quantities that must be solved at each time step are I_1 , I_4 , V_1 and \dot{V}_4 . Vector V_4 itself will have been updated by the integration routine and hence will be known. The fundamental relationships for capacitors permit

$$\dot{V}_4 = S_{44} I_4 \quad (19)$$

$$\dot{V}_1 = S_{11} I_1 \quad (20)$$

Equations (5), (9), (19), and (20) may be combined to obtain

$$I_1 = M_S^{-1} \left\{ -B_{14} S_{44} \left[B_{24}^T I_2 + B_{34}^T I_3 + B_{84}^T J_8 \right] - B_{17} \dot{E}_7 \right\} \quad (21)$$

where

$$M_S = S_{11} + B_{14} S_{44} B_{14}^T \quad (22)$$

At this point, vector I_1 has been isolated in terms of all known quantities. There remains to obtain I_4 , V_1 , and \dot{V}_4 from equations (5), (1), and (19), respectively.

2.3.3 SOLUTION OF INDUCTIVE QUANTITIES

The inductive quantities that must be solved at each time step are \dot{I}_3 , V_3 , V_6 , and I_6 . Vector I_3 will have been updated at the start of each time step and will be known. The fundamental relations for inductors permit

$$V_3 = L_{33} \dot{I}_3 + L_{36} \dot{I}_6 \quad (23)$$

$$V_6 = L_{63} \dot{I}_3 + L_{66} \dot{I}_6 \quad (24)$$

Equations (3), (10), (23), and (24) may be combined to obtain

$$\dot{I}_3 = M_L^{-1} \left\{ -B_{34} V_4 - B_{35} V_5 - B_{37} E_7 - \left[B_{36} L_{66} B_{86}^T + L_{36} B_{86}^T \right] J_8 \right\} \quad (25)$$

where

$$M_L = L_{33} + L_{36} B_{36}^T + B_{36} L_{63} + B_{36} L_{66} B_{36}^T \quad (26)$$

Now \dot{I}_3 is written in terms of known quantities. Following this, V_3 , V_6 , and I_6 may be obtained from Equations (10) and (23), (10) and (24), and (7), respectively.

2.3.4 SOLUTION OF VOLTAGE SOURCE CURRENTS AND CURRENT SOURCE VOLTAGES

A complete list of possible outputs of a network would include the current through voltage sources and the voltage across current sources. These can be obtained directly from equations (8) and (4), respectively, since the right sides of these equations are known at this stage of the computational sequence. These steps complete the formal derivation of all network currents and voltages.

2.4 SCANNING PROCEDURE

2.4.1 IDEAL OPERATION (SUBMATRICES B_{14} , B_{25} , AND $B_{36} = 0$)

The series of matrix operations described in section 2.3 could very well be programmed as they are to produce the solution of the general transient analysis problem. All of the matrix multiplication, addition, etc. could be performed at each time step to generate the necessary currents and voltages. Research completed during the study phase of this contract showed, however, that a very significant improvement in computer running time could be achieved by a more efficient utilization of the information contained in the B matrix.

The study report* indicated that various network voltages and currents could be read or "scanned" directly from the rows and columns of the B matrix respectively. This can be put more precisely by

$$V_L = -B V_{TB} \quad (27)$$

$$\text{and } I_{TB} = B^T I_L \quad (28)$$

where I_{TB} , I_L , V_{TB} and V_L represent the vectors of tree branch currents, link currents, tree branch voltages, and link voltages, respectively. If the vectors and the B matrix are partitioned according to the form of figure 2, equations (27) and (28) lead to the first six equations of section 2.3. Quantities V_2 and I_5 are explicitly written in terms of known quantities of equations (2) and (6) if submatrix $B_{25} = 0$. Once the resistive quantities are

* AFWL-TR-65-101, Volume I

determined, equation (5) presents I_4 in terms of known quantities if submatrix $B_{14} = 0$. In the same manner equation (3) presents V_3 in terms of known quantities if submatrix $B_{36} = 0$. Clearly, then, the condition " $B_{14}, B_{25}, B_{36} = 0$ " permits solution of $V_2, I_5, I_4, V_3, I_6,$ and V_1 directly by scanning and without any matrix manipulation. Quantities $I_2, V_5, I_1, V_6, V_4,$ and I_3 can then be determined from the operations given or implied in the non zero portions of equations (11), (12), (21), (24), (19), and (25), respectively. Once all of these quantities are explicitly obtained in terms of known quantities, the equations are stored, compiled, and executed at each time increment without recourse to repeated matrix manipulation.

2.4.2 OPERATION WHEN $B_{25} \neq 0$

In most large networks, submatrix B_{25} does not equal zero. When this happens, quantities V_2 and I_5 cannot be "scanned" out and either Equation (13) or (15) must be solved. Both of these equations require the inversion of a matrix; thus, some matrix manipulation must be done. To illustrate the procedure, assume that some hypothetical network has given rise to the B matrix shown in figure 3. The network is fairly typical in that $B_{14}, B_{36} = 0$, but $B_{25} \neq 0$. The nature of submatrix B_{25} (outlined in figure 3) is such that resistors R_3 and R_D contribute no rows or columns, and the scanning process immediately yields

$$V_{R3} = E_1 - V_{C2} \text{ and } I_{RD} = I_{L1}, \text{ from which follow}$$

$$I_{R3} = (E_1 - V_{C2})/R_3 \text{ and } V_{RD} = R_D I_{L1}$$

These quantities will be updated at each time step without matrix manipulation. The rest of the resistive quantities in the network may be solved by the matrix manipulation implied in equations (13) and (14). Once the resistive quantities are determined, the remaining network quantities may be scanned out and stored. The only repeated matrix manipulation required will be for the three resistor link currents (I_{R1}, I_{R2}, I_{R4}) that cannot be scanned because $B_{25} \neq 0$.

		Class 4		Class 5				Class 7
		C_1	C_2	R_A	R_B	R_C	R_D	E_1
Class 2	R_1	1	0	1	0	0	0	-1
	R_2	0	1	0	1	1	0	0
	R_3	0	1	0	0	0	0	-1
	R_4	0	0	1	0	0	0	0
Class 3	L_1	0	0	0	0	0	1	0
	L_2	0	0	1	0	0	0	0

Figure 3. B Matrix from Hypothetical Network

2.5 SEMI-AUTOMATIC SOURCE DERIVATIVES

The need for source time derivatives is established in equations (9) and (10) where \dot{E}_7 and \dot{J}_8 are required. In situations where non-zero source time derivatives are needed, the user must supply them. These situations are subsequently described.

2.5.1 VOLTAGE SOURCE IS VARIABLE AND $B_{17} \neq 0$

If $B_{17} \neq 0$ and the voltage source is constant, SCEPTRE will automatically supply a zero derivative. In the case where a non-zero derivative is required and the user fails to supply it, the run will be terminated with a diagnostic message. The situation is best recognized by the presence of any circuit loop composed solely of capacitors and at least one variable voltage source.

2.5.2 CURRENT SOURCE IS VARIABLE AND $B_{86} \neq 0$

If $B_{86} \neq 0$ and the current source is constant, SCEPTRE will automatically supply a zero derivative. In the case where a non-zero derivative is required and the user fails to supply it, the run will be terminated with a

diagnostic message. The situation is best recognized by the presence of any circuit cut set composed solely of inductors and at least one variable current source.

2.6 LINEARLY DEPENDENT SOURCES

The ability of SCEPTRE to correctly process linearly dependent sources permits the user to define current and voltage sources that are linear functions of resistor currents and voltages. This feature is expected to be most useful for, but not limited to, applications involving families of small-signal transistor equivalent circuits such as shown in figure 4.

These linearly dependent sources require special treatment because their magnitudes are direct functions of quantities that are not state variables. Unless these sources are specifically processed, they will be updated at the n^{th} time step according to the values of their independent variables at the $(n-1)$ time step (as they would be in PREDICT, for example). This results in a "computational delay," which can lead to large errors throughout the entire network.

Linearly dependent sources are provided for in the general B matrix by the Y and X classification. When these sources exist, SCEPTRE will set up the B matrix as shown in figure 5. Under these circumstances, equations (2) and (6) can be extended to:

$$V_2 = -B_{24} V_4 - B_{25} V_5 - B_{27} E_7 - B_{2y} EY \quad (29)$$

$$I_5 = B_{25}^T I_2 + B_{35}^T I_3 + B_{85}^T J_8 + B_{x5}^T JX \quad (30)$$

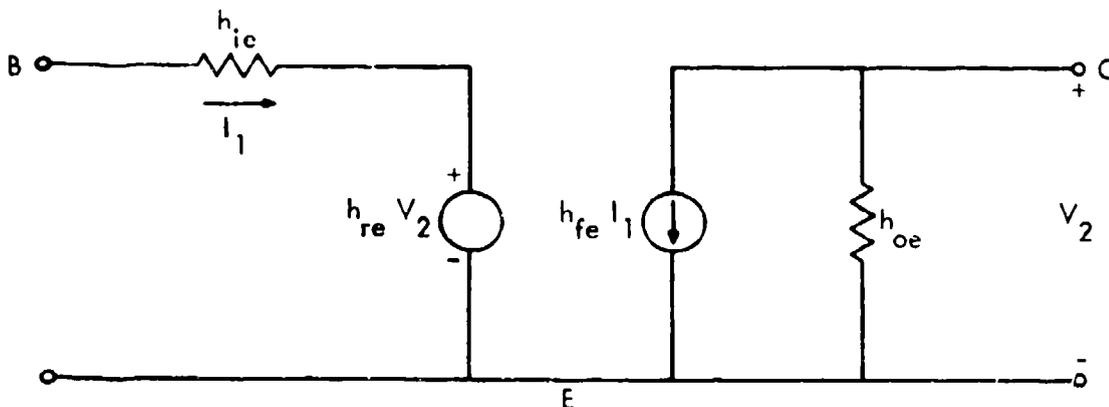


Figure 4. Low-Frequency h-Parameter Equivalent Circuit

	Class 4	Class 5	Class 6	Class 7	Class y
Class 1	B_{14}	0	0	B_{17}	B_{1y}
Class 2	B_{24}	B_{25}	0	B_{27}	B_{2y}
Class 3	B_{34}	B_{35}	B_{36}	B_{37}	B_{3y}
Class 8	B_{84}	B_{85}	B_{86}	B_{87}	B_{8y}
Class x	B_{x4}	B_{x5}	B_{x6}	B_{x7}	B_{xy}

Figure 5. B Matrix with Linearly Dependent Sources

Since any resistor-voltage-dependent voltage source must depend on a resistor tree branch voltage or a resistor link voltage, there arises

$$EY = k_1 V_2 + k_2 V_5 \quad (31)$$

where:

- k_1 is a matrix of constants containing the number of rows equal to the number of class-Y voltage sources and the number of columns equal to the number of nonscannable class-2 elements.
- k_2 is a matrix of constants containing the number of rows equal to the number of class-Y voltage sources and the number of columns equal to the number of nonscannable class-5 elements.
- EY is a vector composed of class-Y voltage sources.

Also, since any resistor-current-dependent current source must depend on a resistor tree branch current or a resistor link current, there arises

$$JX = k_3 I_2 + k_4 I_5 \quad (32)$$

where:

- k_3 is a matrix of constants containing the number of rows equal to the number of class-X current sources and the number of columns equal to the number of nonscannable class-2 elements

- k_4 is a matrix of constants containing the number of rows equal to the number of class-X current sources and the number of columns equal to the number of non-scannable class-5 elements
- JX is a vector composed of class-X current sources.

If Equations (31) and (32), together with $V_2 = R_{22} I_2$ and $I_5 = G_{55} V_5$ are substituted into Equations (29) and (30)

$$R_{22} I_2 + B_{25} V_5 + B_{2y} \left[k_1 R_{22} I_2 + k_2 V_5 \right] = -B_{24} V_4 - B_{27} E_7 \quad (33)$$

$$G_{55} V_5 - B_{25}^T I_2 - B_{x5}^T \left[k_3 I_2 + k_4 G_{55} V_5 \right] = B_{35}^T I_3 + B_{85}^T J_8 \quad (34)$$

The last two equations may be consolidated as

$$\begin{bmatrix} (R_{22} + B_{2y} k_1 R_{22}) (B_{25} + B_{2y} k_2) \\ (-B_{25}^T - B_{x5}^T k_3) (G_{55} - B_{x5}^T k_4 G_{55}) \end{bmatrix} \begin{bmatrix} I_2 \\ V_5 \end{bmatrix} = \begin{bmatrix} -B_{24} V_4 - B_{27} E_7 \\ B_{35}^T I_3 + B_{85}^T J_8 \end{bmatrix}$$

If the large matrix on the left side is called MRG, then

$$\begin{bmatrix} I_2 \\ V_5 \end{bmatrix} = \text{MRG}^{-1} \begin{bmatrix} -B_{24} V_4 - B_{27} E_7 \\ B_{35}^T I_3 + B_{85}^T J_8 \end{bmatrix}$$

and the resistive quantities I_2 and V_5 can be determined without computational delay. Note that since all four "k" matrices are constrained to be constant, the linearly dependent source feature itself will not require any more than one inversion of MRG. If any variable resistors are present in a network, MRG must of course be inverted at each time step. In addition, the extra row and column that was added to the B matrix of figure 2 by the linearly dependent sources will add terms to the equations used in the solution of capacitive, inductive and some source quantities. Specifically, equations (5), (3), (4) and (8) become

$$I_4 = B_{14}^T I_1 + B_{24}^T I_2 + B_{34}^T I_3 + B_{84}^T J_8 + B_{x4}^T JX \quad (5')$$

$$V_3 = -B_{34} V_4 - B_{35} V_5 - B_{36} V_6 - B_{37} E_7 - B_{3y} EY \quad (3')$$

$$V_8 = -B_{84} V_4 - B_{85} V_5 - B_{86} V_6 - B_{87} E_7 - B_{8y} EY \quad (4')$$

$$I_7 = B_{17}^T I_1 + B_{27}^T I_2 + B_{37}^T I_3 + B_{87}^T J_8 + B_{x7}^T JX \quad (8')$$

and the new relations

$$VX = -B_{x4} V_4 - B_{x5} V_5 - B_{x7} E_7 - B_{xy} EY \quad (35)$$

$$IY = +B_{2y}^T I_2 + B_{3y}^T I_3 + B_{84}^T J_8 + B_{xy}^T JX \quad (36)$$

now exist.

A restriction must be placed on these sources based on the content of Section 2.5. The B matrix of figure 5 transforms equations (9) and (10) into:

$$\dot{V}_1 = -B_{14} \dot{V}_4 + B_{17} \dot{E}_7 - B_{1y} \dot{E}Y \quad (9')$$

$$\dot{I}_6 = B_{36}^T \dot{I}_3 + B_{86}^T \dot{J}_8 + B_{x6}^T \dot{J}X \quad (10')$$

Now, additional time derivatives $\dot{E}Y$ and $\dot{J}X$ are required whenever B_{1y} or $B_{x6} \neq 0$.

Differentiation of equations (31) and (32) would involve quantities \dot{V}_2 , \dot{V}_5 , \dot{I}_2 , and \dot{I}_5 . The formulation contains no provisions for these quantities and there is no way the user could know them to supply them as input data. SCEPTRE will automatically check for the existence of non-zero B_{1y} or B_{x6} and terminate the run with a diagnostic message when they occur. The situation can be recognized by the presence of any circuit loop composed solely of capacitors and at least one linearly dependent voltage source, or the presence of any circuit cut set composed solely of inductors and at least one linearly dependent current source.

2.7 INITIAL CONDITIONS SOLUTION

2.7.1 TECHNIQUE DESCRIPTION

Many practical circuits require computer solution of the initial conditions prevailing at the start of the transient (time = t_0) before the transient

solution can begin. These values can always be determined by a separate transient run in which all forcing functions are held at the values for time = t_0 . However, an alternate procedure based on an iteration technique using independent variables other than time was considered desirable. This procedure presents the advantage of economy of machine time on all circuits for which convergence occurs. This section will describe the formulation of this portion of SCEPTRE, which is completely independent of the transient formulation.

If an L tree is set up based on the preference order E, L, R, C, a general B matrix may be set up according to the procedure outlined in section 2.2. The zero-valued submatrices arise from the L tree and the preference order*. The resulting B matrix is shown in table II.

Table II

GENERAL B MATRIX FROM THE L TREE

	4	5	6	7
1	B_{14}	B_{15}	B_{16}	B_{17}
2	0	B_{25}	B_{26}	B_{27}
3	0	0	B_{36}	B_{37}
8	B_{84}	B_{85}	B_{86}	B_{87}
9	B_{94}	B_{95}	B_{96}	B_{97}
0	B_{04}	B_{05}	B_{06}	B_{07}

The following equations (among others) arise from this B matrix if vectors V_6 , I_4 and submatrix B_{94} are assumed to be zero. These assumptions are based on the known final values of V_6 and I_4 for the initial condition

*For example, the submatrix B_{24} must always be zero since non-zero entries in it could only arise when resistor links close loops containing capacitor tree branches. The preference order prohibits this possibility.

problem and the absence of any current-source capacitor cut sets (see the restrictions in section 2.7.4).

$$V_9 + B_{95} V_5 + B_{97} E_7 = 0 \quad (37)$$

$$V_2 + B_{25} V_5 + B_{27} E_7 = 0 \quad (38)$$

$$I_5 - B_{25}^T I_2 - B_{85}^T J_8 - B_{95}^T J_9 - B_{05}^T J_0 = 0 \quad (39')$$

To get all the variables of equation (39') in terms of V_9 , V_2 , V_5 , and effective sources, the following substitutions are made:

$$I_5 = G_{55} V_5$$

$$I_2 = G_{22} V_2$$

$$J_0 = \alpha J_9$$

$$J_9 = G_{99} V_9 + Q_9$$

where:

- α is a matrix containing the number of rows equal to the number of secondary current sources and the number of columns equal to the number of primary current sources
- G_{55} and G_{22} are diagonal matrices containing only conductances
- G_{99} is a diagonal matrix containing only diode and transistor junction conductances
- Q terms are described in appendix II.

Then, equation (39') becomes

$$G_{55} V_5 - B_{25}^T G_{22} V_2 - B_{85}^T J_8 - \left[B_{95}^T + B_{05}^T \alpha \right] \left[G_{99} V_9 + Q_9 \right] = 0 \quad (39)$$

Equations (37), (38), and (39) may be designated as $F_1 (V_9, V_2, V_5)$, $F_2 (V_9, V_2, V_5)$ and $F_3 (V_9, V_2, V_5)$, respectively. If the basic Newton-Raphson method (see appendix III) is applied to equations (37), (38), and (39):

$$F_1 (V_9, V_2, V_5) + \frac{\partial F_1}{\partial V_9} (V_9, V_2, V_5) \Delta V_9 +$$

$$\frac{\partial F_1}{\partial V_2} (V_9, V_2, V_5) \Delta V_2 +$$

$$\frac{\partial F_1}{\partial V_5} (V_9, V_2, V_5) \Delta V_5 = 0$$

$$F_2 (V_9, V_2, V_5) + \frac{\partial F_2}{\partial V_9} (V_9, V_2, V_5) \Delta V_9 +$$

$$\frac{\partial F_2}{\partial V_2} (V_9, V_2, V_5) \Delta V_2 +$$

$$\frac{\partial F_2}{\partial V_5} (V_9, V_2, V_5) \Delta V_5 = 0$$

$$F_3 (V_9, V_2, V_5) + \frac{\partial F_3}{\partial V_9} (V_9, V_2, V_5) \Delta V_9 +$$

$$\frac{\partial F_3}{\partial V_2} (V_9, V_2, V_5) \Delta V_2 +$$

$$\frac{\partial F_3}{\partial V_5} (V_9, V_2, V_5) \Delta V_5 = 0$$

or

$$\begin{bmatrix} F_1 (V_9, V_2, V_5) \\ F_2 (V_9, V_2, V_5) \\ F_3 (V_9, V_2, V_5) \end{bmatrix} + Z \begin{bmatrix} \Delta V_9 \\ \Delta V_2 \\ \Delta V_5 \end{bmatrix} = 0 \quad (40)$$

where the Jacobian is

$$Z \begin{bmatrix} \frac{\partial F_1}{\partial V_9}(V_9, V_2, V_5) & \frac{\partial F_1}{\partial V_2}(V_9, V_2, V_5) & \frac{\partial F_1}{\partial V_5}(V_9, V_2, V_5) \\ \frac{\partial F_2}{\partial V_9}(V_9, V_2, V_5) & \frac{\partial F_2}{\partial V_2}(V_9, V_2, V_5) & \frac{\partial F_2}{\partial V_5}(V_9, V_2, V_5) \\ \frac{\partial F_3}{\partial V_9}(V_9, V_2, V_5) & \frac{\partial F_3}{\partial V_2}(V_9, V_2, V_5) & \frac{\partial F_3}{\partial V_5}(V_9, V_2, V_5) \end{bmatrix} \quad (41)$$

If

$$\Delta V_9 = V_9(n+1) - V_9(n)$$

$$\Delta V_2 = V_2(n+1) - V_2(n)$$

$$\Delta V_5 = V_5(n+1) - V_5(n)$$

where n is used to designate the results of the n^{th} iteration pass, then equation (40) becomes

$$\begin{bmatrix} F_1(V_9, V_2, V_5) \\ F_2(V_9, V_2, V_5) \\ F_3(V_9, V_2, V_5) \end{bmatrix} + Z \begin{bmatrix} V_9(n+1) - V_9(n) \\ V_2(n+1) - V_2(n) \\ V_5(n+1) - V_5(n) \end{bmatrix} = 0 \quad (42)$$

leading directly to

$$\begin{bmatrix} V_9(n+1) \\ V_2(n+1) \\ V_5(n+1) \end{bmatrix} = \begin{bmatrix} V_9(n) \\ V_2(n) \\ V_5(n) \end{bmatrix} - Z^{-1} \begin{bmatrix} F_1(V_9, V_2, V_5) \\ F_2(V_9, V_2, V_5) \\ F_3(V_9, V_2, V_5) \end{bmatrix} \quad (43)$$

Equation (41) may be written more explicitly if the indicated differentiations are performed on equations (37), (38), and (39) to obtain:

$$Z = \begin{bmatrix} I & O & B_{95} \\ O & I & B_{25} \\ -[B_{95}^T + B_{05}^T \alpha] G_{99} & -B_{25}^T G_{22} & G_{55} \end{bmatrix}$$

so that equation (43) becomes

$$\begin{bmatrix} V_9(n+1) \\ V_2(n+1) \\ V_5(n+1) \end{bmatrix} = \begin{bmatrix} V_9(n) \\ V_2(n) \\ V_5(n) \end{bmatrix} - \begin{bmatrix} \\ \\ \\ \end{bmatrix}^{-1} \begin{bmatrix} F_1(V_9, V_2, V_5) \\ F_2(V_9, V_2, V_5) \\ F_3(V_9, V_2, V_5) \end{bmatrix} \quad (44)$$

Equation (44) may be written in more convenient form as follows (see also Appendix IV):

$$\begin{bmatrix} V_9(n+1) \\ V_2(n+1) \\ V_5(n+1) \end{bmatrix} = \begin{bmatrix} I & O & B_{95} \\ O & I & B_{25} \\ -[B_{95}^T + B_{05}^T \alpha] G_{99} & -B_{25}^T G_{22} & G_{55} \end{bmatrix}^{-1} \begin{bmatrix} -B_{97} E_7 \\ -B_{27} E_7 \\ [B_{95}^T + B_{05}^T \alpha] Q_9 + B_{85}^T J_8 \end{bmatrix} \quad (44')$$

Equation (44') signifies a computational sequence as follows: Quantities V_9 , V_2 , and V_5 , the vectors of voltages across primary current sources, resistor link, and resistor tree branches respectively, each have some assumed value (usually zero) to begin the computation at $n = 0$. All members of the right side of equation (44') that may be voltage dependent are updated. The left side of equation (44') is then computed and the first iteration ($n=1$) is complete. The right side of equation (44') is re-evaluated

on the basis of the results of the first iteration and the left side is again computed, thereby completing the second iteration ($n=2$). This process is repeated up to 100 times. After any iteration, if

$$\left| V_{(n+1)} - V_{(n)} \right| \leq 0.001 \left| V_{(n+1)} \right| + 0.0001 \quad (45)$$

is satisfied for the set of all voltages in equation (44), convergence is considered to have occurred and the procedure is terminated. If after 100 iterations equation (45) is not satisfied, a diagnostic message will be printed indicating that convergence has not occurred. Experience to date has shown that the Newton-Raphson procedure will converge for most circuits in less than 30 iterations.

Convergence of the Newton-Raphson method can sometimes be prohibitively delayed if for some reason a large forward bias ($V > 0.8v$) is applied to any diode or transistor junction that has been represented by the user in the conventional closed form $J = I_S (e^{\theta V} - 1)$. In this case, the slope of the diode curve, given by $G_j = \frac{dJ}{dV} = \theta I_S e^{\theta V}$, will contribute a very large term in G_{jj} and consequently in the Z matrix. The practical results of this can be more easily appreciated by consideration of a one equation system as represented by the second equation in Appendix III. A large derivative caused by a highly forward-biased diode leads to a very small step (ΔX) in the independent variable. Many steps will therefore be required to complete convergence. This situation is avoided in SCEPTRE by the inclusion of a subroutine called DISCLF. This subroutine effectively ensures that the true operating point on the diode curve is approached in the iteration process from a lower voltage rather than a higher voltage. Smaller diode slopes are used so that larger (and fewer) steps can be taken toward convergence. This is described in reference 2.

If convergence has occurred, quantities V_9 , V_2 , and V_5 are now known. There remains to compute only capacitor voltages and inductor currents since capacitor currents and inductor voltages must be zero.

2.7.2 COMPUTING CAPACITOR VOLTAGES

From the B matrix in table II the capacitor link voltages can be written in terms of the tree branch voltages as

$$V_1 = -B_{14} V_4 - B_{15} V_5 - B_{17} E_7 \quad (46)$$

In addition the principle of conservation of charge permits

$$-B_{14}^T C_{11} V_1 + C_{44} V_4 = -B_{14}^T C_{11} V_1(0) + C_{44} V_4(0) \quad (47)$$

where $V_1(0)$ and $V_4(0)$ are initial voltages that may be specified by the user.

Equations (46) and (47) lead to

$$V_4 = M_c^{-1} \left\{ -B_{14}^T C_{11} B_{15} V_5 - B_{14}^T C_{11} B_{17} E_7 - B_{14}^T C_{11} V_1(0) + C_{44} V_4(0) \right\} \quad (48)$$

where $M_c = B_{14}^T C_{11} B_{14} + C_{44}$

Once V_4 is determined, V_1 may be found from equation (46).

Note that Class-4 elements can occur only if a capacitor cut set exists. (1) Otherwise equation (48) will be identically zero and all capacitor voltages can be determined from equation (46).

The use of equation (47) permits solution of a class of networks in which the final value of capacitor voltage is dependent on an initial value. Consider the circuit shown in figure 6, which contains a capacitor cut set.

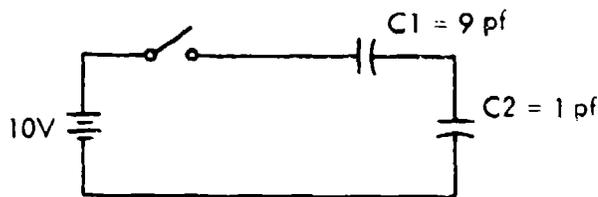


Figure 6. Capacitor Cut Set Circuit

If both capacitors are initially uncharged, the final values of the capacitor voltages after the switch is closed must be $V_{C1} = 1$ volt, $V_{C2} = 9$ volts. If however, capacitor C1 has an initial charge of 5 volts, the principle of conservation of charge (reflected in equation (47)) requires the final result to be $V_{C1} = 5.5$ volts, $V_{C2} = 4.5$ volts.

2.7.3 COMPUTING INDUCTOR CURRENTS

Since V_2 is known,

$$I_2 = R_{22}^{-1} V_2 \quad (49)$$

The inductor tree branch currents then can be written from the B matrix in terms of link currents as

$$I_6 = B_{26}^T I_2 + B_{36}^T I_3 + B_{86}^T J_8 + B_{96}^T J_9 + B_{06}^T J_0 \quad (50)$$

In addition, the flux relations around an inductor loop permit

$$B_{36} L_{66} I_6 + L_{33} I_3 = B_{36} L_{66} I_6(0) + L_{33} I_3(0) \quad (51)$$

where $I_6(0)$ and $I_3(0)$ are initial currents that may be specified by the user. Equations (50) and (51) lead to

$$I_3 = M_L^{-1} \left\{ L_{33} I_3(0) + B_{36} L_{66} I_6(0) - B_{36} L_{66} \left[B_{26}^T I_2 + B_{86}^T J_8 + B_{96}^T J_9 + B_{06}^T J_0 \right] \right\} \quad (52)$$

where $M_L = B_{36} L_{66} B_{36}^T + L_{33}$

Once I_3 is determined, I_6 may be found from equation (50). Note that Class-3 elements can occur only if an inductor loop exists.* Otherwise, equation (52) will be identically zero and all inductor currents can be determined from equation (50).

The use of equation (52) permits the solution of a class of networks in which the final values of inductor currents are dependent on the sizes of the respective inductances. Consider the circuit shown in figure 7, which contains an inductor loop. If both inductors are initially relaxed, the final values of the inductor currents after the switch is closed must be $I_{L1} = 1$ amp, $I_{L2} = 9$ amps.

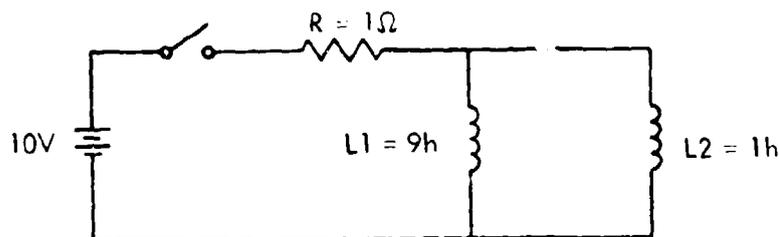


Figure 7. Inductor Loop Circuit

*See paragraph (subsequent section 2.7.4) on network restrictions for qualification of this.

The task of computing the capacitor voltages and inductor currents complete the initial-conditions problem. These quantities may then be transferred to the transient program to serve as initial conditions.

2. 7. 4 RESTRICTIONS

Since the initial-conditions program is formulated differently and for a different purpose than is the transient program, certain restrictions apply to the initial-conditions program that do not apply to the transient program. These restrictions and the practical considerations that lead to them are as follows:

1. No circuit containing a loop composed entirely of voltage sources and inductors will be accommodated. This situation would cause an infinite inductor current and is obviously of no practical importance. The presence of an E-L loop is disclosed by the condition $B_{37} \neq 0$.
2. No circuit containing a cut set composed entirely of current sources and capacitors will be accommodated (see figure 8). This situation would invalidate equation (37) and complicate the solution process by requiring that V_4 be carried along in the Newton-Raphson procedure. This cut set situation can always be removed by arbitrarily connecting a large resistor from node A to the ground. Note that the configuration of figure 8 could be handled by the transient portion of SCEPTRE if a Newton-Raphson solution was not desired. The presence of a J-C cut set is disclosed by the condition that either B_{84} , B_{94} , or $B_{04} \neq 0$.

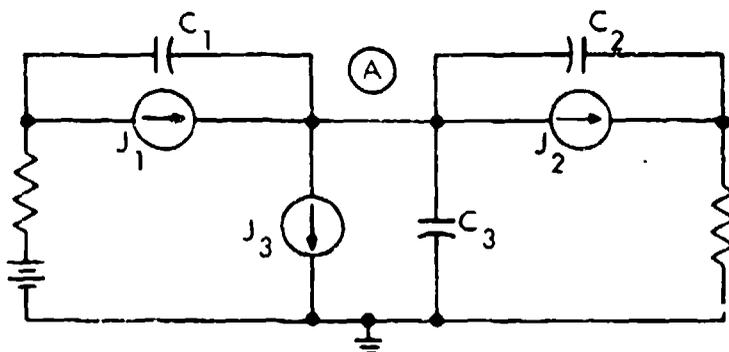


Figure 8. A Current-Source Capacitor Cut Set

3. The choice of independent variables will be somewhat restricted. No resistor or inductor current may be used as an independent variable. Furthermore, a capacitor voltage may be used as an independent variable only if it is in parallel with a resistor or current source ($V_C = V_R$, $V_C = V_J$). The objective here is to avoid the need for any auxiliary computation between passes of the iteration procedure.
4. Networks containing entirely capacitor cut sets can be accommodated only if the members of the cut set are constant.
5. Networks containing only inductor loops can be accommodated only if the members of the loop are constant.

2.8 INTEGRATION ROUTINE

Three integration routines will be optionally available for use in SCEPTRE. Two of these, RUK and TRAP, were available in PREDICT and have been only slightly modified. The third routine, called XPO, was developed at the IBM Scientific Center in Palo Alto, California, by Dr. R. Warten and Mr. M. Fowler and adapted for use in SCEPTRE. Studies to date indicate that XPO is usually, although not always, faster than the other methods. For that reason, this routine will always be used unless the user explicitly requests otherwise in the RUN CONTROL section of any run.

2.8.1 RUK FORMULAS AND VARIABLE STEP SIZE CONTROL

The well-known Runge-Kutta fourth-order-accuracy formulas (Ref. 2) for the numerical solution of the differential equation

$$\dot{y} = f(t, y)$$

are given by

$$y(t+h) = y(t) + 1/6 [k_1 + 2k_2 + 2k_3 + k_4] \quad (5.3)$$

where

$$k_1 = hf [t, y(t)]$$

$$k_2 = hf \left[t + \frac{h}{2}, y(t) + \frac{1}{2} k_1 \right]$$

$$k_3 = h f \left[t + \frac{h}{2}, y(t) + \frac{1}{2} k_2 \right]$$

$$k_4 = h f [t + h, y(t) + k_3]$$

Variable step size control (Ref. 3) is achieved by computing

$$k_5 = h f [t + h, y(t + h)]$$

and

$$E = k_1 - 2k_3 - 2k_4 + 3k_5 \quad (54)$$

The above formulas easily generalize to systems of differential equations, in which case y and E become

$$y(t) = [y_1(t), \dots, y_n(t)]$$

$$E = (e_1, \dots, e_n)$$

Now let

$$u_1, u_2, l_1, l_2 \geq 0$$

Set

$$U_k = u_1 + u_2 \left| y_k(t + h) \right|$$

$$L_k = l_1 + l_2 \left| y_k(t + h) \right|$$

If $|e_k| > 1.5 U_k$ for some k , the integration step h is halved, the independent variable is restored from $t + h$ to t , and the values of y and \dot{y} are restored to the values at time t .

If $|e_k| > 0.75 U_k$ for some k , and $|e_k| \leq 1.5 U_k$ for all k , the current integration step is accepted, but the step size is halved for succeeding steps.

If $L_k \leq |e_k| \leq 0.75 U_k$ for all k , the step size is unaltered.

If $|e_k| \leq U_k$ for all k , and $|e_k| < L_k$ for at least one k , a doubling indicator is activated. Actual doubling is delayed for seven time steps. Halving always takes precedence over doubling; thus anytime a halving signal is received, the step size is halved and doubling is delayed for at least seven steps. Similarly, after successful doubling, another seven steps must elapse before the step size can be doubled again.

Recommended choices for u_1 , u_2 , l_1 , and l_2 are as follows. If absolute error control is desired,

set

$$u_1 = 0.0075 \quad u_2 = 0$$

$$l_1 = 0.00005 \quad l_2 = 0.$$

If relative error control is desired,

set

$$u_1 = 0.005 \quad u_2 = 0.005$$

$$l_1 = 0.00005 \quad l_2 = 0.00005.$$

If smaller step sizes are desired than the ones yielded by the above settings, the 0.0075 and 0.00005 settings should be reduced by a factor of 32. This will yield half the previous step sizes.

2.8.2 MODIFIED TRAPEZOIDAL INTEGRATION (TRAP)

2.8.2.1 Method

Given the differential equation

$$\dot{y}(t) = f(t, y(t)), \quad y(0) = y_0.$$

Consider the following numerical integration scheme,

$$y_p(t + \frac{3}{2}h) = y(t) + \frac{h}{2} \dot{y}(t) \tag{55}$$

$$\begin{aligned}\dot{y}_p(t + \frac{3}{2}h) &= f\left[t + \frac{3}{2}h, y_p(t + \frac{3}{2}h)\right] \\ &= f\left[t + \frac{3}{2}h, y(t) + \frac{h}{2}\dot{y}(t)\right]\end{aligned}\quad (56)$$

$$y_c(t+h) = y(t) + \frac{h}{3}\left[2\dot{y}(t) + \dot{y}_p(t + \frac{3}{2}h)\right]\quad (57)$$

$$\dot{y}_c(t+h) = f\left[t+h, y_c(t+h)\right]\quad (58)$$

Step-size control is achieved by computing

$$E = \frac{2}{3}h \left| \dot{y}_p(t + \frac{3}{2}h) - \dot{y}(t) \right|\quad (59)$$

The magnitude of $|E|$ indicates any changes to be made in the step size.

2.8.2.2 Truncation Error

Inasmuch as $\ddot{y} = f_t + \dot{y} f_y$ and the true solution y_T is approximated by

$$y_T(t+h) = y(t) + h\dot{y}(t) + \frac{h^2}{2}\ddot{y}(t).$$

one obtains,

$$y_T(t+h) = y(t) + h\dot{y}(t) + \frac{h^2}{2}(f_t + \dot{y}f_y).\quad (60)$$

On the other hand

$$f\left[t + \frac{3h}{2}, y(t) + \frac{h}{2}\dot{y}(t)\right] \doteq f(t, y(t)) + \frac{3h}{2}f_t + \frac{h\dot{y}}{2}f_y\quad (61)$$

whence Equation (57) becomes

$$y_c(t+h) \doteq y(t) + \frac{h}{3}\left[2\dot{y}(t) + \dot{y}(t) + \frac{3h}{2}f_t + \frac{h}{2}\dot{y}f_y\right]$$

Subtracting Equation (57) from (60) yields

$$y_T(t+h) - y_c(t+h) \doteq \left(\frac{h^2}{2} - \frac{h^2}{6}\right)\dot{y}f_y = \frac{h^2}{3}\dot{y}f_y$$

which in turn yields the approximate local truncation error.

From equations (56) and (61) one gets

$$\dot{y}_p \left(t + \frac{3h}{2} \right) - \dot{y}(t) \doteq \frac{3h}{2} f_t + \frac{h}{2} \dot{y} f_y$$

whence

$$\frac{2h}{3} \left[\dot{y} \left(t + \frac{3h}{2} \right) - \dot{y}(t) \right] \doteq h^2 f_t + \frac{h^2}{3} \dot{y} f_y \quad (62)$$

One notes that if $f_t = 0$, then equation (62) is a good representation of the local truncation error, neglecting higher order terms.

Experience indicates that for systems of equations of the form $\dot{Y} = A(Y)Y + B$, where A is piecewise constant, the method as well as step-size control is adequate.

2.8.2.3 Stability

Consider the differential equation

$$\dot{y} = -ay, \quad y(0) = y_0, \quad a > 0$$

The numerical integration scheme given by equations (55) through (58) yields

$$y_c(nh) = \left(1 - ah + \frac{(ah)^2}{6} \right)^n y(0) \quad (63)$$

This is easily established by induction. Thus, for numerical stability it is necessary and sufficient that $\left| 1 - ah + \frac{(ah)^2}{6} \right| < 1$. This yields $0 < ah < 6$ as shown in Figure 9.

Inasmuch as $|ah| < 6$ is necessary and sufficient for numerical stability, we say that the modified trapezoidal method has a stability radius, r , equal to 6. Moreover, from equations (55) and (56) we see that the method requires two derivative evaluations (passes) per integration step. Thus, the pass number p associated with the method is 2.

The ratio r/p is a measure of a method's efficiency in the sense of minimum number of integration steps per given time interval.

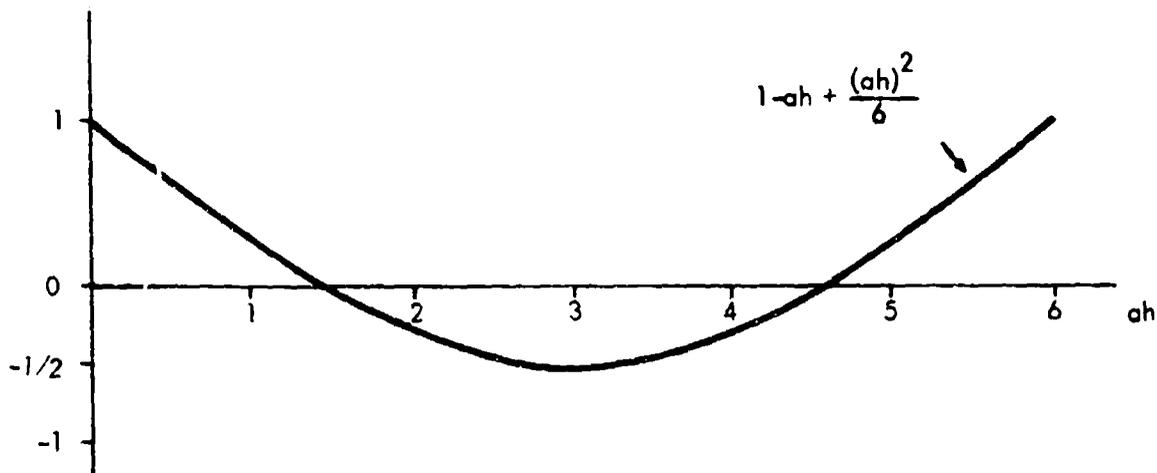


Figure 9. Root Locus of TRAP Characteristic Equation

The higher r/p , the fewer steps are required for a given a .

The following list shows r , p and r/p for a few representative methods:

<u>METHOD</u>	<u>r</u>	<u>p</u>	<u>r/p</u>
Modified Trap.	6	2	3
Euler (Ref. 4)	2	1	2
Trapezoidal (Ref. 5)	2	2	1
NIDE (1PASS) (Ref. 6)	0.8	1	0.8
Runge-Kutta (Fourth-order)	2.78	4	0.7
MIDE (2 PASS)	1	2	0.5
Hamming (Ref. 7)	0.85	2	0.425

2.8.2.4 Step-Size Control

As mentioned previously, the quantity

$$E_2 = \frac{2h}{3} \left[\dot{y}_p \left(t + \frac{3h}{2} \right) - \dot{y}(t) \right]$$

is used for step-size control.

- (1) The method is essentially the same as for RUK. Let u_1, u_2, l_1, l_2 be real numbers ≥ 0 .

$$\text{Compute } U = u_1 + u_2 \left| y_c(t+h) \right|, \quad L = l_1 + l_2 \left| y_c(t+h) \right|$$

If $\left| E_2 \right| > 0.75 U$, the step size is reduced.

If $\left| E_2 \right| < L$, the step size is increased.

If $L \leq \left| E_2 \right| \leq 0.75 U$, the step size is not altered.

- (2) Choice of u_1, u_2, l_1, l_2 .

Recall that E_2 attempts to represent the local truncation error. For absolute error control set

$$u_1 = 0.01, \quad u_2 = 0$$

$$l_1 = 0.0005, \quad l_2 = 0$$

For relative error control set

$$u_1 = 0.001, \quad u_2 = 0.01$$

$$l_1 = 0.00005, \quad l_2 = 0.0005$$

The above values work well in practice.

Reducing the above settings by a factor of 4 will reduce the step size by a factor of 2.

2.8.3 EXPONENTIAL INTEGRATION (XPO)

2.8.3.1 Method

Given the differential equation

$$\dot{y}(t) = f[t, y(t)], \quad y(0) = y_0,$$

this integration routine gives the computed solution

$$y(t+h) = y(t) + \dot{y}_A(t)h + \frac{e^{\lambda h} - 1}{\lambda h} h \dot{y}_p(t) \quad (64)$$

except for two cases that will be covered later. In equation (64)

$$\dot{y}_A(t) = \frac{y(t) - y(t-h_0)}{h_0}$$

where $t-h_0$ is the last point computed:

$$\dot{y}_p(t) = \dot{y}(t) - \dot{y}_A(t); \quad (65)$$

$$\lambda = \dot{y}(t) / \dot{y}_p(t) \quad (66)$$

Since $\dot{y}(t)$ cannot be computed explicitly by the program, we take a small Euler integration step and approximate $\dot{y}(t)$ by $\dot{y}_\delta(t)$:

$$\begin{aligned} y_\delta(t+\delta) &= y(t) + \delta \dot{y}(t) \\ \dot{y}_\delta(t+\delta) &= f[t+\delta, y_\delta(t+\delta)] \\ \ddot{y}_\delta(t) &= \frac{\dot{y}_\delta(t+\delta) - \dot{y}(t)}{\delta}, \quad 0 < \delta \leq \frac{h}{4} \end{aligned} \quad (67)$$

The first exception occurs if $\ddot{y}_\delta(t)$ and $\dot{y}_p(t)$ have the same sign.

Then set

$$\dot{y}_A(t) = 0,$$

which results in equations (65) and (66) becoming

$$\dot{y}_p(t) = \dot{y}(t) \quad (68)$$

$$\lambda = \ddot{y}_\delta(t) / \dot{y}(t).$$

This provides a better approximation to the solution of the equation $\dot{y} = \lambda y + b$ without decreasing the overall effectiveness of the method.

The second exception occurs if λ , computed either by equations (66) or (68) is non-negative. Then the term

$$\frac{e^{\lambda h} - 1}{\lambda h}$$

in Equation (64) is replaced by $1 + \frac{\lambda h}{2}$, thus changing equation (64) to

$$y(t+h) = y(t) + \dot{y}_A(t)h + \left(1 + \frac{\lambda h}{2}\right) h \dot{y}_p(t) \quad (69)$$

It can be shown with some effort that equation (69) is equivalent to a trapezoidal type integration method.

2.8.3.2 Truncation Error

Noting that $\frac{e^x - 1}{x}$, when expanded in a Taylor series, becomes

$$\frac{e^x - 1}{x} = 1 + \frac{x}{2} + \frac{x^2}{6} + O(x^3)$$

Equation (64) can be written

$$y(t+h) = y(t) + \dot{y}_A(t)h + \dot{y}_p(t)h + \frac{h^2}{2} \lambda \dot{y}_p(t) + \frac{h^3}{6} \lambda^2 \dot{y}_p(t) + O(h^4) \quad (70)$$

$$= y(t) + \dot{y}(t)h + \frac{h^2}{2} \ddot{y}_\delta(t) + \frac{h^3}{6} \lambda \ddot{y}_\delta(t) + O(h^4)$$

Referring to equation (67), note that

$$\ddot{y}_\delta(t) = \ddot{y}(t) + \frac{\delta}{2} \left[\dot{y}'(t) - \dot{y}(t) \frac{\partial f}{\partial y} \right] + O(\delta^2) \quad (71)$$

Substituting equation (71) into equation (70) yields

$$y(t+h) = y(t) + \dot{y}(t)h + \frac{h^2}{2}\ddot{y}(t) + \frac{h^2}{4}\delta \left[\dot{y}(t) - \dot{y}(t) \frac{\partial f}{\partial y} \right] + O(h^3) \quad (72)$$

It is noted that the use of either exception to the principal equation (64) still yields the truncated expression, equation (72).

Since $\delta \leq \frac{h}{4}$, the method is second order exact, i. e., the error is of order h^3 . This characteristic will be used in equation (73) in the next section.

2.8.3.3 Step-Size Control

For $t \leq t + \xi \leq t + h$, let $y_T(t + \xi)$ be the true solution to the differential equation, let $y_c(t + \xi)$ be the computed solution, let $\dot{y}_e(t + \xi)$ be an expression obtained by differentiation of $y_c(t + \xi)$, i. e.,

$$\dot{y}_e(t + \xi) = \begin{cases} \dot{y}_A(t) + (1 + \lambda\xi)\dot{y}_p(t), & \lambda \geq 0 \\ \dot{y}_A(t) + e^{-\lambda\xi}\dot{y}_p(t), & \lambda < 0 \end{cases}$$

and let $\dot{y}_c(t + \xi)$ be obtained by substituting $y_c(t + \xi)$ into the differential equation.

Ideally, step control should be based on the expression

$$E_0 = \int_0^h [\dot{y}_T(t + \xi) - \dot{y}_e(t + \xi)] d\xi$$

however, $\dot{y}_T(t + \xi)$ is not available except at $\xi = 0$.

Making use of the fact that

$$\begin{aligned} \dot{y}_c(t + \xi) &= f \left\{ t + \xi, y_T(t + \xi) + [y_c(t + \xi) - y_T(t + \xi)] \right\} \\ &= \dot{y}_T(t + \xi) + O(\xi^3) \end{aligned} \quad (73)$$

we obtain

$$E_0 = \int_0^h [\dot{y}_c(t+\xi) - \dot{y}_e(t+\xi)] d\xi + \int_0^h [\dot{y}_t(t+\xi) - \dot{y}_c(t+\xi)] d\xi$$

$$= \int_0^h [\dot{y}_c(t+\xi) - \dot{y}_e(t+\xi)] d\xi + O(h^4)$$

If $\dot{y}_c(t+\xi) - \dot{y}_e(t+\xi)$ does not change sign for $0 \leq \xi \leq h$,

$$|E_0| \leq h |\dot{y}_T(t+h) - \dot{y}_e(t+h)| \leq h |\dot{y}_c(t+h) - \dot{y}_e(t+h)| + |O(h^4)|$$

Thus for small h ,

$$E = h |\dot{y}_c(t+h) - \dot{y}_e(t+h)|$$

yields an estimate of the truncation error.

Let u_1, u_2, l_1, l_2 be real numbers ≥ 0 . Compute

$$U = u_1 + u_2 |y_c(t+h)|$$

$$L = l_1 + l_2 |y_c(t+h)|$$

If $|E| > U$, the step size is reduced. If $|E| < L$, the step size is increased.
If $L \leq E \leq U$, the step size is not altered.

For absolute error control set

$$u_1 = 0.0075 \quad u_2 = 0$$

$$l_1 = 0.0002 \quad l_2 = 0$$

For relative error control set

$$u_1 = 0.005 \quad u_2 = 0.005$$

$$l_1 = 0.0001 \quad l_2 = 0.0002$$

2.8.4 GENERAL ABSOLUTE AND RELATIVE ERROR CRITERIA

All three numerical integration methods described in this section are approximate methods that are used to integrate differential equations and thereby update the state variables at each time step. Each of these methods has an automatic control that allows step size to be increased and decreased during the transient problem. The method of control used compares some function of the step size and the derivatives of the state variables to a quantity that serves as a standard as, for example, in equation (59). The standard may be a constant or a variable function of the state variable. If the former is used, it is termed an absolute error criterion; if the latter is used, it is termed a relative error criterion.

Consider the situation in which two state variables are very different in size. Let $y_1(t) = 1$ and $y_2(t) = 100$. For all practical purposes, it is usually unnecessary to integrate the derivatives of these state variables to the same accuracy. If an absolute error criterion is used, this is just what is done, and the step size may be unnecessarily inhibited. If, however, a relative error criteria is used, effectively a larger error will be tolerated for the integration of $\dot{y}_2(t)$ and a larger step size will be permitted. In general, then, it would be best for the user to use relative error control when large values of state variables (capacitor voltages and inductor currents) are expected.

Relative error control is programmed with all three integration methods, but the user may easily modify the relative controls or enter absolute controls (section 2.2.9 of Volume I). If larger numbers are used for the u_1 or u_2 entries, the solution process will be less likely to halve any particular solution step size; smaller numbers would increase the likelihood of reduced step sizes. If larger numbers are used for the l_1 and l_2 entries, the solution process is more likely to increase any particular solution step size; smaller numbers make increased step sizes less likely. The user should realize that any increase in solution speed that may result from adjustment of these numbers must necessarily come at the expense of integration accuracy.

2.8.5 IMPLICIT METHOD

The integration methods described in subsections 2.8.1, 2.8.2 and 2.8.3 are all explicit in form and can be used interchangeably within the mathematical formulation of SCEPTRE without difficulty. If, however, any implicit method is to be used, whether single step or multistep, an additional computational step is necessary.

2.8.5.1 Basic Implicit Format

A generalized form of all implicit integration methods can be written as

$$Y(n+1) = \sum_{i=0}^p a_i Y(n-i) + h \sum_{i=-1}^p b_i \dot{Y}(n-i) \quad (74)$$

where Y is the vector of system state variables, \dot{Y} is the vector of state variable derivatives, n is the step number, h is the step size and the a_i, b_i are suitably chosen constants. Multistep methods are introduced if $p > 0$. The simplest derivative form of equation (74) is

$$Y(n+1) = Y(n) + h \dot{Y}(n+1)$$

which is commonly referred to as the implicit or backward Euler technique. If an m th order system of differential equations is to be solved by this method, the following generalized matrix equation will result:

$$\begin{bmatrix} 1 - hg \frac{\partial \dot{Y}_1}{\partial Y_1}(Y_1, \dots, Y_m, t) & \dots & -hg \frac{\partial \dot{Y}_1}{\partial Y_m}(Y_1, \dots, Y_m, t) \\ \vdots & \ddots & \vdots \\ -hg \frac{\partial \dot{Y}_m}{\partial Y_1}(Y_1, \dots, Y_m, t) & \dots & 1 - hg \frac{\partial \dot{Y}_m}{\partial Y_m}(Y_1, \dots, Y_m, t) \end{bmatrix} \begin{bmatrix} \Delta Y_1^k \\ \vdots \\ \Delta Y_m^k \end{bmatrix} = \begin{bmatrix} -F_1(Y_1, \dots, Y_m, t) \\ \vdots \\ -F_m(Y_1, \dots, Y_m, t) \end{bmatrix} \quad (75)$$

The iteration implied in equation (75) is carried out to convergence at each time step. The k superscripts here indicate the k th approximation to the final value at convergence at each step, g is a constant that depends on the order of integration, and F_i is a function of the i th differential equation, the step size and past values of the m th state variable. Sparse matrix techniques will be applied to the operation implied by equation (75) when large problems are encountered.

2.8.5.2 The Jacobian

The \dot{Y}_i terms in equation (75) are readily available from the basic program formulation; but the general partial derivative term

$$\frac{\partial \dot{Y}_i}{\partial Y_j}, \quad 1 \leq i \leq m, \quad 1 \leq j \leq m$$

is not. To determine what is really needed, it is desirable to frame the entire derivation of the symbolic Jacobian in terms of the general matrix equation

$$\dot{Y} = AY + BU + N\dot{U} \quad (76)$$

Since the desired quantities are the general

$$\frac{\partial \dot{Y}_i}{\partial Y_j}$$

it can be seen from partial differentiation of equation (76) that these are contained in the matrix A. Hence the convenience of the general notation given in equation (76).

What now follows is the construction in symbolic form of the general matrix A in terms of the mathematical formulation that was derived in subsection 2.3. Begin with

$$C_{44} \dot{V}_4 = I_4 = B_{14}^T I_1 + B_{24}^T I_2 + B_{34}^T I_3 + B_{84}^T J_8 \quad (5)$$

and

$$V_3 = B_{34} V_4 - B_{35} V_5 - B_{36} V_6 - B_{37} E_7 \quad (4)$$

Substitute equation (9) and (13) into equation (5) for

$$\begin{aligned} C_{44} \dot{V}_4 = & B_{14}^T C_{11} (-B_{14} \dot{V}_4 - B_{17} \dot{E}_7) + B_{24}^T M_R^{-1} (-B_{24} V_4 - B_{27} E_7 \\ & - B_{25} R_{55} B_{35}^T I_3 - B_{25} R_{55} B_{85}^T J_8) + B_{34}^T I_3 + B_{84}^T J_8 \end{aligned} \quad (77)$$

Use equations (24), (10) (23) and (15) into equation (4) for

$$\begin{aligned}
 & L_{33} \dot{I}_3 + L_{36} (B_{36}^T \dot{I}_3 + B_{86}^T \dot{J}_8) \quad (76) \\
 & = -B_{34} V_4 - B_{35} M_G^{-1} \left\{ -B_{25}^T G_{22} (B_{24} V_4 + B_{27} E_7) + B_{35}^T I_3 + B_{85}^T J_8 \right\} \\
 & \quad - B_{36} (L_{63} + L_{66} B_{36}^T) \dot{I}_3 - B_{36} L_{66} B_{86}^T \dot{J}_8 - B_{37} E_7
 \end{aligned}$$

Equation (74) can be manipulated to yield

$$\begin{aligned}
 & (C_{44} + B_{14}^T C_{11} B_{14}) \dot{V}_4 = -B_{14}^T C_{11} B_{17} \dot{E}_7 - B_{24}^T M_R^{-1} (B_{24} V_4 \\
 & \quad - B_{27} E_7 - B_{25} R_{55} B_{35}^T I_3 - B_{25} R_{55} B_{85}^T J_8) \\
 & \quad + B_{34}^T I_3 + B_{84}^T J_8 \quad (77')
 \end{aligned}$$

Equation (78) yields

$$\begin{aligned}
 & \left[L_{33} + L_{36} B_{36}^T + B_{36} L_{63} + B_{36} L_{66} B_{36}^T \right] \dot{I}_3 \quad (78') \\
 & = -B_{34} V_4 - B_{35} M_G^{-1} \left(-B_{25}^T G_{22} B_{24} V_4 - B_{25}^T G_{22} B_{27} E_7 + B_{35}^T I_3 + \right. \\
 & \quad \left. B_{85}^T J_8 \right) - B_{37} E_7 - L_{36} B_{86}^T \dot{J}_8 - B_{36} L_{66} B_{86}^T \dot{J}_8
 \end{aligned}$$

If only the coefficients of V_4 , I_3 , \dot{V}_4 and \dot{I}_3 (the state variables and the state variable derivatives) are retained, equations (77') and (78') become considerably simplified and can be written in matrix notation as

$$\begin{bmatrix} \dot{V}_4 \\ \dot{I}_3 \end{bmatrix} = \begin{bmatrix} M_C & 0 \\ 0 & M_L \end{bmatrix}^{-1} \quad (79)$$

$$\begin{bmatrix} (-B_{24}^T M_R^{-1} B_{24}) (B_{34}^T - B_{24}^T M_R^{-1} B_{25} R_{55} B_{35}^T) \\ (B_{35} M_G^{-1} B_{25}^T G_{22} B_{24} - B_{24}) (-B_{35} M_G^{-1} B_{35}^T) \end{bmatrix} \begin{bmatrix} V_4 \\ I_3 \end{bmatrix}$$

where $M_C = C_{44} + B_{14}^T C_{11} B_{14}$

and $M_L = L_{33} + L_{36} B_{36}^T + B_{36} L_{63} + B_{36} L_{66} B_{36}^T$

The matrix appearing on the right side of equation (79) is now in a form corresponding to the first term on the right side of equation (76). This matrix is the symbolic form of the general A matrix that is needed to use implicit integration (Ref. 8, 9) with the basic program formulation.

Another method is available to construct the Jacobian that is based on a numerical rather than a symbolic approach. Each time that the Jacobian is to be reevaluated, m calls are made to SIMUL 8 to compute

$$\dot{Y}_i' = f(Y_1', \dots, Y_m', t)$$

Approximations to the desired partials are then obtained by

$$\frac{\partial \dot{Y}_i}{\partial Y_j} \approx \frac{\dot{Y}_i' - \dot{Y}_i}{Y_j' - Y_j} \quad (80)$$

Section III

SYSTEM OPERATION

3.1 INTRODUCTION

The SCEPTRE System is a Fortran IV computer program written for the IBM 7090/94 Data Processing System. It consists of two major phases. The first, called the Program Generator, creates (on magnetic tape) another FORTRAN IV computer program containing circuit equations for electrical networks. SCEPTRE does this automatically, using the input data describing the circuit to be analyzed. Both d-c steady-state and transient equations for non-linear circuits can be generated. The second phase, the Circuit Solution Executive, computes the circuit response by solving these equations generated in the FORTRAN IV program.

Operation of the SCEPTRE System depends upon the monitor system --IBSYS-- which controls its execution on the IBM 7090/94. Under control of IBSYS, SCEPTRE executes in two phases as separate job steps that are loaded and executed sequentially. Prior to loading, however, the monitor system performs any required FORTRAN compilation. Programs to be compiled do not have to reside on the standard input tape upon which the input data are stored. Instead, the system can be instructed, via system control cards, to compile and load programs from an alternate input tape. SCEPTRE uses this feature of IBSYS as a means of linking its two job steps. This is illustrated in the System Flow Diagram, figure 10.

3.2 PROGRAM GENERATOR

The Program Generator is an executive program that controls the inputting of circuit description data, generation of a FORTRAN IV subprogram for calculating circuit response, generation of circuit parameter data, storage of circuit models on library tapes, restart of discontinued runs, and re-outputting of computed results. Each of these six program tasks as well as the Program Generator (EXEC1) are described in the flow diagrams, figures 11 through 16.

3.2.1 CIRCUIT DESCRIPTION PROCESSOR

The SCEPTRE circuit description language is used to describe electrical networks. This application-oriented language is powerful, easy to learn and use, and nearly format free. With it, circuits composed of fundamental circuit elements and prestored circuit models can be described. The types of fundamental electrical characteristics allowed are resistance,

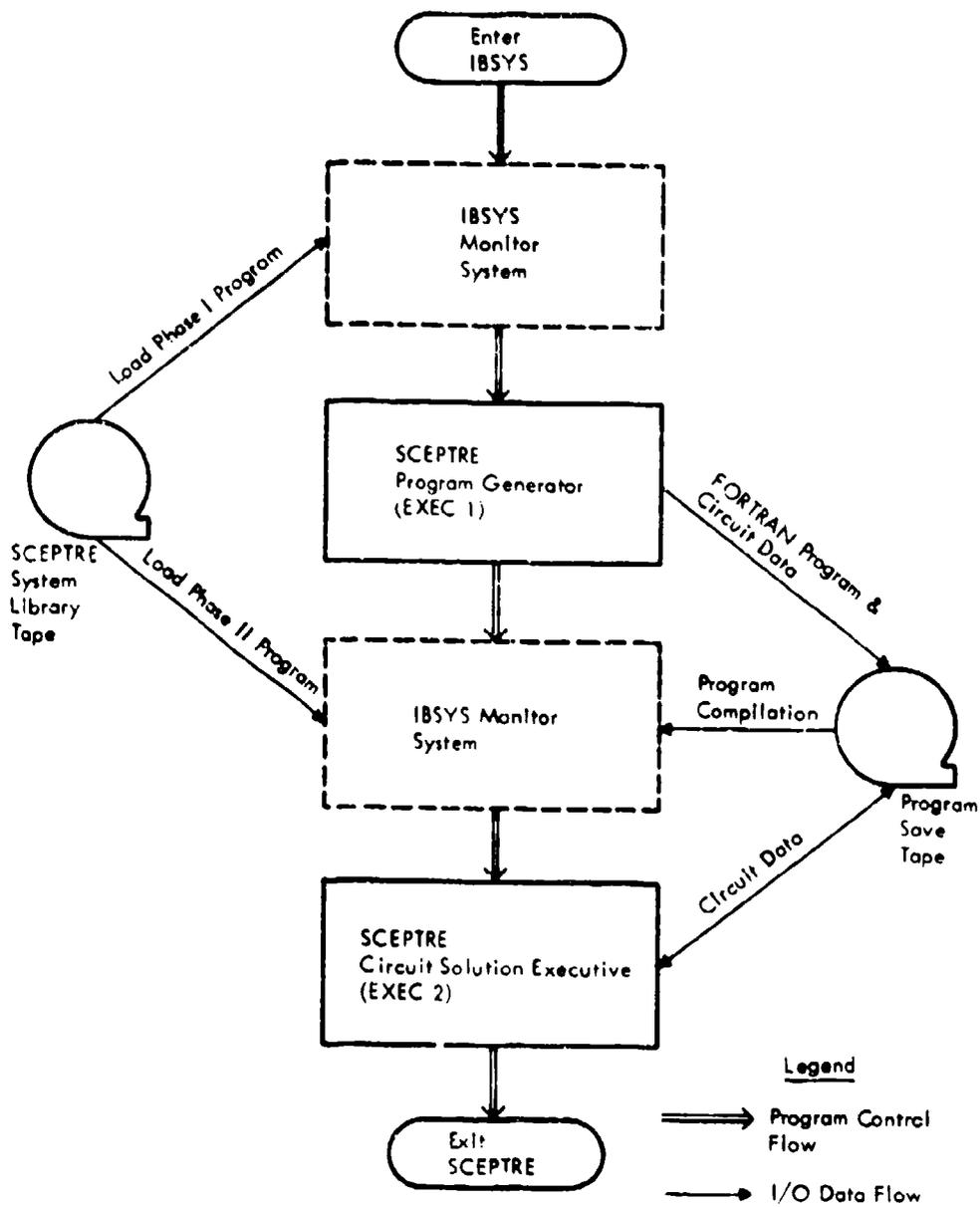


Figure 10. System Flow Diagram

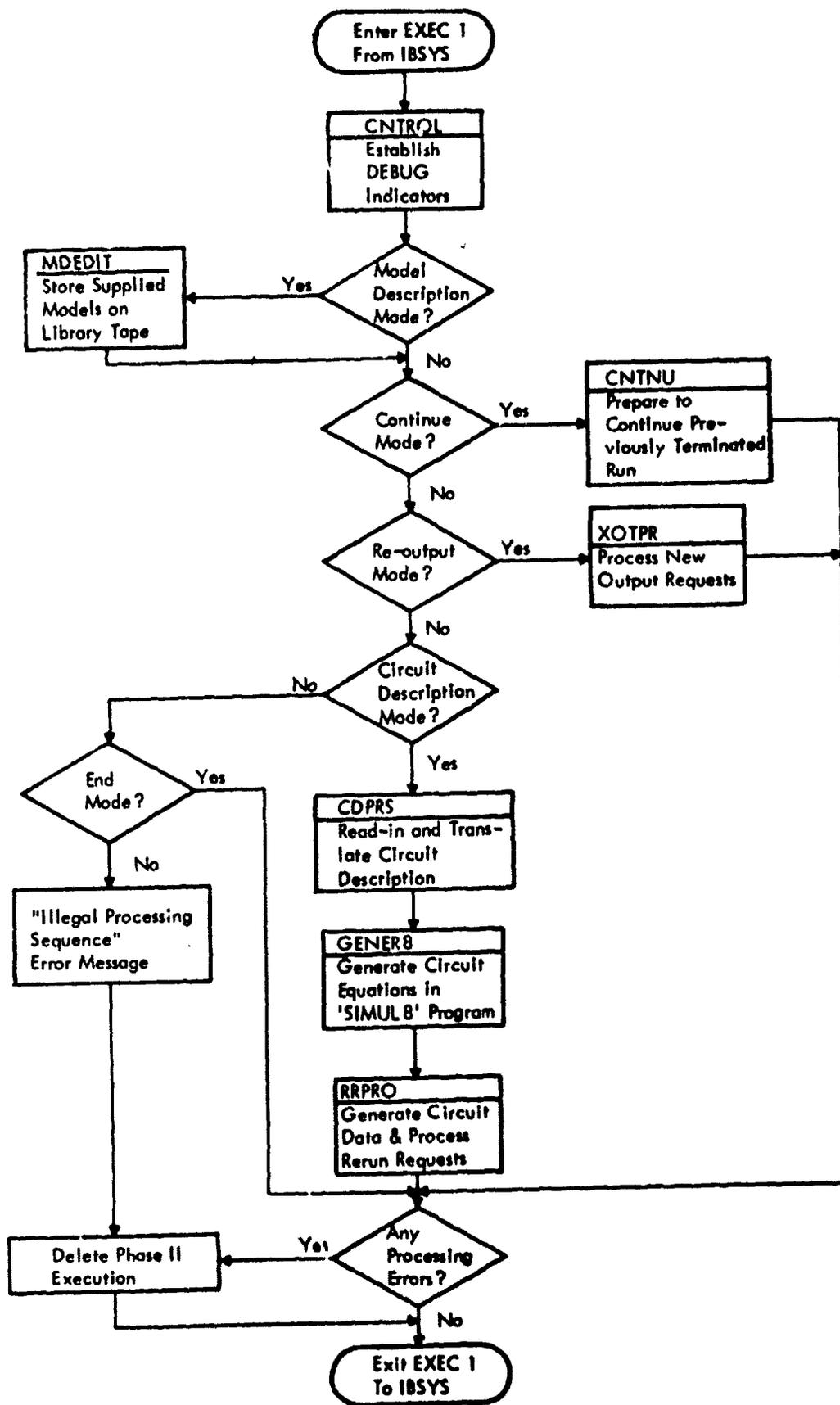


Figure 11. Program Generator (EXEC 1) Flow Diagram

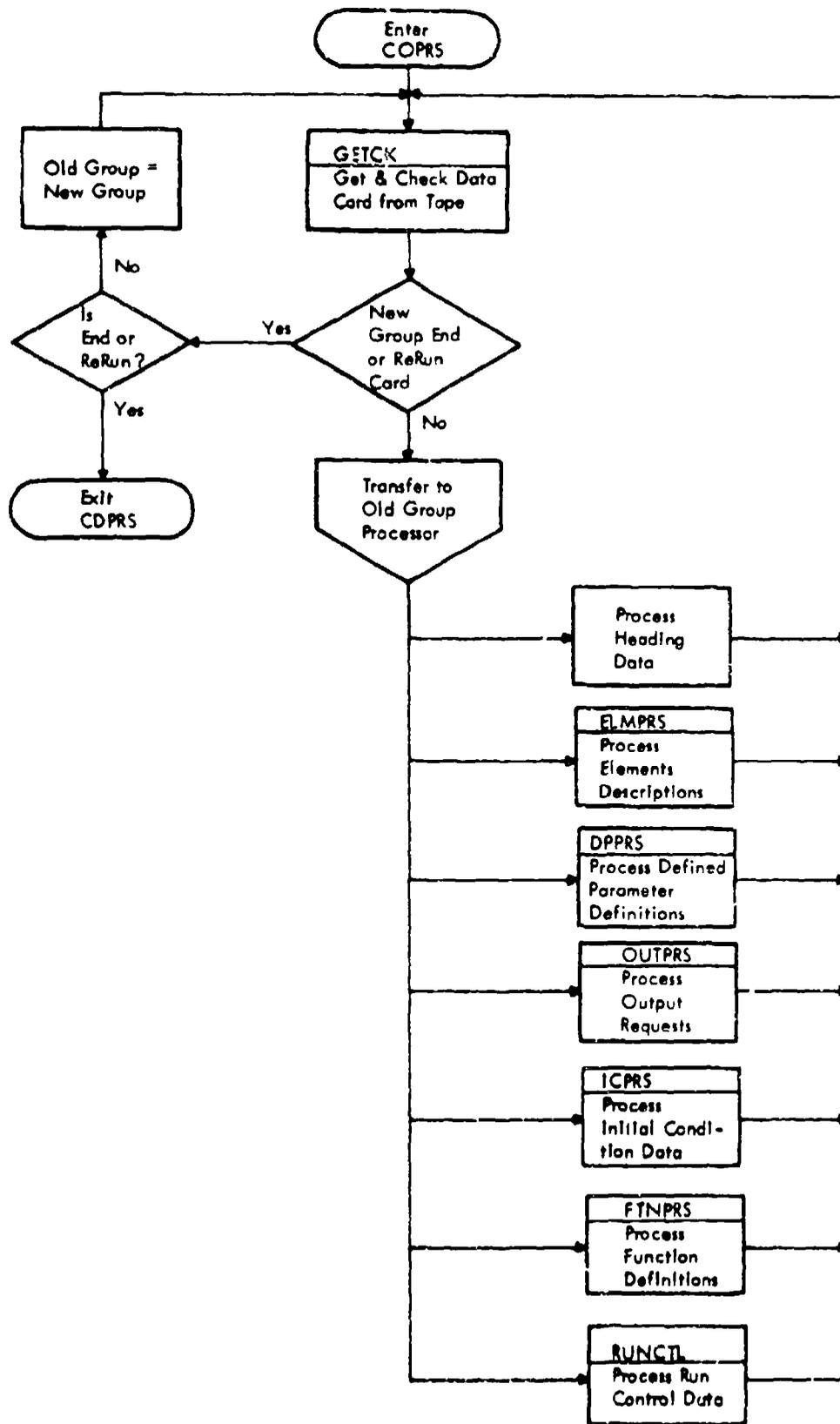


Figure 12. Circuit Description Processor (CDPRS) Flow Diagram

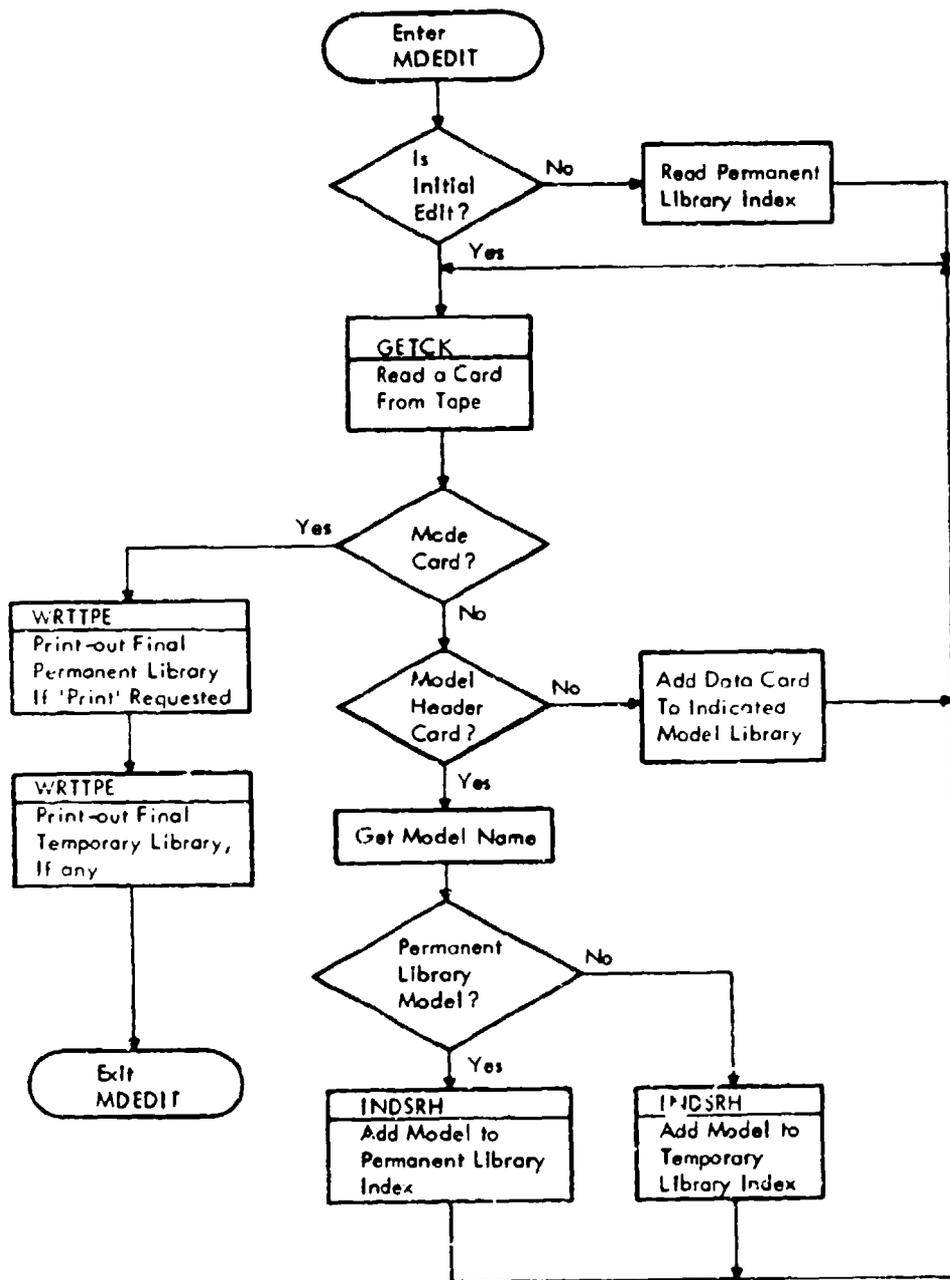


Figure 13. Model Editor (MEDIT) Flow Diagram

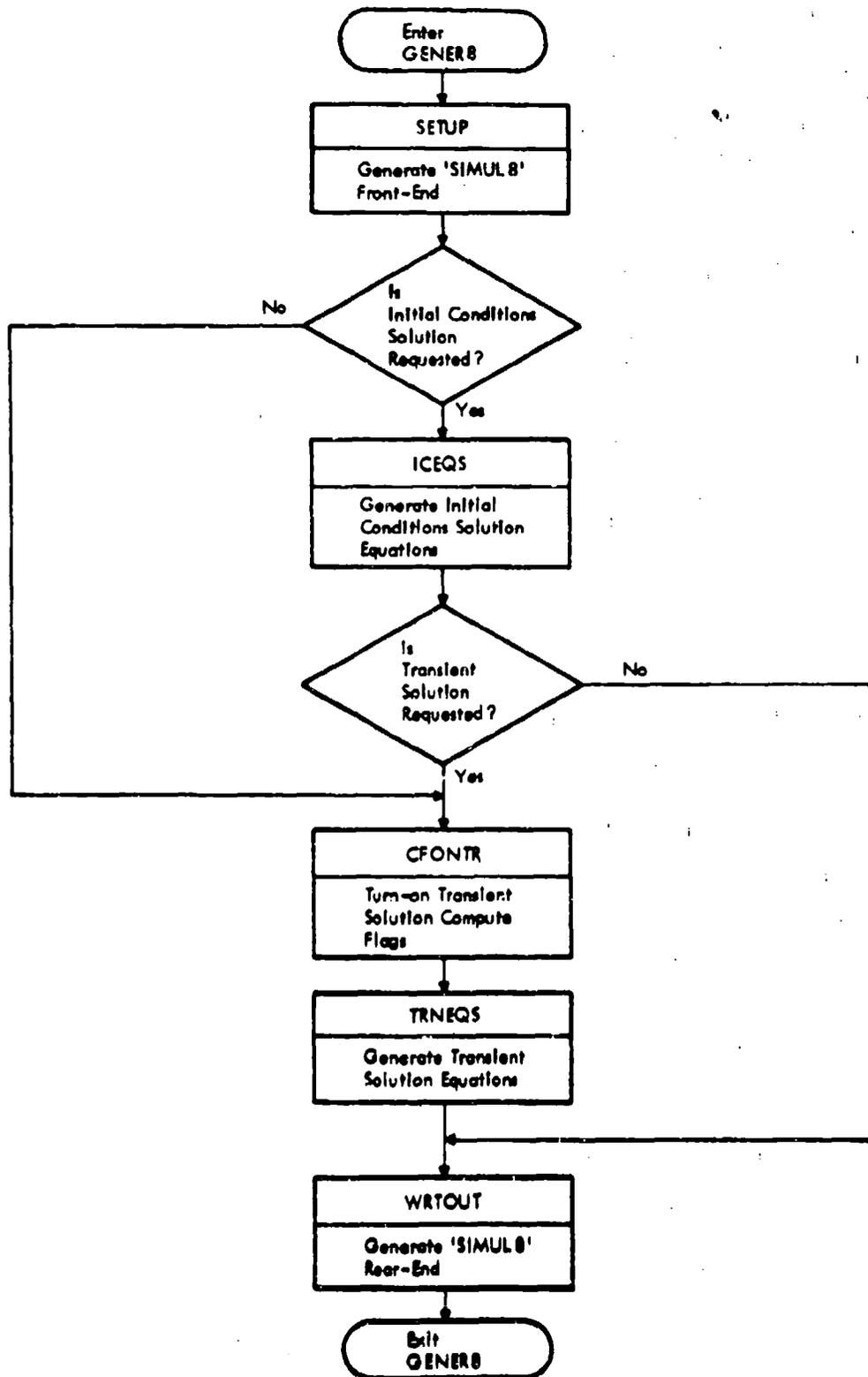


Figure 14. Circuit Equation Generator (GENER 8) Flow Diagram

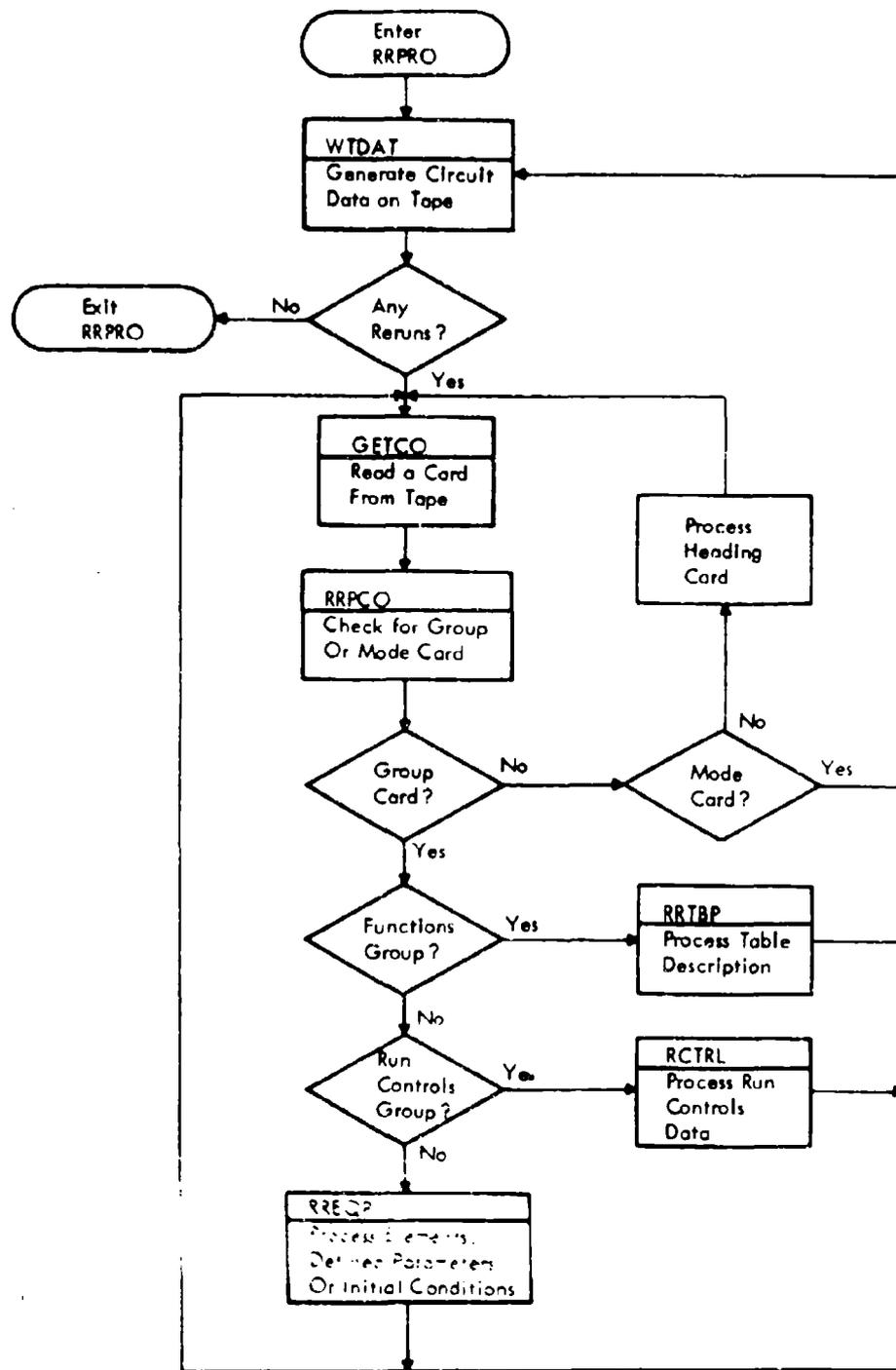


Figure 15. Rerun Processor and Data Generator (RRPRO) Flow Diagram

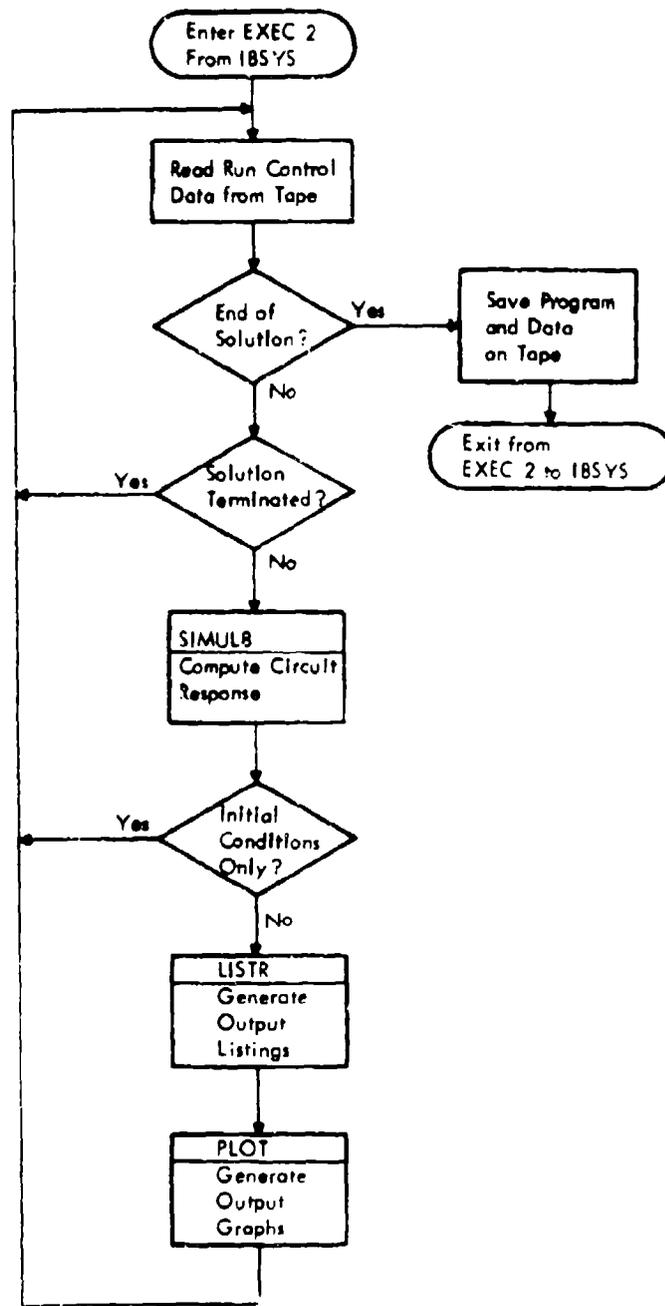


Figure 16. Circuit Solution Executive (EXEC 2) Flow Diagram

capacitance, inductance, current and voltage sources, and mutual inductance. Using the same components and circuit description language, equivalent circuit models of devices such as transistors and diodes can be described and stored on magnetic tape for future use. When a stored model is referenced in a circuit description, it is located on the specified library tape and its elements appropriately substituted into the circuit being described.

Often, when a model is called out in describing a circuit, the desired parameter values are different from those originally specified. Rather than permanently change the model stored on tape, the SCEPTRE circuit description language allows model parameter values to be changed when model element substitution takes place.

3.2.2 MODEL EDITOR

Using the SCEPTRE circuit description language, circuit models constructed by the user can be stored on a model library tape by the Model Editor program. One permanent library tape and one temporary library tape are provided for storing models. Circuit models may be any arbitrary n-terminal configuration of the allowed fundamental circuit elements. Any model so described can be stored on either library tape using the Model Editor program.

3.2.3 CIRCUIT EQUATION GENERATOR

After the description of the circuit to be analyzed has been reconstructed in memory by the Circuit Description Processor, the FORTRAN IV sub-program called SIMUL8 is created. It contains either d-c steady-state equations, transient solution equations, or both, depending on the type of analysis requested by the user. These equations are based on the formulation presented in Section II of this report. The SIMUL8 program is written on magnetic tape (PROGRAM SAVE TAPE) and stored until the second phase of SCEPTRE operation, when it will be compiled and executed. The manner in which the circuit equations are written provides a very efficient computation of the circuit response during execution, thus allowing the response of a large circuit to be calculated rapidly.

3.2.4 DATA GENERATOR AND RERUN PROCESSOR

Essentially the SIMUL8 program contains only the circuit equations for the network under investigation. The parameter or component values of the network are stored separately as input data to the SIMUL8 program. The Data Generator program organizes and stores the circuit data on the PROGRAM SAVE TAPE.

Once a circuit has been described to SCEPTRE, multiple or repeated circuit solutions can be run by changing parameter values between runs. The circuit description language is used to specify the number of repeated runs, each known as a rerun, and the changes in parameter values desired for each rerun. The Rerun Processor interprets and processes this information. The Data Generator then creates and stores one block of circuit data for each rerun on the PROGRAM SAVE TAPE. Since only parameter values are changed between reruns (i. e., no topological changes), the SIMUL8 program containing the original circuit equations is simply re-executed during the solution phase using the data created for each rerun.

3.2.5 CONTINUE PROCESSOR

The Continue Processor allows previously terminated transient solution runs to be restarted and continued. In addition, it is necessary that the PROGRAM SAVE TAPE from the original transient solution run be available. This tape contains, in addition to the SIMUL8 program, all of the parameter values and data required to continue the transient solution. The Continue Processor also allows certain run control parameters to be changed and used for the continuation of the original transient run.

3.2.6 RE-OUTPUT PROCESSOR

The SCEPTRE user may wish to re-create both the printed and plotted outputs for a particular transient analysis run. This can be done, providing the OUTPUT SAVE TAPE is saved from the original solution run, by using the Re-Output Processor. Several changes in the output can be made during a Re-Output run. For example:

- Output labels can be changed
- New quantities can be plotted
- The order in which output quantities are printed out and plotted can be changed
- Printing and plotting of output quantities can be suppressed.

3.3 SOLUTION EXECUTIVE

In the second phase of SCEPTRE operation, the FORTRAN IV subprogram SIMUL8 is compiled and executed. During the loading of this job step, the IBSYS system is instructed to read the SIMUL8 program from the PROGRAM SAVE TAPE, compile it, and then load it along with the other programs being loaded.

Execution commences, as shown in figure 16, with the reading of the run control parameters from the PROGRAM SAVE TAPE. Then the SIMUL8 program is called. It reads the remaining circuit parameter values stored on the PROGRAM SAVE TAPE and then enters the circuit solution equations. The circuit equations are solved, thus computing the state-variable derivatives. Calling the selected numerical integration routine then produces new values of the state variables for the next time step. At the conclusion of each successful integration step, the requested output quantities are buffered in memory. When the buffer is full, it is written on the OUTPUT SAVE TAPE in FORTRAN IV binary format. After the circuit solution is complete, control is returned to the Solution Executive Program, whereupon the contents of the OUTPUT SAVE TAPE (computed results) are re-formatted in lists and graphs and stored on the SYSTEM OUTPUT TAPE for peripheral processing.

If any reruns were requested, control is then returned to SIMUL8 for re-execution of the solution phase and subsequent outputting. This recycling is continued until all circuit reruns have been processed.

At the conclusion of each transient solution, the critical parameter values and data are stored on the PROGRAM SAVE TAPE. Thus, if the tape is saved at the end of the run, the solution can be continued at some future time. Use of the Continue Processor allows previously terminated solution runs to be restarted and continued with changes in the run control parameters.

Transient solution runs may be terminated or simply saved by the computer operator by depressing sense switch No. 6 on the 7090/94 console. Using this feature, a PROGRAM SAVE TAPE could, for example, be generated every 15 minutes on long running transient solutions, eliminating the need for repeating previous calculations in the event of an abnormal run termination.

If, at the conclusion of a transient analysis run, the OUTPUT SAVE TAPE is saved, the Re-Output Processor may be used to reproduce lists and graphs of any of the circuit quantities originally requested for output. In addition, graphs of variables plotted against variables other than time which may not have been requested originally can be conveniently produced.

Appendix I

B MATRIX DERIVATION

The derivation of the general B matrix that expresses link voltages in terms of tree branch voltages and tree branch currents in terms of link currents is as follows:

Two fundamental incidence matrices that arise from network topology theory will be called the Q and T matrices here. The fundamental cut set matrix, $Q = [q_{ij}]$ is a matrix containing $(n-1)$ rows and b columns for a network containing n nodes and b elements, where:

$q_{ij} = +1$ if the i^{th} fundamental cut set direction coincides with the reference direction of the j^{th} element

$q_{ij} = -1$ if the i^{th} fundamental cut set direction is in opposition to the reference direction of the j^{th} element

$q_{ij} = 0$ if the i^{th} fundamental cut set does not include the j^{th} element

If the elements are properly ordered, it is always true that

$$Q = \begin{bmatrix} -B^T & U \end{bmatrix}$$

where the columns of the unit matrix U correspond to the tree branch elements.

The fundamental circuit matrix, $T = [t_{ij}]$ is a matrix containing m rows and b columns for a network containing m independent loops and b elements, where:

$t_{ij} = +1$ if the i^{th} independent loop direction coincides with the reference direction of the j^{th} element

$t_{ij} = -1$ if the i^{th} independent loop direction opposes the reference direction of the j^{th} element

$t_{ij} = 0$ if the i^{th} independent loop does not include the j^{th} element

If the elements are properly ordered, it is always true that

$$T = [U \ B]$$

where the columns of the unit matrix U correspond to the network links.

Since it is always true that

$$Q I_b = 0. \quad T V_b = 0$$

direct substitution yields

$$\begin{bmatrix} -B^T U \\ \end{bmatrix} \begin{bmatrix} I_L \\ I_{TB} \end{bmatrix} = 0 \text{ and } \begin{bmatrix} U \ B \end{bmatrix} \begin{bmatrix} V_L \\ V_{TB} \end{bmatrix} = 0$$

Expansion of these relations gets

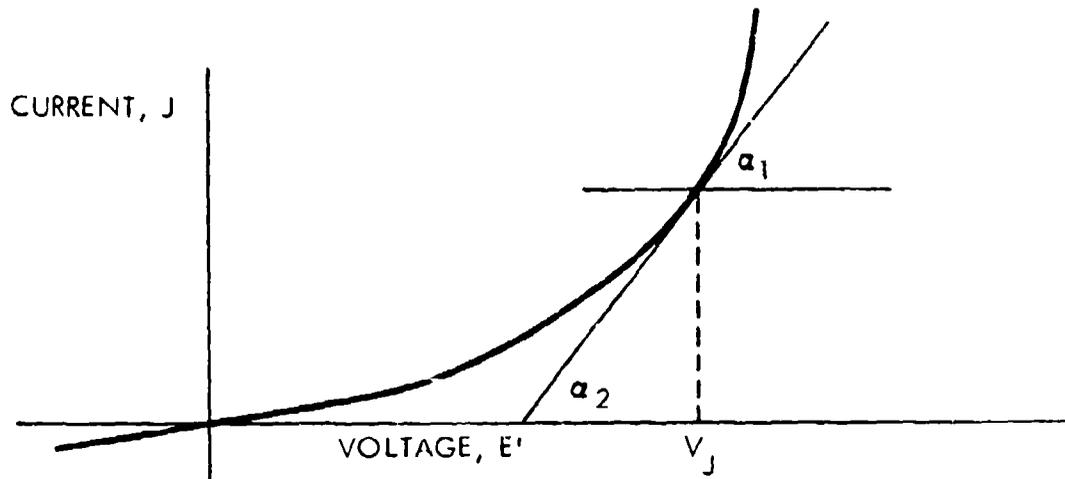
$$I_{TB} = B^{-1} I_L \text{ and } V_L = -B V_{TB}$$

so that the tree branch currents may be expressed in terms of the link currents and the link voltages may be expressed in terms of the tree branch voltages through the B matrix.

Appendix II

DIODE REPRESENTATION IN THE INITIAL-CONDITIONS PROGRAM

The current in a diode or transistor junction at a point on its voltage vs. current (V/I) characteristic may be separated into two components as $J = G_J V_J + Q$. Consider the typical diode curve below.



Angle $\alpha_1 = \text{angle } \alpha_2$ is enclosed by the slope of the diode characteristic at any point and the horizontal at that point. E' is an offset voltage that marks the intersection of a continuation of the slope G_J and the line $J = 0$.

From the figure:

$$\tan \alpha_1 = \tan \alpha_2 = G_J = \frac{J}{V_J - E'}$$

$$\text{or } J = G_J V_J - G_J E'$$

$$\text{or } J = G_J V_J + Q \text{ if } Q \text{ is defined as } -G_J E'$$

Appendix III

BASIC NEWTON-RAPHSON METHOD

Given a single algebraic or transcendental equation of the form $F(x) = 0$, that is single valued and differentiable in the domain of interest, a Newton-Raphson procedure can be constructed using

$$F(x) + \frac{\partial F(x)}{\partial x} \Delta x = 0$$

An initial $x = x_0$ may be assumed and $F(x_0)$, $\frac{\partial F(x_0)}{\partial x}$ determined.

Then $\Delta x = \frac{-F(x_0)}{\frac{\partial F(x_0)}{\partial x}}$ and $x_1 = x_0 + \Delta x$

The procedure is repetitive until

$$\left| F(x_{n+1}) - F(x_n) \right| < z$$

where z is some specified convergence criteria. When this last relation is satisfied, the process is said to have converged. Extension to systems of equations adds no complications.

Appendix IV

EQUIVALENCE OF EQUATIONS (44) AND (44')

To show the equivalence of equations (44) and x (44'), it will be sufficient to show that

$$\begin{bmatrix} V_9 \\ V_2 \\ V_5 \end{bmatrix} - [Z]^{-1} \begin{bmatrix} F_1 (V_9, V_2, V_5) \\ F_2 (V_9, V_2, V_5) \\ F_3 (V_9, V_2, V_5) \end{bmatrix} = [Z]^{-1} \begin{bmatrix} -B_{97} E_7 \\ -B_{27} E_7 \\ [B_{95}^T + B_{05}^T] Q_9 + B_{85}^T J_8 \end{bmatrix}$$

or

$$\begin{bmatrix} V_9 \\ V_2 \\ V_5 \end{bmatrix} = [Z]^{-1} \left\{ \begin{bmatrix} -B_{97} E_7 \\ -B_{27} E_7 \\ [B_{95}^T + B_{05}^T] Q_9 + B_{85}^T J_8 \end{bmatrix} \cdot \begin{bmatrix} F_1 (V_9, V_2, V_5) \\ F_2 (V_9, V_2, V_5) \\ F_3 (V_9, V_2, V_5) \end{bmatrix} \right\}$$

or

$$[Z] \begin{bmatrix} V_9 \\ V_2 \\ V_5 \end{bmatrix} = \begin{bmatrix} -B_{97} E_7 \\ -B_{27} E_7 \\ [B_{95}^T + B_{05}^T] Q_9 + B_{85}^T J_8 \end{bmatrix} \cdot \begin{bmatrix} F_1 (V_9, V_2, V_5) \\ F_2 (V_9, V_2, V_5) \\ F_3 (V_9, V_2, V_5) \end{bmatrix}$$

The left side of the above equation expands into

$$\begin{bmatrix} V_9 + B_{95} V_5 \\ V_2 + B_{25} V_5 \\ - [B_{95}^T + B_{05}^T] G_{99} V_9 - B_{25}^T G_{22} V_2 + G_{55} V_5 \end{bmatrix}$$

and the right side, upon substitution of equations (37) through (39) becomes

$$\begin{bmatrix} V_9 + B_{95} V_5 \\ V_2 + B_{25} V_5 \\ - [B_{95}^T + B_{05}^T \alpha] G_{99} V_9 - B_{25}^T G_{22} V_2 + G_{55} V_5 \end{bmatrix}$$

which shows the equivalence.

REFERENCES

1. Reed, M. B. and Seshu, S., Linear Graphs and Electrical Networks, Addison-Wesley Publishing Company, Reading, Mass., 1961.
2. F. B. Hildebrand, Introduction to Numerical Analysis, McGraw-Hill, New York, 1956, p. 237.
3. R. M. Warten, "Automatic Step-Size Control for Runge-Kutta Integration. "IBM Jnl. Res. Dev., Vol. 7, No. 4, October 1963, pp. 340-341.
4. (Euler Method) W. Kaplan, Ordinary Differential Equations, Addison Wesley, Reading, Mass., 1958, Section 10-2, pp. 400-401.
5. (Trapezoidal Method) Ibid., Section 10-4, "Heun's Method," pp. 402-403.
6. (NIDE) Ashar, K. G., Ghosh, H. N., Aldridge, A. W., Patterson, L. J., "Transient Analysis and Device Characterization of ACP Circuits," IBM Jnl. of Res. & Dev., Vol. 7, p. 218. 1963.
7. (Hamming's Method) P. E. Chase, "Stability Properties of Predictor-Corrector Methods for Ordinary Differential Equations," Jnl. A. C. M. 9, October 1962, pp. 457-468.
8. Gear, C. W., "The Automatic Integration of Ordinary Differential Equations," Comm. of A. C. M., Vol. 14, March 1971.
9. Gear, C. W., "The Numerical Integration of Ordinary Differential Equations," Math. Comp., 21, 1967.

BIBLIOGRAPHY

Portasik, J., and Glass, C., "Systems Analysis Using the SCEPTRE Computer Program," Air Force Weapons Laboratory Technical Report 68-98, Sept. 1968.

Sedore, S. R., "SCEPTRE: A Program for Automatic Network Analysis," IBM Journal of Research and Development, Vol. II, Nov. 1967.

Sedore, S. R., "More Efficient Use of the F Matrix in Practical Circuit Analysis Program," IEEE Trans. on Computers, Vol. C-17, May 1968.

Fowler, M., and Warten, R., "A Numerical Integration Technique for Ordinary Differential Equations with Widely Separated Eigenvalues," IBM Journal of Research and Development, Vol. II, Sept. 1967.

Dickhaut, R. H., "Advances in Computer Techniques for Radiation Effects Calculation," Boeing Document D2-90571, 1964.

Cordwell, W. C., "Radiation Device Modeling for the SCEPTRE Program", International Business Machines, Owego, N. Y., Report 69-825-2371, June 1969.

Bowers, J., and Sedore, S., "SCEPTRE: A Computer Program for Circuit and Systems Analysis," Prentice-Hall, 1971.